



Τεχνολογικό Εκπαιδευτικό Ίδρυμα Κρήτης
Σχολή Τεχνολογικών Εφαρμογών
Τμήμα Εφαρμοσμένης Πληροφορικής & Πολυμέσων



Πτυχιακή Εργασία

Τίτλος: Υλοποίηση πλατφόρμας ηλεκτρονικού εμπορίου

Σφακιανάκης Νικόλαος (ΑΜ:1697)
Φανταουτσάκης Μάριος (ΑΜ:1933)

Επιβλέπων Καθηγητής: Παχουλάκης Ιωάννης

ΗΡΑΚΛΕΙΟ 2012

Ευχαριστίες

Ευχαριστώ όλους όσους συμμετείχαν για την εκπόνηση της παρούσας πτυχιακής εργασίας. Ειδικότερα τον εισηγητή ΚΟ Παχουλάκη για την καθοδήγηση του. Τα μέλη της εξεταστικής επιτροπής που έκαναν την τιμή να αξιολογήσουν την προσπάθεια μας. Τέλος, την οικογένεια μου, που όλα τα χρόνια με στήριξε με κάθε τρόπο.

Σας Ευχαριστώ,
Φανταουτσάκης Μάριος

Εγώ με τη σειρά μου, να ευχαριστήσω τον επιβλέπων καθηγητή Κο Παχουλάκη για την υποστήριξη που μας παρείχε καθ όλη τη διάρκεια της υλοποίησης της πτυχιακής, τον καθηγητή Κο Ακουμιανάκη και τη συνδρομή του στο σχεδιασμό της βάσης δεδομένων, την οικογένεια μου και τέλος, τα μέλη της εξεταστικής επιτροπής για το χρόνο που δαπάνησαν για την αξιολόγηση αυτής εδώ της πτυχιακής.

Ευχαριστώ πολύ,
Σφακιανάκης Νίκος

Abstract

Implementation of e-commerce platform

It's about a dynamic website that will set up an e-commerce store computer from which customers can buy products varied from already composed computers to the smallest accessory for upgrading their computer.

To achieve this e-shop we will use the Microsoft Visual Studio, an integrated development environment. The site is built in ASP.NET in order to take advantage of Web Forms and functions of Visual Studio and we can have well-organized and understandable code. The code used is written in Visual Basic, a language relatively easy for beginners to programming.

The application will have the database Sql Server. The base is necessary for the registration of products, categories that products are separated, customer information and history for the purchases he make. Also, in base the characteristics of each product category are stored and then the value attribute of the product. This makes it easier for the consumer to do market research.

When a user enters the website, he can browse, search for products that interest him and add to shopping cart. Since those who have given the necessary information to create an account and have logged in, can purchase the products they placed in their cart. When login is done successfully, user may access the page with all the personal information that can be modified and history of purchases.

Finally, the shop owners (administrators) are redirected to a separate page where they can edit all data in the database, such as registration of new products, modify characteristics or attribute value product, delete product, remove order upon notification by the user, etc.

To get this result we are using the following techniques and tools: Microsoft Visual Studio 2010, Photoshop, CSS, Sql Server Management Studio, HTML etc.

Σύνοψη

Υλοποίηση πλατφόρμας ηλεκτρονικού εμπορίου

Έχει να κάνει για μια δυναμική ιστοσελίδα στην οποία θα στηθεί ένα ηλεκτρονικό κατάστημα εμπορίου υπολογιστών από το οποίο θα μπορούν οι πελάτες να προμηθεύονται προϊόντα από έτοιμους υπολογιστές μέχρι και το μικρότερο εξάρτημα για την αναβάθμιση του υπολογιστή τους.

Για την υλοποίηση του καταστήματος αυτού θα χρησιμοποιηθεί το Microsoft Visual Studio, ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE - Integrated development environment). Η ιστοσελίδα είναι κτισμένη σε ASP.NET και έτσι εκμεταλλευόμενοι τα Web Forms και τις λειτουργίες του Visual Studio μπορούμε να έχουμε καλά οργανωμένο και κατανοητό κώδικα. Ο κώδικας που χρησιμοποιούμε είναι γραμμένος σε γλώσσα Visual Basic, μια γλώσσα σχετικά εύκολη και για αρχάριους στον προγραμματισμό.

Η εφαρμογή θα έχει βάση δεδομένων Sql Server. Η βάση είναι απαραίτητη για την καταχώρηση των προϊόντων, τις κατηγορίες που θα χωρίζονται τα προϊόντα, τα στοιχεία του πελάτη και το ιστορικό για τις αγορές που θα πραγματοποιεί. Επίσης, στη βάση αποθηκεύονται τα χαρακτηριστικά κάθε κατηγορίας προϊόντος και στη συνέχεια η τιμή χαρακτηριστικού του προϊόντος. Αυτό κάνει πιο εύκολη την έρευνα αγοράς του καταναλωτή.

Όταν ένας χρήστης εισέρχεται στην ιστοσελίδα, μπορεί να περιηγηθεί, να ψάξει για προϊόντα που τον ενδιαφέρουν και να τα προσθέσει στο καλάθι αγορών. Εφ' όσον έχει δώσει τα απαραίτητα στοιχεία για να δημιουργήσει λογαριασμό και έχει κάνει το login ο χρήστης μπορεί να προβεί στην αγορά των προϊόντων που έχει τοποθετήσει στο καλάθι αγορών. Όταν έχει γίνει επιτυχώς το login μπορεί να μεταβεί και στη σελίδα με όλα τα προσωπικά του στοιχεία, που μπορεί να τα τροποποιήσει και το ιστορικό των αγορών του.

Τέλος, για τους ιδιοκτήτες του καταστήματος (administrators), υπάρχει ξεχωριστή σελίδα όπου μπορούν να επεξεργαστούν όλα τα δεδομένα της βάσης, όπως καταχώρηση νέων προϊόντων, τροποποίηση χαρακτηριστικών ή τιμή χαρακτηριστικού προϊόντος, διαγραφή προϊόντων, διαγραφή παραγγελίας κατόπιν ειδοποίησης από το χρήστη κ.α.

Για να έχουμε αυτό το αποτέλεσμα χρησιμοποιούνται οι παρακάτω τεχνικές και εργαλεία: Microsoft Visual Studio 2010, Photoshop, CSS , Sql Server Management Studio, HTML κ.α.

Πίνακας Περιεχομένων

Ευχαριστίες.....	ii
Abstract.....	iii
Σύνοψη.....	iv
Πίνακας Περιεχομένων.....	v
Πίνακας Εικόνων.....	vii
1.....	Εισαγωγή
1	
1.1 Περίληψη.....	1
1.2 Σκοπός και Στόχοι Εργασίας.....	1
1.3 Απαιτήσεις Συστήματος.....	1
1.4 Δυναμικές και Στατικές Ιστοσελίδες.....	3
2.....	Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης
4	
2.1 Η πλατφόρμα ASP.NET.....	4
2.2 Αρχιτεκτονική της ASP.NET.....	4
2.3 Αρχιτεκτονική του Visual Studio.NET.....	5
2.4 Πλεονεκτήματα.....	5
2.5 Visual Basic .NET.....	6
3.....	Βάση Δεδομένων
7	
3.1 Βάση Δεδομένων και Sql Server.....	7
3.2 Sql Server Management Studio.....	8
3.3 Πίνακες της βάσης δεδομένων.....	10
3.4 Λίγα Λόγια για τις Stored Procedures στην Database.....	16
4.....	Ανάλυση διαδικασιών σε ASP.NET
18	
4.1 Επικύρωση για τις Web φόρμες.....	18
4.1.1 RequiredFieldValidator.....	18
4.1.2 RegularExpressionValidator.....	19
4.2 CSS (Cascading Style Sheet).....	19
4.2.1 Τι είναι το CSS.....	19
4.2.2 Σύνταξη.....	19
4.3 Ασφάλεια ASP.NET και IIS.....	20
4.3.1 Ροή ασφαλείας με μια αίτηση.....	20
4.3.2 Έλεγχος ταυτότητας.....	21
4.3.3 web.config.....	21
4.4 Cookies.....	22
4.5 Σύνδεση με βάση δεδομένων.....	22
5.....	Η εφαρμογή μας
24	
5.1 Ecenter.Master.....	24
5.2 Default.aspx.....	24
5.3 Desktop.aspx.....	25
5.4 Hardware.aspx.....	26
5.5 Search.aspx.....	27
5.6 Product.aspx.....	27
5.7 Login.aspx.....	28
5.8 Register.aspx.....	29
5.9 AccInfo.aspx.....	31
5.10 ViewCart.aspx.....	32
5.11 HowToPay.aspx-HowToSend.aspx & SecurePay.aspx.....	32
5.12 Admin.aspx.....	34

5.13	Buy.aspx	35
6.Βιβλιογραφία - Πηγές	
	36	
7.Παράρτημα	
	37	
7.1	Ecenter.master.....	37
7.2	Default.aspx	41
7.3	Desktop.aspx.....	41
7.4	Hardware.aspx	47
7.5	Search.aspx	52
7.6	Product.aspx.....	53
7.7	Login.aspx.....	56
7.8	Register.aspx.....	58
7.9	AccInfo.aspx	63
7.10	ViewCart.aspx.....	66
7.11	Admin.aspx	71
7.12	Buy.aspx	77

Πίνακας Εικόνων

Εικόνα 1: Αρχιτεκτονική ASP.NET	4
Εικόνα 2: Αρχιτεκτονική Visual Studio.NET.....	5
Εικόνα 3: Management Studio Authentication.....	9
Εικόνα 4: Management Studio Σελίδα εκκίνησης.....	9
Εικόνα 5: Management Studio Διάγραμμα.....	10
Εικόνα 6: Διάγραμμα Βάσης Δεδομένων	10
Εικόνα 7: Πίνακας tblCategory	11
Εικόνα 8: Πίνακας tblSubCategory	11
Εικόνα 9: Πίνακας tblManufacturer	12
Εικόνα 10: Πίνακας tblProduct.....	12
Εικόνα 11: Πίνακας tblSubCategoryCharacteristics.....	13
Εικόνα 12: Πίνακας tblProductCharacteristics	13
Εικόνα 13: Πίνακας tblProductOrder	14
Εικόνα 14: Πίνακας tblOrder.....	14
Εικόνα 15: Πίνακας tblCustomer.....	15
Εικόνα 16: Πίνακας tblInvoice	15
Εικόνα 17: Στοιχεία Ελέγχου Επικύρωσης.....	18
Εικόνα 18: Default.aspx.....	24
Εικόνα 19: Χρήστης on-line – Αποσύνδεση.....	25
Εικόνα 20: Desktop.aspx	25
Εικόνα 21: Hardware.aspx.....	26
Εικόνα 22: Hardware.aspx - Φίλτρα CPU	26
Εικόνα 23: Hardware.aspx - Φίλτρα RAM.....	27
Εικόνα 24: Search.aspx.....	27
Εικόνα 25: Product.aspx	28
Εικόνα 26: Login.aspx.....	28
Εικόνα 27: Register.aspx - Φόρμα εγγραφής.....	29
Εικόνα 28: Register.aspx - Φόρμα Τιμολογίου	29
Εικόνα 29: Register.aspx – RequiredFieldValidator	30
Εικόνα 30: AccInfo.aspx	31
Εικόνα 31: Register.aspx - Αλλαγή Στοιχείων	31
Εικόνα 32: Καλάθι Αγορών.....	32
Εικόνα 33: Τρόποι Πληρωμής.....	33
Εικόνα 34: Τρόποι Αποστολής.....	33
Εικόνα 35: Ασφάλεια Συναλλαγών	34
Εικόνα 36: Admin.aspx	34
Εικόνα 37: Buy.aspx.....	35

1. Εισαγωγή

1.1 Περίληψη

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας πλατφόρμας ηλεκτρονικού εμπορίου. Πιο συγκεκριμένα, η κατασκευή ενός ηλεκτρονικού καταστήματος, στο οποίο ο πελάτης μπορεί να πραγματοποιήσει τις αγορές του μέσω του διαδικτύου.

Το ηλεκτρονικό εμπόριο περιλαμβάνει την ψηφιοποίηση των αγορών: αγορά προϊόντων ή υπηρεσιών μέσω ηλεκτρονικών συστημάτων, όπως είναι το Διαδίκτυο. Περιλαμβάνει, εκτός από τις τυπικές αγορές μέσω του Παγκόσμιου Ιστού, και όλη τη βιομηχανία που υποστηρίζει αυτές τις αγορές: online επεξεργασία δοσοληψιών (για τράπεζες κυρίως), διαχείριση αλυσίδας παροχών, ανταλλαγή ηλεκτρονικών δεδομένων, κ.ά. Το ηλεκτρονικό εμπόριο τα τελευταία χρόνια έχει αναπτυχθεί σημαντικά και έχει κερδίσει την εμπιστοσύνη των πελατών. Πουλώντας on-line, ο έμπορος μειώνει ή και καταργεί τη δική του αλυσίδα διανομής εμπορευμάτων, ελαχιστοποιώντας πάγια έξοδα, όπως ενοίκια ή προσωπικό. Ο πελάτης μπορεί να κάνει έρευνα αγοράς εύκολα και γρήγορα ανά πάσα στιγμή και από οποιαδήποτε τοποθεσία βρίσκεται.

Το πρακτικό τμήμα της πτυχιακής εργασίας έχει αναπτυχθεί σε τεχνολογία ASP.NET. Η Microsoft έφερε επανάσταση στο web-application για την ανάπτυξη με την αρχική έκδοση του ASP.NET, αυξάνοντας την παραγωγικότητα του έργου σε ασύγκριτο επίπεδο με οποιαδήποτε άλλη τεχνολογία. Οι εργασίες σε ASP.NET παίρνουν λιγότερο χρόνο για να κτιστούν, είναι πιο εύκολο να διατηρηθούν, έχουν απλοποιημένη επαναχρησιμοποίηση του κώδικα και έχουν τη μεγαλύτερη απόδοση απ' ότι σε Java, Perl, PHP, Cold Fusion, και ASP 3.0.

Η βάση δεδομένων έχει υλοποιηθεί στον Sql Server 2008. Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα. Ο SQL Server είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server έχει υπηρεσίες Reporting για στοιχεία που συγκεντρώνονται από τις βάσεις δεδομένων SQL Server. Με αυτόν τον τρόπο είναι δυνατό να παραχθεί μια έκθεση Ιστού (Web Report) σχεδόν αυτόματα χωρίς προγραμματισμό HTML, CSS ή JavaScript, μέσα από το Visual Studio.

1.2 Σκοπός και Στόχοι Εργασίας

Στόχος της παρούσας πτυχιακής εργασίας είναι η δημιουργία μιας διαδικτυακής εφαρμογής (web application) με λειτουργίες όπως η εγγραφή χρήστη, αναζήτηση και προβολή των προϊόντων, διαχωρισμό των προϊόντων σε κατηγορίες, περιορισμό στην αναζήτηση με πολλαπλά φίλτρα, προσθήκη προϊόντων στο καλάθι αγορών, την αγορά των προϊόντων από τους χρήστες και τη διαχείριση των δεδομένων της βάσης από τους διαχειριστές.

Σκοπός μας είναι η απόκτηση γνώσεων πάνω σε θέματα όπως ανάπτυξη και σχεδιασμό βάσης δεδομένων με χρήση Sql Server, κατασκευή ιστοσελίδας και διεπαφής χρήστη με χρήση ASP.NET, Visual Basic, CSS και HTML χρησιμοποιώντας τα προγράμματα Microsoft Visual Studio 2010 και Sql Server Management Studio.

1.3 Απαιτήσεις Συστήματος

Οι απαιτούμενες τεχνικές προδιαγραφές για την υλοποίηση της πλατφόρμας είναι οι εξής:

- Για την πλατφόρμα .NET Framework απαιτείται ένα από τα παρακάτω λειτουργικά συστήματα:
 - Windows 7 Ultimate x86
 - Windows 7 Ultimate N

- Windows 7 Ultimate x64
- Windows 7 Enterprise x86
- Windows 7 Enterprise N
- Windows 7 Enterprise x64
- Windows 7 Professional x86
- Windows 7 Professional N
- Windows 7 Professional x64
- Windows 7 Home Premium x86
- Windows 7 Home Premium N
- Windows 7 Home Premium x64
- Windows 7 Home Basic x86
- Windows 7 Home Basic N
- Windows 7 Starter x86
- Windows 7 Starter N
- Windows Vista R2 Enterprise (x64)
- Windows Vista R2 Enterprise
- Windows Vista R2 Business (x64)
- Windows Vista R2 Business
- Windows Vista R2 Ultimate (x64)
- Windows Vista R2 Ultimate
- Windows Vista R2 Home Premium (x64)
- Windows Vista R2 Home Premium
- Windows Vista R2 Home Basic
- Windows Vista R2 Starter Ed. Digital Boost (x64)
- Windows Vista R2 Starter Ed. Digital Boost
- Windows Vista R2 Starter
- Windows Vista Ultimate
- Windows Vista Ultimate x64 Edition
- Windows Vista Enterprise
- Windows Vista Enterprise x64 Edition
- Windows Vista Business
- Windows Vista Business x64 Edition
- Windows Vista Home Premium
- Windows Vista Home Premium x64 Edition
- Windows Vista Home Basic
- Windows Vista Starter Ed. Digital Boost
- Windows Vista Starter Ed. Digital Boost (x64)
- Windows Vista Starter
- Windows XP Professional
- Windows XP Professional x64 Edition
- Windows XP Home Edition

➤ Για το SQL Server 2008 ένα από τα παρακάτω:

- Windows XP Professional x64
- Windows Server 2003 SP2 64-bit x64 Standard
- Windows Server 2003 SP2 64-bit x64 Datacenter
- Windows Server 2003 SP2 64-bit x64 Enterprise
- Windows Vista Ultimate x64
- Windows Vista Enterprise x64
- Windows Vista Business x64
- Windows Server 2008 x64 Web
- Windows Server 2008 x64 Standard, Windows Server 2008 x64 Standard without

Hyper-V

- Windows Server 2008 x64 Datacenter, Windows Server 2008 x64 Datacenter without

Hyper-V

- Windows Server 2008 x64 Enterprise, Windows Server 2008 x64 Enterprise without Hyper-V
 - Windows Small Business Server 2008
 - Windows Server 2008 for Windows Essential Server Solutions
 - Windows Server 2008 without Hyper-V for Windows Essential Server Solutions
 - Windows 7 64-bit x64 Ultimate
 - Windows 7 64-bit x64 Enterprise
 - Windows 7 64-bit x64 Professional
 - Windows 2008 R2 64-bit x64 Web
 - Windows 2008 R2 Foundation Server
 - Windows 2008 R2 64-bit x64 Standard
 - Windows 2008 R2 64-bit x64 Enterprise
 - Windows 2008 R2 64-bit x64 Datacenter
- Προδιαγραφές υλικού:
- Επεξεργαστής: 1.4 GHz (το ελάχιστο), 2 GHz (προτεινόμενο)
 - Μνήμη: 512 MB (το ελάχιστο), 2048 MB (προτεινόμενο)
 - Σκληρός δίσκος: 850 MB (για 32-bit λειτουργικό σύστημα), 2 GB (για 64-bit λειτουργικό σύστημα)

1.4 Δυναμικές και Στατικές Ιστοσελίδες

Με την ραγδαία άνθηση του διαδικτύου , την ταχεία χρήση του ως βασικό μέσο μαζικής επικοινωνίας, είναι προφανές ότι κάθε άτομο η εταιρία που θέλει να προβληθεί η να παρουσιάσει τα προϊόντα μιας επιχείρησης ή να προβάλει ιδέες γίνεται μέσω μιας ιστοσελίδας. Τα είδη μιας τέτοιας ιστοσελίδας είναι πολλά και ποικίλουν ανάλογα τις ανάγκες του καθενός. Οι δυο μεγαλύτερες και βασικότερες κατηγορίες ιστοσελίδων είναι οι στατικές (static) και οι δυναμικές (dynamic).

Στατικές ονομάζονται οι ιστοσελίδες που έχουν μόνιμο περιεχόμενο (αυτό δεν σημαίνει ότι το περιεχόμενο τους είναι στατικό , αντιθέτως μπορεί να έχει πολλά κινούμενα γραφικά). Δυναμικές είναι οι ιστοσελίδες στις οποίες ο χρήστης αλληλεπιδρά με την ίδια την σελίδα. Σήμερα κυριαρχεί η άποψη ότι οι στατικές ιστοσελίδες είναι ξεπερασμένες κυρίως επειδή για να κατασκευαστούν βασίζονται σε παλαιότερες τεχνολογίες. Αυτή η εντύπωση όμως είναι λανθασμένη. Μερικές από τις πιο μοντέρνες ιστοσελίδες είναι στατικές, ενώ οι στατικές ιστοσελίδες προσφέρουν διάφορα πλεονεκτήματα που δεν έχουν οι δυναμικές . Ο ορός «στατικές» αναφέρεται στο περιεχόμενο και όχι σε κάποιο είδος κίνησης. Τα δεδομένα για να αλλάξουν σε μια στατική ιστοσελίδα θα πρέπει ο διαχειριστής να τα αλλάξει χειροκίνητα μέσω κάποιου προγράμματος κατασκευής ιστοσελίδων .

Αντίθετα στις δυναμικές ιστοσελίδες ο χρήστης αλληλεπιδρά άμεσα με την ίδια την σελίδα και αυτό σημαίνει ότι μπορεί να αλλάξει το περιεχόμενο της. Για αυτόν τον λόγο, οι δυναμικές ιστοσελίδες από open source CMS (content management system) , ότι δηλαδή το σύστημα διαχείρισης της ιστοσελίδας είναι ανοιχτό σε όλους τους χρηστές , οι όποιοι μπορούν να το αλλάξουν η να το τροποποιήσουν. Το κατά πόσο έχει το δικαίωμα ο χρήστης να αλλάξει το περιεχόμενο μιας δυναμικής ιστοσελίδας το ορίζει ο υπερδιαχειριστής της σελίδας (administrator).

Για την ανάπτυξη και σχεδίαση δυναμικών ή στατικών ιστοσελίδων χρησιμοποιούνται πολλές γλώσσες. Η html χρησιμοποιείται ευρέως στις στατικές ιστοσελίδες και η php στις δυναμικές. Στις δυναμικές μπορεί όμως να χρησιμοποιηθεί και html γιατί η html δεν είναι γλώσσα προγραμματισμού σε αντίθεση με την php όπου είναι.

2. Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης

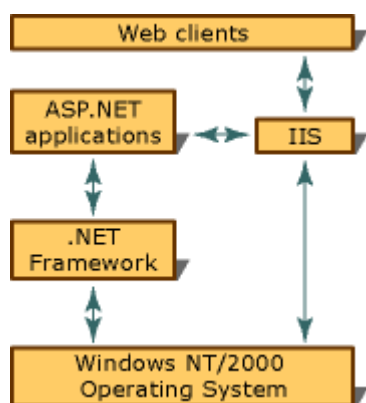
2.1 Η πλατφόρμα ASP.NET

Η ASP.NET είναι πλαίσιο διαδικτυακής εφαρμογής ανεπτυγμένη και διατεθειμένη στο εμπόριο από τη Microsoft, για να επιτρέπει στους προγραμματιστές τη δημιουργία δυναμικών ιστοσελίδων, Web εφαρμογών και υπηρεσίες Web. Κυκλοφόρησε για πρώτη φορά τον Ιανουάριο του 2002 με την έκδοση 1.0 του .NET Framework και είναι η διάδοχος της τεχνολογίας Active Server Pages (ASP).

Οι Web εφαρμογές φτιάχνονται χρησιμοποιώντας Web Forms. Η ASP.NET έρχεται με έτοιμα Web Form controls, που είναι υπαίτια για τη δημιουργία διεπαφής χρήστη. Κατοπτρίζουν τυπικά HTML widgets, όπως πεδία κειμένου ή κουμπιά. Επίσης, δίνει τη δυνατότητα δημιουργίας προσωπικών user controls. Τα Web Forms είναι σχεδιασμένα να κάνουν το κτίσιμο των web εφαρμογών τόσο εύκολο όσο και των απλών εφαρμογών Visual Basic.

2.2 Αρχιτεκτονική της ASP.NET

Σε αυτήν την ενότητα θα δούμε μια επισκόπηση της υποδομής της ASP.NET και τις σχέσεις του υποσυστήματος, καθώς σχετίζονται με το θέμα τις ασφάλειας. Στην εικόνα που ακολουθεί φαίνονται οι σχέσεις μεταξύ των συστημάτων ασφάλειας στην ASP.NET.



Εικόνα 1: Αρχιτεκτονική ASP.NET

Όπως δείχνει η εικόνα, όλοι οι Web Clients επικοινωνούν με τις ASP.NET εφαρμογές μέσω του Microsoft Internet Information Services (IIS). Το IIS αποκρυπτογραφεί και προαιρετικά επικυρώνει το αίτημα. Αν υπάρχει η επιλογή Allow Anonymous, δε γίνεται έλεγχος ταυτότητας. Το IIS βρίσκει την πηγή που ζητήθηκε (όπως μια εφαρμογή ASP.NET) και αν ο πελάτης έχει λάβει αδειά, επιστρέφει την κατάλληλη πηγή.

Εκτός από τα ενσωματωμένα χαρακτηριστικά ασφάλειας, μια εφαρμογή ASP.NET μπορεί να χρησιμοποιήσει τα χαμηλού επιπέδου χαρακτηριστικά ασφάλειας του .Net Framework.

Στην ASP.NET ο compiler μεταγλωττίζει γλώσσες όπως οι Visual Basic, C++, C# στο Visual Studio και στη συνέχεια δημιουργείται ένας ενδιάμεσος κώδικας, MSIL (Microsoft Intermediate Language). Ο κώδικας αυτός βρίσκεται σε υψηλότερο επίπεδο από τον κώδικα μηχανής και σε αυτόν περιέχονται metadata. Αυτά χρησιμοποιούνται από το CLR για την φόρτωση συγκεκριμένων λειτουργιών.

Το θετικό σε αυτή τη λειτουργία είναι ότι προσφέρεται ένας ενιαίος τρόπος για την αξιοποίηση των υπηρεσιών του λειτουργικού συστήματος. Άλλες υπηρεσίες βρίσκονται σε αρχεία DLL και άλλες σε αρχεία COM. Αποτέλεσμα είναι να υπάρχει πλέον διευκόλυνση, εφ' όσον η εγκατάσταση μιας εφαρμογής σήμαινε την αντικατάσταση ενός αρχείου DLL μιας άλλης εφαρμογής με απρόβλεπτα αποτελέσματα.

2.3 Αρχιτεκτονική του Visual Studio.NET

Η ASP.NET είναι βασισμένη στη θεμελιώδη αρχιτεκτονική του .NET Framework. Το Visual Studio προσφέρει ένα ομοιόμορφο τρόπο να συνδυαστούν τα διάφορα χαρακτηριστικά αυτής της αρχιτεκτονικής.



Εικόνα 2: Αρχιτεκτονική Visual Studio.NET

Στο κάτω μέρος της αρχιτεκτονικής είναι το Common Language Runtime. Η ASP είναι κτισμένη σε Common Language Runtime (CLR), που επιτρέπει στους προγραμματιστές να γράψουν κώδικα χρησιμοποιώντας οποιαδήποτε γλώσσα προγραμματισμού υποστηρίζει η .NET.

Το .NET Framework προσφέρει πλούσια σειρά βιβλιοθηκών. Αυτές περιλαμβάνουν τις βασικές κλάσεις, όπως δικτύωσης και κλάσεις εισόδου/εξόδου. Όλες οι κλάσεις συγκεντρώνονται από το Services Framework, το οποίο είναι ένα στάδιο πάνω από το CLR.

Το ADO.NET (ActiveX Data Object) είναι μοντέλο της Microsoft για το .NET Framework. Δεν είναι απλά η μεταφορά του γνωστού ADO μοντέλου σε περιβάλλον διαχειρισμού, αλλά ένα τελείως καινούριο παράδειγμα για πρόσβαση σε δεδομένα.

Το τέταρτο επίπεδο του framework αποτελείται από το μοντέλο εφαρμογής Windows και παράλληλα το μοντέλο εφαρμογής Web. Οι υπηρεσίες Web είναι λύσεις λογισμικού διαμοιρασμένες από το διαδίκτυο σε οποιαδήποτε συσκευή.

Το CLS είναι μια συλλογή από κανόνες που περιορίζουν την κάθε γλώσσα προγραμματισμού και πρέπει να τους ακολουθήσει, έτσι ώστε να μην έχουμε γλώσσες που τρέχουν και να δημιουργούν δικές τους παραμέτρους και νέους τύπους δεδομένων.

Το CLR και το .NET Framework είναι σχεδιασμένα με τέτοιο τρόπο που ο κώδικας που έχει γραφτεί σε μια γλώσσα να μην μπορεί απρόσκοπτα να χρησιμοποιηθεί από άλλη γλώσσα.

2.4 Πλεονεκτήματα

Η ASP.NET είναι κάτι περισσότερο από την επόμενη γενιά της κλασικής ASP. Παρέχει ένα εντελώς καινούριο μοντέλο προγραμματισμού για τη δημιουργία διαδικτυακών εφαρμογών.

- Βελτιωμένες επιδόσεις και επεκτασιμότητα
 - Compiled Execution: Η ASP.NET είναι πολύ πιο γρήγορη από την κλασική ASP, καθώς διατηρεί το «just hit save» ενημερωμένο μοντέλο της ASP. Δεν χρειάζεται το βήμα explicit compile.
 - Rich Output Caching: Όταν το output caching είναι ενεργοποιημένο σε μια σελίδα, η ASP.NET εκτελεί τη σελίδα μια φορά και αποθηκεύει το αποτέλεσμα στη μνήμη πριν

- ο το στείλει στο χρήστη. Αν κάποιος άλλος χρήστης ζητήσει την ίδια σελίδα, η ASP.NET στέλνει τη σελίδα από την κρυφή μνήμη χωρίς να την «ξανατρέχει».
 - ο Web Farm Session State: Το session state της ASP.NET μας επιτρέπει να διαμοιραζόμαστε δεδομένα συνεδρίας σε όλες τις μηχανές σε μια «δικτυακή φάρμα»
- Βελτιωμένη αξιοπιστία
 - ο Διαρροή μνήμης, αδιέξοδα και σπάσιμο προστασίας: η ASP.NET αυτόματα ανιχνεύει και ανακτά από λάθη όπως διαρροή μνήμης, αδιέξοδα.
- Εύκολη εγκατάσταση
 - ο "No Touch" εγκατάσταση εφαρμογής: Με την ASP.NET μπορούμε να εγκαταστήσουμε ολόκληρη εφαρμογή αντιγράφοντας τη στο Server. Οι ρυθμίσεις διαμόρφωσης είναι αποθηκευμένες σε ένα XML αρχείο μέσα στην εφαρμογή.
- Παραγωγικότητα προγραμματιστή
 - ο Εύκολο μοντέλο προγραμματισμού: η ASP.NET κάνει το κτίσιμο πραγματικών εφαρμογών διαδικτύου δραματικά ευκολότερο με server controls που μας επιτρέπουν να κτίσουμε μεγάλες σελίδες με λιγότερο κώδικα από την κλασική ASP.
 - ο Ευέλικτες επιλογές γλώσσας: Η ASP.NET δεν υποστηρίζει μόνο τη Microsoft Visual Basic Scripting Edition (VBScript) και τη Microsoft Jscript, αλλά περισσότερες από 25 γλώσσες .NET, περιλαμβάνοντας ενσωματωμένη υποστήριξη για Visual Basic .NET, Microsoft C#, και Jscript .NET
 - ο Πλούσιες κλάσεις Framework: Η βιβλιοθήκη κλάσεων του .NET framework προσφέρει πάνω από 4500 κλάσεις που ενθυλακώνουν πλούσια λειτουργικότητα όπως XML, data access, file upload, regular expressions, δημιουργία εικόνων, παρακολούθηση και καταγραφή επιδόσεων κα.

2.5 Visual Basic .NET

Η Visual Basic .NET αναπτύχθηκε από την BASIC (Beginner's All-Purpose Symbolic Instruction Code), που είχε αναπτυχθεί στα μέσα της δεκαετίας του '60 από τους John Kemeny και Thomas Kurtz, ως μια γλώσσα για την εγγραφή απλών προγραμμάτων. Όταν ο Bill Gates ίδρυσε την Microsoft Corporation, υλοποίησε την BASIC σε διάφορους από τους πρώτους προσωπικούς υπολογιστές. Με την ανάπτυξη του γραφικού περιβάλλοντος χρήστη Microsoft Windows στα τέλη της δεκαετίας του '80, η φυσική εξέλιξη της BASIC ήταν η Visual Basic, που εισήχθη από την Microsoft το 1991.

Η Visual Basic είναι γλώσσα τρίτης γενιάς οδηγούμενη από συμβάντα (event driven) και έχει ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για το μοντέλο προγραμματισμού COM. Θεωρείται σχετικά εύκολη γλώσσα προγραμματισμού στην εκμάθηση και τη χρησιμοποίηση καθώς έχει γραφικό περιβάλλον χρήστη. Επιτρέπει την πρόσβαση σε βάσεις δεδομένων χρησιμοποιώντας αντικείμενα (Data Access Objects, Remote Data Objects ή ActiveX Data Objects) και τη δημιουργία στοιχείων έλεγχου ActiveX και αντικειμένων. Η νέα έκδοση της VB .NET μας δίνει τη δυνατότητα να δημιουργούμε κλάσεις, κληρονομικότητα, overloading και όλα τα άλλα στοιχεία αντικειμενοστρέφειας.

Η Visual Basic έχει τα εξής χαρακτηριστικά που διαφέρουν από τις γλώσσες που προέρχονται από τη C:

- ο Η πολλαπλή ανάθεση που διατίθεται στη C δεν είναι δυνατή. Το $\alpha = \beta = \gamma$ δε σημαίνει ότι οι τιμές των α, β, γ είναι ίδιες. Το αποτέλεσμα της Boolean $\beta = \gamma$ αποθηκεύεται στη μεταβλητή α .
- ο Οι λογικοί και δυαδικοί τελεστές είναι ενοποιημένοι.

3. Βάση Δεδομένων

3.1 Βάση Δεδομένων και Sql Server

Με τον όρο **βάση δεδομένων** εννοείται μία συλλογή από *συστηματικά οργανωμένα* (formatted) σχετιζόμενα δεδομένα. Ένας τηλεφωνικός κατάλογος, για παράδειγμα, θεωρείται βάση δεδομένων, καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου. Ωστόσο, στον κόσμο των υπολογιστών, με τον όρο βάση δεδομένων αναφερόμαστε σε μια συλλογή σχετιζόμενων δεδομένων τμημάτων πληροφορίας ηλεκτρονικά αποθηκευμένων. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει βάσει του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων της σε προγράμματα ή συλλογές προγραμμάτων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων.

Μια βάση δεδομένων είναι ένα εργαλείο συλλογής και οργάνωσης πληροφοριών. Οι βάσεις δεδομένων μπορούν να αποθηκεύουν πληροφορίες για άτομα, προϊόντα, παραγγελίες ή οτιδήποτε άλλο. Πολλές βάσεις δεδομένων ξεκινούν ως μια λίστα σε κάποιο πρόγραμμα επεξεργασίας κειμένου ή ένα υπολογιστικό φύλλο. Καθώς η λίστα γίνεται ολοένα μεγαλύτερη, στα δεδομένα αρχίζουν να εμφανίζονται επαναλήψεις και ανακρίβειες. Η κατανόηση των δεδομένων αυτών σε μορφή λίστας, καθίσταται δυσκολότερη και οι τρόποι αναζήτησης και εξαγωγής υποσυνόλων των δεδομένων για αναθεώρηση είναι πραγματικά πολύ περιορισμένοι. Με την εμφάνιση αυτών των προβλημάτων, συνιστάται να μεταφέρετε τα δεδομένα σας σε μια βάση δεδομένων που έχει δημιουργηθεί σε ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS). Μια βάση δεδομένων που έχει δημιουργηθεί σε ένα υπολογιστικό σύστημα είναι ένα κοντέινερ αντικειμένων και μπορεί να περιέχει περισσότερους από έναν πίνακες. Για παράδειγμα, ένα σύστημα παρακολούθησης απογραφής το οποίο χρησιμοποιεί τρεις πίνακες δεν θεωρείται ότι είναι τρεις βάσεις δεδομένων, αλλά μία βάση δεδομένων που περιέχει τρεις πίνακες. Εκτός και αν έχει σχεδιαστεί ειδικά ώστε να χρησιμοποιεί δεδομένα ή κώδικα από μια άλλη πηγή. Η ηλεκτρονική βάση δεδομένων χρησιμοποιεί ιδιαίτερου τύπου λογισμικό προκειμένου να οργανώσει την αποθήκευση των δεδομένων της. Το διακριτό αυτό λογισμικό είναι γνωστό ως Σύστημα διαχείρισης βάσης δεδομένων συντομευμένα (DBMS).

Η γλώσσα SQL (Structured Query Language) είναι η πιο διαδομένη γλώσσα που χρησιμοποιείται για την με επικοινωνία του χρήστη με σχεσιακές Βάσεις Δεδομένων. Πρόκειται για μια γλώσσα μη διαδικαστική στην οποία ο χρήστης διατυπώνει διάφορα αιτήματα και το DBMS αναλαμβάνει να ικανοποιήσει. Η SQL δίνει την δυνατότητα στο χρήστη να δημιουργήσει, να τροποποιήσει τους πίνακες της βάσης καθώς και να αναζητά πληροφορίες από την βάση εφαρμόζοντας διάφορα κριτήρια. Η γνώση της SQL είναι απαραίτητη ακόμη και αν χρησιμοποιούμε ένα DBMS με γραφικό περιβάλλον καθώς πολλά συστήματα της μορφής πελάτη-Εξυπηρετητή έχουν SQL Server. Τα περισσότερα σχεσιακά DBMS χρησιμοποιούν το πρότυπο ANSI / ISO SQL-92 υπάρχει όμως και το νέο πρότυπο ANSI / ISO SQL-99 που χρησιμοποιείται κυρίως από αντικειμενοστραφή συστήματα διαχείρισης.

Ο **SQL Server** είναι μια σχεσιακή βάση δεδομένων, η οποία αναπτύσσεται από τη Microsoft. Οι κύριες γλώσσες που χρησιμοποιούνται είναι η T-SQL και η ANSI SQL. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase.

Η κύρια μονάδα αποθήκευσης στοιχείων είναι μια βάση δεδομένων, η οποία αποτελείται από μια συλλογή πινάκων και κώδικα.

Η κεντρική βάση δεδομένων του SQL υποστηρίζει διαφορετικούς τύπους, συμπεριλαμβανομένων των ακεραίων αριθμών, αριθμών κινητής υποδιαστολής, δεκαδικών, αλφαριθμητικών, Varchar (σειρές χαρακτήρων μεταβλητού μήκους), δυαδικών αριθμών (για τα μη δομημένα δεδομένα), κειμένων (για κείμενα).

Επιτρέπει επίσης καθορισμένους από το χρήστη σύνθετους τύπους δεδομένων (UDTs), δηλαδή τύπους που βασίζονται στους βασικούς τύπους αλλά μπορούν να τροποποιηθούν. Τα στοιχεία στη βάση δεδομένων αποθηκεύονται σε ένα (ή περισσότερα) αρχεία με επέκταση .mdf.

Τα δευτεροβάθμια στοιχεία αποθηκεύονται στο αρχείο με επέκταση .ndf. Το αρχείο καταγραφής το οποίο περιέχει όλες τις πρόσφατες αλλαγές στη βάση δεδομένων αποθηκεύεται σε αρχείο με επέκταση .ldf. Η ερώτηση (query) είναι ο κύριος τρόπος για την ανάκτηση στοιχείων από μια βάση δεδομένων. Η ερώτηση εκφράζεται χρησιμοποιώντας μια παραλλαγή της αποκαλούμενου SQL T-SQL, είναι μια διάλεκτος SQL που αναπτύχθηκε από την Microsoft και Sybase. Η T-SQL είναι πολύ κοντά στα ANSI standards που έχουν καθιερωθεί διεθνώς, σε αντιδιαστολή με άλλες διαλέκτους όπως η PL-SQL της Oracle που διαφέρουν περισσότερο από το ANSI standards. Η ερώτηση διευκρινίζει επεξηγηματικά αυτό που είναι να ανακτηθεί από την βάση δεδομένων. Η ερώτηση υποβάλλεται σε επεξεργασία από τον **επεξεργαστή ερώτησης**, ο οποίος υπολογίζει τη σειρά των βημάτων που θα είναι απαραίτητα για να ανακτηθούν τα ζητούμενα στοιχεία. Η ακολουθία ενεργειών απαραίτητων να εκτελέσουν μια ερώτηση καλείται σχέδιο ερώτησης. Υπάρχουν πολλοί τρόποι για να υποβληθεί σε επεξεργασία η ίδια ερώτηση. Π.χ, μια ερώτηση που ενώνει 2 πίνακες A & B μαζί θα μπορούσε να γίνει ή πηγαίνοντας πρώτα στο A βρισκοντας τα στοιχεία και μετά ενώνοντας με τα στοιχεία του πίνακα B, ή πρώτα από το B και μετά πίνακα A. Ο SQL Server και όχι ο Προγραμματιστής παίρνει την απόφαση μέσω στατιστικών στοιχείων και άλλων στοιχείων που διαθέτει για τους 2 πίνακες να ακολουθήσει την διαδρομή A-B γιατί αυτός είναι ο γρηγορότερος τρόπος. Στις procedural γλώσσες προγραμματισμού πχ C, Pascal, .NET, ο προγραμματιστής θα πρέπει να κάνει την απόφαση το οποίο πολλές φορές σημαίνει λάθος απόφαση. Αυτό καλείται **βελτιστοποίηση ερώτησης** και εκτελείται από τον επεξεργαστή ερώτησης. Ο SQL παίρνει πολύ περίπλοκες αποφάσεις βασισμένος σε αλγόριθμους που έχουν αναπτυχθεί τα τελευταία 20-30 χρόνια πάνω στο αντικείμενο. Μια παρόμοια ερώτηση σε μια προηγούμενη έκδοση του SQL Server θα μπορούσε να πάρει 10-20 φορές περισσότερο χρόνο λόγω του διαφορετικού αλγόριθμου. Γενικά κάθε καινούργια έκδοση βελτιώνει την αποτελεσματικότητα, και άρα τον χρόνο εκτέλεσης. Εδώ ακριβώς είναι ένα από το πλεονεκτήματα προγραμματισμού ενός προβλήματος σε μια βάση δεδομένων σε σύγκριση με C, Pascal, C++ κτλ. Βέβαια όλα τα προβλήματα δεν είναι κατάλληλα για εφαρμογή σε βάση δεδομένων. Ο SQL Server επιτρέπει επίσης αποθηκευμένες ερωτήσεις (stored procedures). Οι αποθηκευμένες ερωτήσεις είναι ερωτήσεις με παραμέτρους T-SQL, οι οποίες αποθηκεύονται στον SQL Server. Μια stored procedure τυπικά είναι από 100-1000 γραμμές κώδικα SQL. Οι αποθηκευμένες ερωτήσεις μπορούν να δεχτούν τιμές που στέλνονται από τον πελάτη ως παράμετροι εισαγωγής, και να στείλουν τα αποτελέσματα ως παραμέτρους παραγωγής. Μπορούν να καλέσουν functions, και άλλες stored procedures. Οι stored procedures είναι γρηγορότερες από κατασκευή σε σχέση με απλές ακολουθίες SQL. Ο Microsoft SQL Server 2005 – 2008 περιλαμβάνει ένα module που λέγεται SQL CLR μέσω του οποίου ενσωματώνει το .NET μέσα στον SQL Server. Με το SQL CLR, οι αποθηκευμένες διαδικασίες μπορούν να γραφτούν σε οποιαδήποτε γλώσσα .NET συμπεριλαμβανομένου C# και VB.NET και να δημιουργήσουν μια stored procedure. Αυτό σημαίνει ότι ο SQL Server έχει όλες τις βιβλιοθήκες και πλεονεκτήματα του .NET, αυτόχθονα μέσα στο περιβάλλον του, τα οποία μπορεί να τα καλέσει οποιαδήποτε στιγμή.

3.2 Sql Server Management Studio

Το Sql server management studio είναι ένα δωρεάν εργαλείο που περιλαμβάνεται μέσα στο Microsoft Sql server 2005 και αργότερα το οποίο χρησιμοποιείται για να τροποποιείς, να διαχειρίζεσαι και οτιδήποτε άλλο χρειαστεί να κάνεις σε μια βάση η στα component της βάσης μέσα στον Sql server. Υπάρχουν βέβαια και κάποιοι περιορισμοί στην χρήση γιατί η Microsoft το έβγαλε δωρεάν. Αυτά τα εργαλεία συνήθως χρησιμοποιούνται από διαχειριστές βάσεων δεδομένων σε sql καθώς μπορούν να δουν γραφικό περιβάλλον όταν θα εγκαταστήσουν το πρόγραμμα και έτσι θα μπορούν με ευκολία να δουλέψουν πάνω στην βάση. Το ιδανικό σενάριο όταν χρησιμοποιείς τον sql server θα ήταν να έχουμε ένα μηχάνημα A και να θέλουμε να έχουμε πρόσβαση στην βάση δεδομένων από μηχάνημα B με το visual studio σαν να θέλαμε να δούμε τα δεδομένα πριν την χρήση. Τότε θα έπρεπε να εγκαταστήσουμε το sql management studio στο μηχάνημα B και να έχουμε πρόσβαση στα δεδομένα από κει. Το εργαλείο αυτό περιλαμβάνει και script editors και graphical tools τα όποια και τα 2 δουλεύουν με αντικείμενα του server.

Υπάρχουν πολλές διαφορετικές εκδόσεις διαθέσιμες από το site της Microsoft για αυτό το δωρεάν εργαλείο αλλά το sql server management studio 2008 είναι η καλύτερη επιλογή για κάποιον που θέλει να διαχειριστεί δεδομένα.

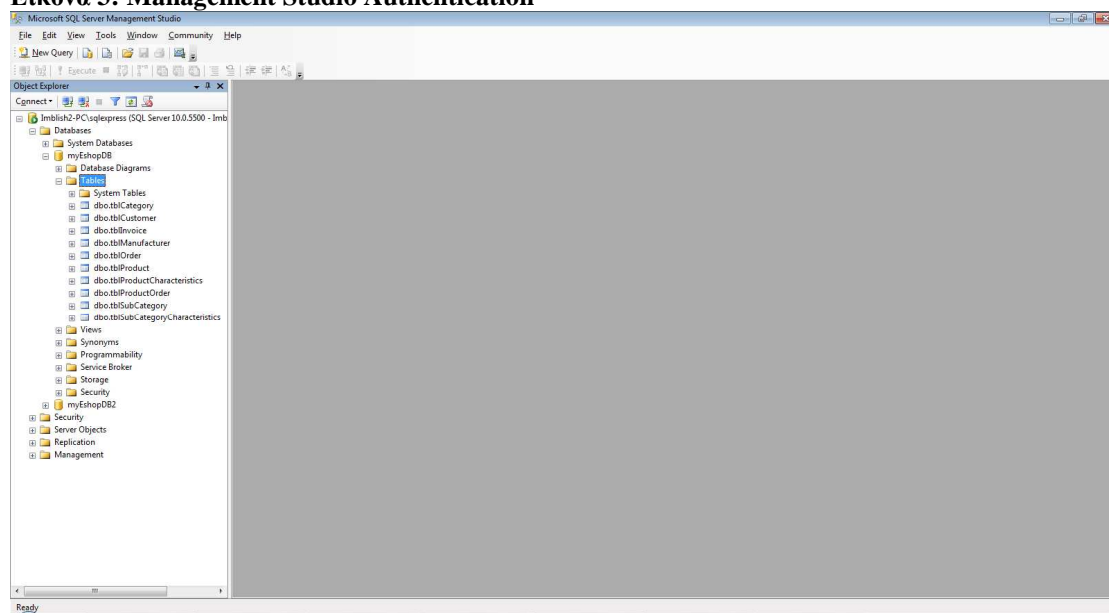
Ο Sql Server Management Studio είναι το βασικό εργαλείο που χρησιμοποιήσαμε για την διαχείριση της βάσης δεδομένων για την παρούσα πτυχιακή.

Ένα κεντρικό αντικείμενο του Sql server studio management studio είναι το object Explorer, το οποίο επιτρέπει στον χρήστη να πλοηγηθεί να επιλέξει και να επιδράσει πάνω σε οποιοδήποτε αντικείμενο μέσα στον server.

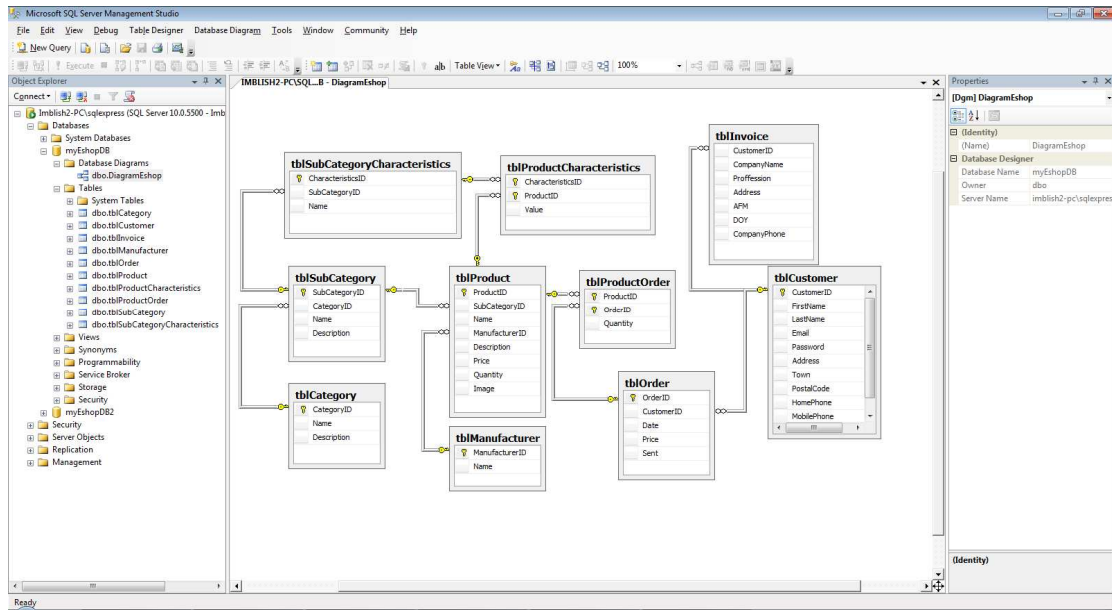
Μερικές εικόνες από την πλοήγηση στο πρόγραμμα:



Εικόνα 3: Management Studio Authentication



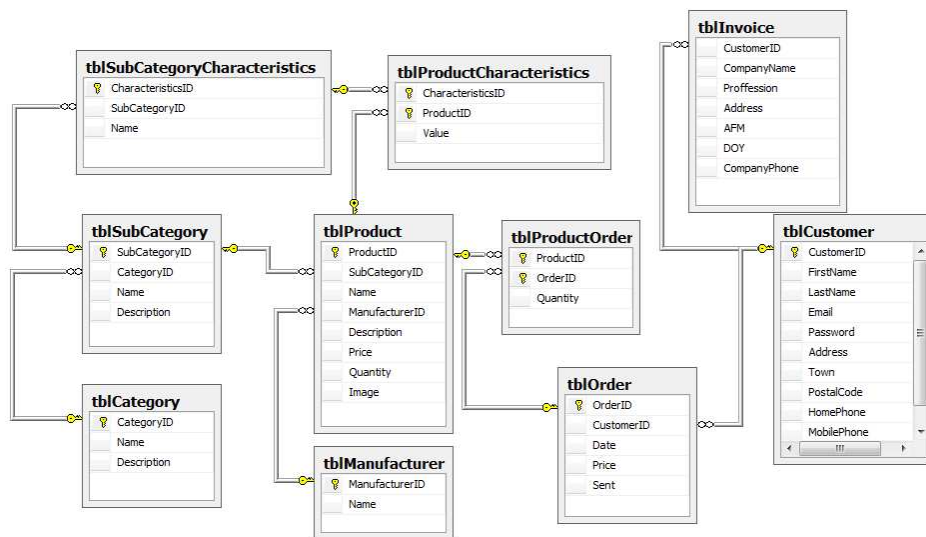
Εικόνα 4: Management Studio Σελίδα εκκίνησης



Εικόνα 5: Management Studio Διάγραμμα

3.3 Πίνακες της βάσης δεδομένων

Η βάση δεδομένων αποτελείται από τους πίνακες που φαίνονται παρακάτω (database Diagram) :



Εικόνα 6: Διάγραμμα Βάσης Δεδομένων

- Πινάκας dbo.tblCategory :

Column Name	Data Type	Allow Nulls
CategoryID	int	<input type="checkbox"/>
Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
Description	nvarchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εικόνα 7: Πινάκας tblCategory

Στον πίνακα dbo.tblCategory γίνεται ο διαχωρισμός των κατηγοριών των προϊόντων που προσφέρονται στο site μας παράδειγμα αν είναι Έτοιμος υπολογιστής (Desktop/Laptop) ή ένα κομμάτι Υλικού π.χ. μια κάρτα γραφικών (Hardware). Για να εισάγουμε μια κατηγορία στον πίνακα αυτό γίνεται μέσα από την καρτέλα Admin.aspx πατώντας το menu item “Category” και εισάγοντας ένα όνομα και μια Περιγραφή αυτό μας δημιουργεί μια νέα εγγραφή στον πίνακα με αυτά τα στοιχεία. Αυτός ο πίνακας συνδέεται με τον tblSubcategory μέσω του κλειδιού CategoryID του tblCategory και του κλειδιού CategoryID του tblSubcategory με επιλογές στα Delete Rule και Update Rule στο No Action.

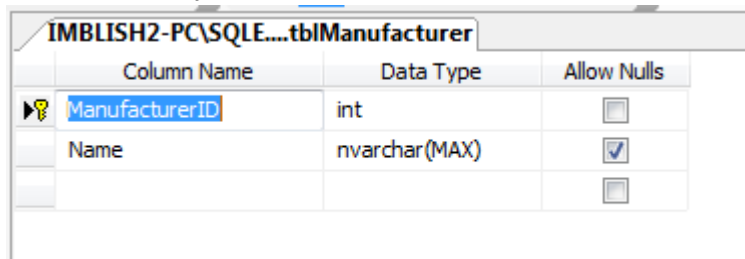
- Πινάκας dbo.tblSubCategory

Column Name	Data Type	Allow Nulls
SubCategoryID	int	<input type="checkbox"/>
CategoryID	int	<input checked="" type="checkbox"/>
Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
Description	nvarchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εικόνα 8: Πινάκας tblSubCategory

Στον πίνακα tblSubCategory διαχωρίζουμε ακόμα περισσότερο τα προϊόντα μας σε μικρότερες κατηγορίες π.χ. Οθόνες , πληκτρολόγια , Κάρτες ήχου για την κατηγορία Hardware ή Desktop,Laptop,Net book για την κατηγορία Desktop. Η εισαγωγή των στοιχείων γίνεται πάλι από την καρτέλα Admin.aspx αλλά αυτή την φορά από το menu item “SubCategory” και εισάγοντας ένα όνομα και μια περιγραφή καθώς επίσης και την Κατηγορία που ανήκει η συγκεκριμένη υποκατηγορία. Αυτός ο πίνακας συνδέεται με τον tblCategory , τον tblProduct μέσω του κλειδιού SubCategoryID με επιλογές στα Delete Rule και Update Rule σε No Action και Cascade αντίστοιχα και τον tblSubCategoryCharacteristics μέσω του κλειδιού SubCategoryID με επιλογές στα Delete Rule και Update Rule σε No Action και Cascade αντίστοιχα.

- Πινάκας `dbo.tblManufacturer`

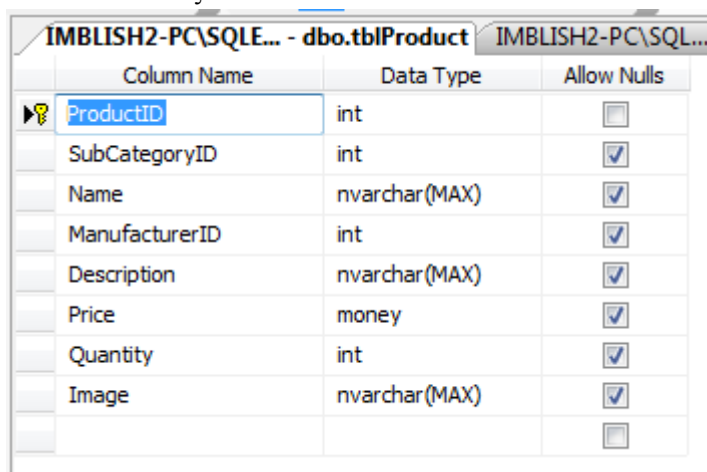


Column Name	Data Type	Allow Nulls
ManufacturerID	int	<input type="checkbox"/>
Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εικόνα 9: Πινάκας `tblManufacturer`

Ο Πινάκας αυτός χρησιμοποιείται για να έχουμε μια μορφή «τυποποίησης» στους κατασκευαστές μας και μια ομοιογένεια ώστε να είναι εύκολος ο διαχωρισμός των προϊόντων μας στο site με βάση των κατασκευαστή. Η εισαγωγή στοιχείων γίνεται από την καρτέλα `Admin.aspx` από το menu item “Manufacturer” και εισάγοντας το όνομα μας εισάγει στον πίνακα την νέα εγγραφή. Αυτός ο πίνακας συνδέεται με τον `tblProduct` μέσω του `ManufacturerID` με επιλογές στα `Delete Rule` και `Update Rule` σε `No Action` και `Cascade` αντίστοιχα.

- Πινάκας `dbo.tblProduct`



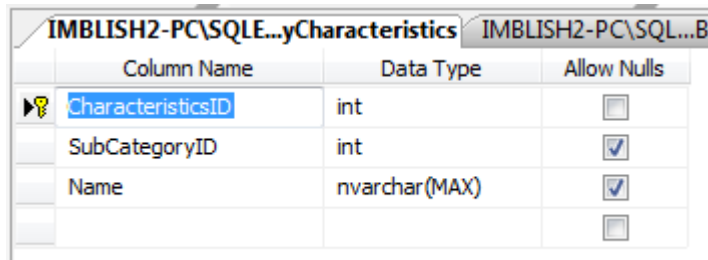
Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
SubCategoryID	int	<input checked="" type="checkbox"/>
Name	nvarchar(MAX)	<input checked="" type="checkbox"/>
ManufacturerID	int	<input checked="" type="checkbox"/>
Description	nvarchar(MAX)	<input checked="" type="checkbox"/>
Price	money	<input checked="" type="checkbox"/>
Quantity	int	<input checked="" type="checkbox"/>
Image	nvarchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εικόνα 10: Πινάκας `tblProduct`

Ο πίνακας `Product` χρησιμοποιείται για να αποθηκεύει τα προϊόντα που έχουμε. Σε αυτόν αποθηκεύουμε διάφορα στοιχεία όπως το όνομα, την υποκατηγορία το οποίο ανήκει, το όνομα του κατασκευαστή, την τιμή, την διαθέσιμη ποσότητα καθώς και ένα πεδίο `Image` το οποίο εμείς αποθηκεύουμε την «διαδρομή» που θα βρει την εικόνα που αντιστοιχεί στο κάθε προϊόν ώστε να το δείξει στον χρήστη. Η εισαγωγή των στοιχείων γίνεται μέσα από την καρτέλα `Admin.aspx` όπου ο διαχειριστής πατώντας το menu Item “product” και βάζοντας τα στοιχεία που χρειάζεται δηλαδή όνομα, περιγραφή, κατασκευαστή, υποκατηγορία, τιμή, ποσότητα και μια εικόνα. Για την τροποποίηση ή την διαγραφή ο διαχειριστής επιλέγει την κατάλληλη επιλογή και αφού επιλέξει το προϊόν και του εμφανιστούν τα στοιχεία τα τροποποιεί ή διαγράφει το προϊόν αν θέλει. Ο «απλός» χρήστης κατά την διάρκεια της περιήγησης του στο site από αυτόν τον πίνακα διαβάζει τα προϊόντα και διάφορα χαρακτηριστικά για το προϊόν αυτό τα οποία όμως τα βλέπει από άλλον πίνακα που θα αναλύσουμε παρακάτω. Ο Πίνακας `Products` συνδέεται με τον `tblSubCategory`, τον `tblManufacturer`, τον `tblOrder` μέσω του `ProductID` με επιλογές στα `Delete Rule` και `Update Rule` σε `No Action` και στα

2 και τον tblProductCharacteristics μέσω του ProductID με επιλογές στα Update και Delete Rules σε Cascade και τα δυο.

- Ο Πίνακας dbo.tblSubCategoryCharacteristics

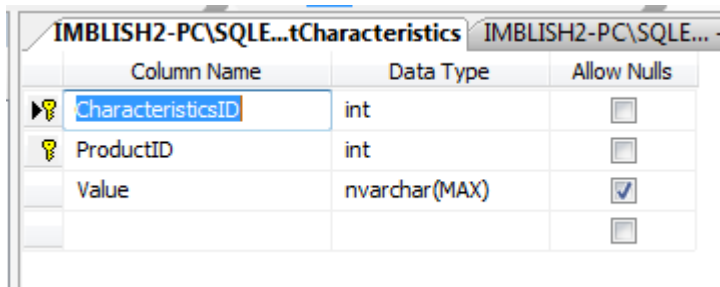


Column Name	Data Type	Allow Nulls
CharacteristicsID	int	<input type="checkbox"/>
SubCategoryID	int	<input checked="" type="checkbox"/>
Name	nvarchar(MAX)	<input checked="" type="checkbox"/>

Εικόνα 11: Πίνακας tblSubCategoryCharacteristics

Ο Πίνακας αυτός χρησιμοποιείται όπως περίπου και ο πίνακας του κατασκευαστή για να έχουμε μια μορφή τυποποίησης των χαρακτηριστικών που έχουμε για κάθε υποκατηγορία προϊόντων όπως πχ για μια οθόνη υπολογιστή αποθηκεύουμε σαν χαρακτηριστικό την σύνδεση του (dvi-d,vga) την μέγιστη ανάλυση της οθόνης όπως και την διάσταση της οθόνης σε ίντσες. Για την εισαγωγή , τροποποίηση , διαγραφή σε αυτόν τον πίνακα πάμε στην καρτέλα admin.aspx και επιλέγοντας “Χαρακτηριστικά” Μπορεί από κει να κάνει ότι χρειαστεί. Αυτός ο πίνακας συνδέεται με τον πίνακα tblSubcategory και με τον ProductCharacteristics μέσω του CharacteristicsID με επιλογές στο Update Rule και Delete Rule και τα δυο σε Cascade.

- Ο Πίνακας dbo.tblProductCharacteristics

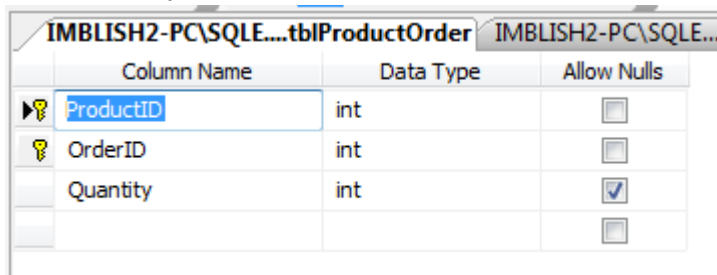


Column Name	Data Type	Allow Nulls
CharacteristicsID	int	<input type="checkbox"/>
ProductID	int	<input type="checkbox"/>
Value	nvarchar(MAX)	<input checked="" type="checkbox"/>

Εικόνα 12: Πίνακας tblProductCharacteristics

Ο πίνακας tblProductCharacteristics χρησιμοποιείται για να αποθηκεύσουμε στην βάση την τιμή που έχει το κάθε προϊόν για ένα συγκεκριμένο χαρακτηριστικό πχ μια οθόνη έχει διάσταση 22’. Για την εισαγωγή/διαγραφή/τροποποίηση αυτών των τιμών ο διαχειριστής θα πρέπει να πλοηγηθεί στην καρτέλα admin.aspx και επιλέγοντας «Τιμή Χαρακτηριστικού» και επιλέγοντας το τι θέλει να κάνει φορά κάνει και την αντίστοιχη ενεργεία. Για την εισαγωγή πχ θα πρέπει να επιλέξει από τις λίστες που εμφανίζονται στην οθόνη ένα προϊόν και το χαρακτηριστικό που θέλει να προσθέσει την τιμή και εισάγοντας την τιμή στο textbox από κάτω πατώντας τροποποίηση θα εισαγάγει μια καινούργια εγγραφή στο πίνακα με το id του προϊόντος και το id της τιμής του χαρακτηριστικού και τέλος την τιμή του χαρακτηριστικού για το προϊόν. Ο πίνακας αυτός αρά συνδέεται και με τον tblProduct μέσω του ProductID με επιλογές Cascade και στο Update και στο Delete Rule και με τον tblSubCategoryCharacteristics.

- Ο πίνακας dbo.tblProductOrder

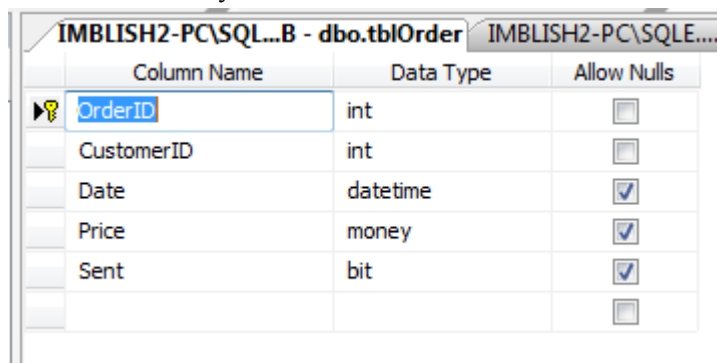


Column Name	Data Type	Allow Nulls
ProductID	int	<input type="checkbox"/>
OrderID	int	<input type="checkbox"/>
Quantity	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εικόνα 13: Πίνακας tblProductOrder

Στον πίνακα αυτόν αποθηκεύουμε στοιχεία σχετικά με τις παραγγελίες και τα προϊόντα. Ειδικότερα όπως φαίνεται και στην εικόνα από πάνω η κάθε εγγραφή του πίνακα έχει το id του προϊόντος, το id της παραγγελίας καθώς και την ποσότητα την οποία έχει ζητηθεί στην παραγγελία. Η εισαγωγή των στοιχείων γίνεται αυτόματα μέσα από ένα function στην σελίδα buy.aspx και αφού έχει πατήσει ο χρήστης το κουμπι αγοράσε. Αυτός ο πίνακας συνδέεται με τον tblProduct και με τον tblOrder μέσω του OrderID με επιλογές No Action σε Update και Delete Rule.

- Ο πίνακας dbo.tblOrder



Column Name	Data Type	Allow Nulls
OrderID	int	<input type="checkbox"/>
CustomerID	int	<input type="checkbox"/>
Date	datetime	<input checked="" type="checkbox"/>
Price	money	<input checked="" type="checkbox"/>
Sent	bit	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εικόνα 14: Πίνακας tblOrder

Ο πίνακας αυτός αποθηκεύει στοιχεία για την παραγγελία όπως το id της παραγγελίας, το id του χρήστη ο οποίος έκανε την παραγγελία, την ακριβή ημερομηνία και ώρα την οποία έγινε, την συνολική αξία όλων των προϊόντων και μια Boolean τιμή για τον αν έχει σταλεί η παραγγελία η όχι. Η εισαγωγή των στοιχείων αυτών εκτός από την Boolean που αναφέραμε πριν γίνεται αυτόματα όταν ο χρήστης αγοράσει προϊόντα από το κατάστημα μας. Η τροποποίηση της Sent γίνεται μέσα από την καρτέλα του admin.aspx όπου ο διαχειριστής βλέπει τις παραγγελίες και αν υπάρχουν τα προϊόντα επιλέγει να ολοκληρώσει την παραγγελία. Ο πίνακας αυτός συνδέεται με τους tblProductOrder και τον TblCustomer μέσω του CustomerID με No Action σε Update και Delete Rule.

- ο Πίνακας dbo.tblCustomer

Column Name	Data Type	Allow Nulls
CustomerID	int	<input type="checkbox"/>
FirstName	nvarchar(MAX)	<input checked="" type="checkbox"/>
LastName	nvarchar(MAX)	<input checked="" type="checkbox"/>
Email	varchar(MAX)	<input checked="" type="checkbox"/>
Password	varchar(MAX)	<input checked="" type="checkbox"/>
Address	nvarchar(MAX)	<input checked="" type="checkbox"/>
Town	nvarchar(MAX)	<input checked="" type="checkbox"/>
PostalCode	char(10)	<input checked="" type="checkbox"/>
HomePhone	char(10)	<input checked="" type="checkbox"/>
MobilePhone	char(10)	<input checked="" type="checkbox"/>

Εικόνα 15: Πίνακας tblCustomer

Στον πίνακα tblCustomer αποθηκεύουμε τα στοιχεία του χρήστη όπως το ονοματεπώνυμο του, το email του το οποίο χρησιμοποιείται και σαν user name, τον κωδικό του χρήστη, διεύθυνση, πόλη διαμονής, ταχυδρομικό κωδικό, τηλέφωνο οικείας και κινητό τηλέφωνο. Η εισαγωγή των στοιχείων γίνεται μέσα από την καρτέλα Register.aspx όπου ο ίδιος ο χρήστης εισάγει τα δεδομένα. Η τροποποίηση αυτών γίνεται πάλι από την ίδια καρτέλα register.aspx. Ο πίνακας αυτός συνδέεται με τον tblInvoice (για την περίπτωση που ο χρήστης θέλει να κάνει αγορά με τιμολόγιο) μέσω του CustomerID με επιλογές σε Update και Delete Rule σε No Action και με τον tblOrder για τις παραγγελίες που έχει κάνει.

- Ο πίνακας dbo.tblInvoice

Column Name	Data Type	Allow Nulls
CustomerID	int	<input type="checkbox"/>
CompanyName	nvarchar(MAX)	<input checked="" type="checkbox"/>
Profession	nvarchar(MAX)	<input checked="" type="checkbox"/>
Address	nvarchar(MAX)	<input checked="" type="checkbox"/>
AFM	nchar(10)	<input checked="" type="checkbox"/>
DOY	nvarchar(MAX)	<input checked="" type="checkbox"/>
CompanyPhone	nchar(10)	<input checked="" type="checkbox"/>

Εικόνα 16: Πίνακας tblInvoice

Στον πίνακα αυτό κρατάμε στοιχεία τιμολόγιου (αν το έχει επιλέξει ο χρήστης). Αποθηκεύουμε στοιχεία σχετικά με το όνομα της εταιρίας, το επάγγελμα, την διεύθυνση, το ΑΦΜ, την ΔΟΥ που ανήκει και ένα τηλέφωνο της εταιρίας. Η εισαγωγή και τροποποίηση των στοιχείων αυτών γίνεται από τον ίδιο τον χρήστη από την καρτέλα register.aspx. Ο πίνακας αυτός συνδέεται με τον tblCustomer για να ξέρουμε για ποιον χρήστη έχει γίνει η καταχώρηση.

3.4 Λίγα Λόγια για τις Stored Procedures στην Database

Για την καλύτερη λειτουργία του site και για να είναι πιο κατανοητό και εύχρηστο στην διαχείριση του κώδικα κατά την δημιουργία της πτυχιακής δημιουργήσαμε κάποιες stored procedures στην βάση. Υπάρχουν κάποια πλεονεκτήματα και μειονεκτήματα στην χρήση αυτών των stored procedures τα οποία θα δούμε παρακάτω.

Στα πλεονεκτήματα μπορούμε να πούμε ότι είναι η βελτίωση της απόδοσης της εφαρμογής καθώς όταν αποθηκεύονται στον κατάλογο της βάσης δεδομένων εκτελούνται πιο γρήγορα από τις άπλες εντολές Sql. Μια store procedure μειώνει το επικοινωνιακό φορτίο μεταξύ εφαρμογής και βάσης γιατί η εφαρμογή στέλνει μόνο το όνομα της διαδικασίας και όχι όλο το Sql statement. Επίσης ένα άλλο πλεονέκτημα είναι ότι έτσι οι διαδικασίες αυτές γίνονται εύκολα προσβάσιμες από όλη την εφαρμογή διευκολύνοντας τον προγραμματιστή. Ένα τελευταίο πλεονέκτημα είναι ότι μια τέτοια stored procedure είναι περισσότερο ασφαλής από ένα Sql statement καθώς δικαίωμα δίνει στην procedure και όχι στην επεξεργασία των πινάκων απευθείας.

Έκτος από πλεονεκτήματα όμως έχουμε και μειονεκτήματα. Αυξάνουν τον φόρτο εργασίας του εξυπηρετητή της βάσης από πλευράς μνήμης και επεξεργασίας. Δεν μπορούν να χρησιμοποιηθούν για πολύπλοκα προγράμματα που χρησιμοποιούν C++,java. Η ανίχνευση λαθών γίνεται δυσκολότερη γιατί κατά την μεταγλώττιση δεν εμφανίζονται τυπικά μηνύματα λάθους. Τέλος η δημιουργία και η διατήρηση stored procedures είναι μια ειδικευμένη ικανότητα που δεν διαθέτουν όλοι οι προγραμματιστές.

Παρακάτω παραθέτουμε μερικές Stored Procedures που χρησιμοποιήσαμε κατά την υλοποίηση της πτυχιακής.

```
create procedure [dbo].[listitems_sel]
(
  @ordid int
)
as
select tblProduct.ProductID,tblProduct.Quantity ,tblProductOrder.Quantity
from tblProduct inner join tblproductorder on tblProduct.ProductID=tblProductOrder.ProductID
where OrderID=@ordid
```

Η παραπάνω procedure παίρνει σαν είσοδο το ID μιας παραγγελίας και μας επιστρέφει τα προϊόντα, την ζητηθείσα ποσότητα, και την διαθέσιμη ποσότητα που υπάρχει stock στην αποθήκη.

```
create procedure [dbo].[tblProductCharacteristics_ins]
(
  @prodid int,
  @charid int,
  @value nvarchar(max)
)
as
insert into tblProductCharacteristics
([CharacteristicsID],[ProductID],[Value])
values (@charid,@prodid,@value)
```

Η παραπάνω procedure παίρνει σαν είσοδο το ID ενός προϊόντος, το ID ενός χαρακτηριστικού και μια τιμή και μας κάνει μια καινούργια εγγραφή στον πίνακα tblProductCharacteristics. Αυτή χρησιμοποιείται από την διεπαφή του Admin για να εισάγει τιμή χαρακτηριστικού σε ένα προϊόν.

```
create procedure [dbo].[customer_upd]
(
    @FirstName nvarchar(max),
    @LastName nvarchar(max),
    @Password varchar(max),
    @Email varchar(max),
    @Address nvarchar(max),
    @Town nvarchar(max),
    @PostalCode char(10),
    @HomePhone char(10),
    @MobilePhone char(10),
    @CustomerID int
)
AS
UPDATE tblCustomer
SET FirstName = @FirstName, LastName = @LastName, [Password] = @Password, Email =
@Email, [Address] = @Address,
    Town = @Town, PostalCode = @PostalCode, HomePhone = @HomePhone, MobilePhone =
@MobilePhone
WHERE CustomerID = @CustomerID
SELECT * FROM tblCustomer WHERE CustomerID = @CustomerID
```

Η παραπάνω procedure παίρνει σαν είσοδο διάφορα στοιχεία όπως όνομα, επώνυμο, τηλέφωνο κλπ και κάνει ανανέωση στα στοιχεία του χρήστη.

```
CREATE procedure [dbo].[tblOrder_ins]
(
    @customerid int,
    @date date,
    @price money,
    @sent bit
)
as
insert into tblOrder ([CustomerID] , [Date] ,[Price],[Sent] )
values (@customerid , @date, @price,@sent)
select tblOrder.OrderID from tblOrder where CustomerID=@customerid and Price=@price and
Date=@date
```

Η παραπάνω procedure παίρνει σαν είσοδο το ID του πελάτη, την ημερομηνία, τιμή, και τα δημιουργεί μια νέα εγγραφή στον πίνακα tblOrder. Το @sent bit είναι μια πληροφορία για τον διαχειριστή ώστε να μπορεί να καταλάβει αν έχει σταλθεί η παραγγελία η όχι ακόμα.

4. Ανάλυση διαδικασιών σε ASP.NET

4.1 Επικύρωση για τις Web φόρμες

Πολλές φορές ένα πρόγραμμα μπορεί να αντιμετωπίσει προβλήματα λόγω λανθασμένων δεδομένων, όπως αριθμούς αντί χαρακτήρες, κενό αντί μιας συμβολοσειράς κοκ. Για να αποφύγουμε αποθήκευση τυχόν λανθασμένων δεδομένων, χρησιμοποιούμε τα στοιχεία ελέγχου επικύρωσης.

Τα στοιχεία ελέγχου επικύρωσης (ή validator) καθορίζουν αν τα δεδομένα σε ένα άλλο στοιχείο ελέγχου Web είναι στην κατάλληλη μορφή. Στην ASP.NET μπορούμε να προσθέσουμε στοιχεία ελέγχου επικύρωσης σε όλα σχεδόν τα στοιχεία δεδομένων εισόδου της σελίδας. Τα στοιχεία ελέγχου επικύρωσης παρέχουν ένα μηχανισμό για την επικύρωση της εισόδου χρήστη στον client. Όταν δημιουργηθεί η HTML για τη σελίδα μας, το στοιχείο επικύρωσης μετατρέπεται σε ECMAScript που εκτελεί την επικύρωση. ECMAScript είναι μια γλώσσα scripting που βελτιώνει τη λειτουργικότητα και την εμφάνιση των σελίδων Web. Για παράδειγμα, στην παρακάτω φόρμα έχουμε βάλει έλεγχο επικύρωσης στο πεδίο του κωδικού. Αν δώσουμε μόνο τρία ψηφία η επικύρωση θα αποτύχει και στη συνέχεια η σελίδα μας επιστρέφει ένα μήνυμα σφάλματος και μας ενημερώνει ότι τα στοιχεία που δώσαμε δεν είναι στην μορφή που περιμένει.



The image shows two side-by-side screenshots of a registration form titled "Εγγραφή". The left screenshot shows the form with empty input fields for "Όνομα:", "Επώνυμο:", "e-mail:", and "Κωδικός:". The right screenshot shows the form with the following values: "Όνομα:" is "Μάριος", "Επώνυμο:" is "Φανταουτσάκης", "e-mail:" is "marios_gatos@hotmail.com", and "Κωδικός:" is "...". Below the "Κωδικός:" field, there is a message: "Ο κωδικός πρέπει να έχει 6-15 ψηφία".

Εικόνα 17: Στοιχεία Ελέγχου Επικύρωσης

Η ASP.NET έχει έξι είδη στοιχείων ελέγχου επικύρωσης: το `RequiredFieldValidator`, το `RegularExpressionValidator`, το `CompareValidator`, το `RangeValidator`, το `CustomValidator` και το `ValidationSummary`. Και τα έξι στοιχεία έχουν μερικές κοινές ιδιότητες και μεθόδους. Κληρονομούνται από την κλάση `BaseValidator`, από την κλάση `WebControl` του χώρου `System.UI.WebControls` και από την κλάση `Controls` που βρίσκεται στο `System.Web.UI`. Μοναδική εξαίρεση αποτελεί η ιδιότητα `Text`, η οποία κληρονομείται από την κλάση `Label`.

4.1.1 RequiredFieldValidator

Το στοιχείο ελέγχου `RequiredFieldValidator` είναι το στοιχείο το οποίο ελέγχει αν ο χρήστης έχει δώσει μια τιμή δεδομένων στο στοιχείο ελέγχου εισόδου. Η ιδιότητα `ControlToValidate` παίρνει τιμή το ID του στοιχείου εισόδου που θέλουμε. Μόλις γίνει η αντιστοίχιση αυτή διασφαλίζεται ότι στο συγκεκριμένο πεδίο εισόδου θα εισαχτεί μια τιμή και δε θα παραμείνει κενό.

Το `RequiredFieldValidator` λειτουργεί με την ιδιότητα `InitialValue`. Το `InitialValue` είναι η αρχική τιμή του στοιχείου ελέγχου επικύρωσης. Αν η τιμή του `InitialValue` είναι ίση με την τιμή του πεδίου εισόδου τότε η επικύρωση αποτυγχάνει και μας εμφανίζει στη σελίδα το κείμενο που υπάρχει στην ιδιότητα του `RequiredFieldValidator`, `ErrorMessage`.

```
<asp:TextBox ID="txtFName" runat="server" CssClass="textEntry"></asp:TextBox>
<asp:RequiredFieldValidator ID="FName" runat="server" CssClass="failureNotification"
ErrorMessage="Δώστε Όνομα" ControlToValidate="txtFName">*</asp:RequiredFieldValidator>
```

Στο παραπάνω κομμάτι κώδικα δημιουργούμε ένα TextBox με το όνομα txtFName. Στη συνέχεια, βάζουμε το RequiredFieldValidator με όνομα FName να ελέγχει αυτό το TextBox. Σε περίπτωση που ο χρήστης δεν εισάγει δεδομένα στο txtFName στη σελίδα θα εμφανιστεί το μήνυμα «Δώστε Όνομα». Σε αυτό το παράδειγμα λείπει η ιδιότητα InitialValue. Αυτό σημαίνει ότι η τιμή της θα είναι η προεπιλεγμένη από το πρόγραμμα, δηλαδή το κενό.

4.1.2 RegularExpressionValidator

Το στοιχείο ελέγχου RegularExpressionValidator διαβάζει την τιμή από ένα στοιχείο ελέγχου εισόδου και ελέγχει αν ταιριάζει με ένα σχέδιο που ορίζεται από μια κανονική έκφραση. Αυτή η έκφραση αποτελείται από μια σειρά χαρακτήρων και συμβόλων, κοινώς γνωστοί ως μετα-χαρακτήρες (ή meta-characters). Λειτουργεί ως πρότυπο και ελέγχει αν η δοθείσα συμβολοσειρά το ακλουθεί.

Το RegularExpressionValidator συναντάται συχνά σε πεδία εισόδου όπως e-mail, αριθμού τηλεφώνου κα, έτσι ώστε να σιγουρέψει την εγκυρότητα των δεδομένων.

```
<asp:RegularExpressionValidator Id="vldRejex" RunAt="server" ControlToValidate="txtemail"
ErrorMessage="Εισάγετε μια έγκυρη διεύθυνση e-mail"
ValidationExpression="^[a-zA-Z0-9_-\.\.+]@[a-z0-9-]+\(\.[a-z0-9-]+\)*\(\.[a-z]{2,3}\)$">
</asp:RegularExpressionValidator>
```

Στο παραπάνω κομμάτι κώδικα έχουμε δημιουργήσει ένα RegularExpressionValidator με όνομα vldRejex. Αυτό ελέγχει το πεδίο κειμένου txtEmail αν η συμβολοσειρά που περιέχει είναι της μορφής «^[a-zA-Z0-9_-\.\.+]@[a-z0-9-]+\(\.[a-z0-9-]+\)*\(\.[a-z]{2,3}\)\$». Αυτό σημαίνει ότι η συμβολοσειρά πρέπει να περιέχει στην αρχή της αλφαριθμητικά μέχρι να συναντήσει το χαρακτήρα @. Μετα περιμένει να βρει στη συνέχεια της συμβολοσειράς χαρακτήρες μόνο μικρούς και αριθμούς μέχρι να συναντήσει την τελεία. Τέλος, πρέπει να βρει στο τέλος της συμβολοσειράς δυο ή τρεις χαρακτήρες. Αν η συμβολοσειρά που υπάρχει στο txtEmail δεν έχει τέτοια μορφή τότε μας επιστρέφει το μήνυμα «Εισάγετε μια έγκυρη διεύθυνση e-mail»

4.2 CSS (Cascading Style Sheet)

4.2.1 Τι είναι το CSS

Η CSS (Cascading Style Sheets ή Διαδοχικά Φύλλα Στυλ) είναι μια γλώσσα που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου HTML. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα, δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την HTML. Για μια όμορφη και καλοσχεδιασμένη σελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

4.2.2 Σύνταξη

Η CSS έχει απλή σύνταξη και χρησιμοποιεί μερικές αγγλικές λέξεις-κλειδιά, για να καθορίσει τα ονόματα διαφόρων ιδιοτήτων στυλ.

Ένα style sheet απαρτίζεται από μια λίστα με κανόνες. Κάθε κανόνας περιέχει ένα ή περισσότερους selectors (επιλογείς) και ένα μπλοκ δήλωσης. Κάθε μπλοκ έχει μια σειρά από δηλώσεις τοποθετημένες μέσα σε αγκύλες ({...}). Κάθε δήλωση έχει τη μορφή *ιδιότητα-άνω και*

κάτω τελεία (:)-τιμή. Αν υπάρχουν πάνω από μια δήλωση στο μπλοκ, ξεχωρίζουν με το ελληνικό ερωτηματικό (;).

Στη CSS οι επιλογείς χρησιμοποιούνται για να δηλώσουν σε ποιο στοιχείο markup θα εφαρμοστούν.

Ένα παράδειγμα CSS είναι το εξής:

```
body
{
    background-color: #ffffff;
    background-image: url('../Images/bodybg.png');
    background-repeat: no-repeat;
}
```

Ονομάζουμε τον επιλογέα “body” και στη συνέχεια δίνουμε την ιδιότητα το χρώμα στο υπόβαθρο να είναι άσπρο, να πάρει μια εικόνα για υπόβαθρο από αρχείο και να μην επαναλαμβάνεται αυτή η εικόνα.

Η CSS γράφεται σε ξεχωριστό αρχείο και ενώνεται με την HTML με σύνδεσμο. Για παράδειγμα, αν έχουμε το αρχείο Style.css, στο head του HTML εγγράφου μας θα το καλέσουμε κάπως έτσι:

```
<link href="Styles/Style.css" rel="Stylesheet" type="text/css" />
```

Εφ’ όσον έχουμε συνδέσει CSS και HTML μπορούμε να δώσουμε σε ένα HTML element τις ιδιότητες του “body”, δηλαδή <div class=“body” />

4.3 Ασφάλεια ASP.NET και IIS

Η asp.net προσφέρει έλεγχο και ασφάλεια στην εφαρμογή μας. Η ασφάλεια του asp λειτουργεί σε συνδυασμό με τις υπηρεσίες Microsoft Internet Information Services Ασφάλεια (IIS) και περιλαμβάνει υπηρεσίες έλεγχου ταυτότητας και εξουσιοδότησης. το μοντέλο ασφαλείας του asp.net περιλαμβάνει επίσης μια βασισμένη σε ρόλους δυνατότητα ασφαλείας που μπορείτε να εφαρμόσετε σε συνδυασμό με το Microsoft windows και τον λογαριασμό χρήστη του windows.

4.3.1 Ροή ασφαλείας με μια αίτηση

Τα ακόλουθα βήματα περιγράφουν την ακολουθία συμβάντων, όταν ένα πρόγραμμα-πελάτη κάνει μια αίτηση:

1. Ένας υπολογιστής-πελάτης ζητά μια σελίδα .aspx που βρίσκεται σε ένα διακομιστή IIS ο διακομιστής.
2. Τα διαπιστευτήρια του πελάτη, μεταβιβάζονται IIS.
3. IIS ελέγχει την ταυτότητα του προγράμματος-πελάτη και προωθεί τον έλεγχο ταυτότητας το διακριτικό μαζί με αίτηση του προγράμματος-πελάτη για το ASP. Η διαδικασία εργασίας ΔΙΚΤΥΟΥ.
4. Με βάση το διακριτικό έλεγχο ταυτότητας από το IIS και το ρυθμίσεις παραμέτρων για την εφαρμογή Web, οι ASP.NET αποφασίζει αν θα απομίμηση ενός χρήστη σε νήμα που η επεξεργασία της αίτησης. Σε ένα σαφώς διαφορά μεταξύ Microsoft Active Server Pages (ASP) και ASP.NET ASP. Χωρίς ΚΑΘΑΡΗ πλέον υποδύεται το χρήστη με έλεγχο ταυτότητας από προεπιλογή. Για να ενεργοποιήσετε την απομίμηση, πρέπει να ορίσετε το μίμηση χαρακτηριστικό στην ενότητα ταυτότητα στο αρχείο Web.config για TRUE.

Σχετικές ρυθμίσεις παραμέτρων

Ο IIS διατηρεί τις ρυθμίσεις παραμέτρων που σχετίζονται με την ασφάλεια στο IIS μετα-βάσης. Ωστόσο, η ASP.NET διατηρεί το δίκτυο ασφαλείας (και άλλων) ρύθμιση παραμέτρων ρυθμίσεις στα αρχεία παραμέτρων Extensible Markup Language (XML). Παρόλο που αυτό γενικά απλοποιεί την ανάπτυξη της εφαρμογής σας από μια ασφαλείας άποψη, το μοντέλο ασφαλείας που θεσπίζει η

εφαρμογή σας απαιτεί την σωστή ρύθμιση παραμέτρων μετα-βάση IIS και σας ASP.NET εφαρμογής μέσω του αρχείου ρύθμισης παραμέτρων (Web.config).

4.3.2 Έλεγχος ταυτότητας

Έλεγχος ταυτότητας είναι η διαδικασία με την οποία μπορείτε να αποκτήσετε αναγνώριση πιστοποίησης, όπως το όνομα χρήστη και κωδικό πρόσβασης και να επικυρώσετε τις πιστοποιήσεις έναντι ορισμένων αρχή. Η ASP.NET παρέχει τέσσερις ελέγχους ταυτότητας υπηρεσίες παροχής:

- Φόρμες έλεγχος ταυτότητας

Έλεγχος ταυτότητας φορμών αναφέρεται σε ένα σύστημα στο οποίο χωρίς έλεγχο ταυτότητας οι αιτήσεις ανακατευθύνονται σε μια φόρμα Hypertext Markup Language (HTML) που οι χρήστες πληκτρολογούν τις πιστοποιήσεις τους. Αφού ο χρήστης παρέχει διαπιστευτήρια και υποβάλλει στη φόρμα την εφαρμογή, πιστοποιεί την αίτηση και ζητήματα συστήματος ένα δελτίο εξουσιοδότησης με τη μορφή ενός cookie. Αυτό το cookie που περιέχει το πλήκτρο για να σαρώσετε ξανά την ταυτότητα ή διαπιστευτήρια. Οι επόμενες αιτήσεις από το το cookie περιλαμβάνουν αυτόματα το πρόγραμμα περιήγησης.

- Τα Windows έλεγχος ταυτότητας

Με έλεγχο ταυτότητας των Windows, IIS εκτελεί έλεγχο ταυτότητας, και με έλεγχο ταυτότητας διακριτικού προωθείται η ASP. Η διαδικασία εργασίας δικτύου. Το πλεονέκτημα της χρήσης των Windows έλεγχος ταυτότητας είναι ότι απαιτεί ελάχιστη κωδικοποίηση. Αν θέλετε να χρησιμοποιήσετε τον έλεγχο ταυτότητας, τα Windows απομειώνονται το λογαριασμό χρήστη των Windows που ελέγχει την ταυτότητα των υπηρεσιών IIS πριν παραδώσετε την αίτηση για ASP.NET.

- Το Passport έλεγχος ταυτότητας

Ο έλεγχος ταυτότητας Passport είναι μια υπηρεσία κεντρικού ελέγχου ταυτότητας που η Microsoft παρέχει και προσφέρει ένα μεμονωμένο αρχείο καταγραφής σε βασικών υπηρεσιών προφίλ για τοποθεσίες του μέλους. Συνήθως χρησιμοποιείται έλεγχος ταυτότητας Passport όταν χρειάζεται μονό αρχείο καταγραφής τη δυνατότητα σε πολλούς τομείς.

- Προεπιλογή έλεγχος ταυτότητας

Ο προεπιλεγμένος έλεγχος ταυτότητας χρησιμοποιείται όταν δεν θέλετε ασφαλείας στην εφαρμογή σας Web. Απαιτείται ανώνυμη πρόσβαση για την ασφάλεια αυτή η υπηρεσία παροχής. Ανάμεσα σε όλες τις υπηρεσίες παροχής έλεγχου ταυτότητας παρέχει προεπιλεγμένο έλεγχο ταυτότητας μέγιστη απόδοση για την εφαρμογή σας. Είναι επίσης αυτήν την υπηρεσία παροχής έλεγχου ταυτότητας χρησιμοποιείται όταν χρησιμοποιήσετε το δικό σας λειτουργικής μονάδας ασφαλείας.

4.3.3 web.config

Το web.config είναι xml αρχείο το οποίο βρίσκεται μέσα στο φάκελο του e-center και προσφέρει κάποιες πληροφορίες σχετικά με διαφορές ρυθμίσεις που χρησιμοποιούμε για το site. Παράδειγμα :

```
<connectionStrings>
  <add name="myEshopDBConnectionString" connectionString="Data
Source=marios-pc\sqlexpress;Initial Catalog=myEshopDB;Integrated Security=True"
  providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Αυτό εδώ είναι κομμάτι του web config και υποδηλώνει το connection string Και ρυθμίσεις που τυχόν χρειάζεται για να κάνουμε σύνδεση με την βάση δεδομένων μας όπως το source, τον κατάλογο και διάφορες ρυθμίσεις ασφάλειας.

```
<appSettings>
  <add key="CONNECTION_STRING" value="Data Source=marios-
pc\sqlexpress;Initial Catalog=myEshopDB;Integrated Security=True"/>
</appSettings>
```

Επίσης κάποιες ρυθμίσεις για τις σελίδες της εφαρμογής.

```
<system.web>
  <compilation defaultLanguage="vb" debug="true" targetFramework="4.0"/>
  <authentication mode="Forms">
    <forms loginUrl="~/Account/Login.aspx" timeout="2880" />
  </authentication>
</system.web>
```

Και εδώ του έχουμε πει ότι η εφαρμογή έχει γραφτεί με vb και ότι θέλουμε να έχουμε debugging mode. Επίσης ότι χρησιμοποιεί αυθεντικοποίηση φορμών κάτι που αναφέραμε παραπάνω.

4.4 Cookies

Τα cookies είναι μικρά αρχεία κειμένου τα οποία αποθηκεύονται στον υπολογιστή μας κατά την πλοήγησή μας στο διαδίκτυο. Συνήθως περιγράφουν στοιχεία μας όπως όνομα χρήστη (user name) και συνθηματικό πρόσβασης (password) με σκοπό κατά την επίσκεψή μας στον ίδιο ιστότοπο αργότερα, να μας "θυμάται" και να κάνει login χωρίς να γράψουμε εμείς τίποτα. Τα cookies μπορεί να προέρχονται από τον ιστότοπο τον οποίο έχουμε επισκεφθεί ή από κάποιον άλλον. Συνήθως είναι άκακα, έχει όμως αποδειχθεί ότι μπορούν να στείλουν πληροφορίες για τη συμπεριφορά μας στο διαδίκτυο. Υπάρχουν προγράμματα που καθαρίζουν τα κακόβουλα cookies, ενώ αν ο χρήστης επιθυμεί να τα διαγράψει δίνεται αυτή η δυνατότητα μέσα από το φυλλομετρητή ιστοσελίδων.

4.5 Σύνδεση με βάση δεδομένων

Για να πραγματοποιήσουμε σύνδεση με την βάση δεδομένων έχουμε δυο τρόπους, είτε ορίζουμε εμείς απευθείας το connection String είτε δίνουμε το connection string από το web config.

Δηλώσεις και παραδείγματα:

```
Dim conn As New SqlConnection
(ConfigurationManager.AppSettings("CONNECTION_STRING"))
```

Εδώ δηλώνουμε μια sqlconnection με το connection_string από το appsettings του webconfig.

```
Dim conn As SqlConnection
conn = New SqlConnection("Server = marios-PC\sqlexpress;" & "initial Catalog =
myEshopDB;" & " Trusted_Connection=yes")
```

Εδώ δηλώνουμε ένα sqlconnection αλλά του ορίζουμε το connection string απευθείας δηλαδή του λέμε πήγαινε στο marios-PC\sqlexpress βρες τον κατάλογο myEshopDB και δημιούργησε μια σύνδεση έμπιστη.

Παράδειγμα ο παρακάτω κώδικας διαβάζει από την βάση κάποια στοιχεία του χρηστή και τα τυπώνει στην οθόνη που έχουμε βάλει εμείς

```
Dim reader As SqlDataReader
Dim conn As New SqlConnection
Dim command As New SqlCommand
Dim cmd As New SqlCommand
```

```

command.CommandType = CommandType.StoredProcedure
cmd.CommandType = CommandType.StoredProcedure

command.CommandText = "customer_sel"
command.Connection = conn
command.Parameters.AddWithValue("@CustomerID", Session("CustomerID"))
conn.Open()

reader = command.ExecuteReader()
While (reader.Read)
    Try
        Dim name As String
        name = reader("FirstName").ToString & " " & reader("LastName").ToString
        txtNameAtbuy.Text = name
        txtAddAtbuy.Text = reader("Address").ToString
        txtEmailAtbuy.Text = reader("Email").ToString
        txtMobileAtbuy.Text = reader("MobilePhone").ToString
        txtPhoneAtbuy.Text = reader("HomePhone").ToString
        txtTKAtbuy.Text = reader("PostalCode").ToString
        txtTownAtbuy.Text = reader("Town").ToString
    Catch ex As Exception
    End Try
End While
reader.Close()

```

Παράδειγμα στον παρακάτω κώδικα δημιουργούμε μια καινούργια κατηγορία στην βάση μας στον πίνακα tblCategory

```

Dim conn As New SqlConnection()
conn = sqlconnect()
Dim command As New SqlCommand
command.CommandType = CommandType.StoredProcedure
command.CommandText = "tblCategory_ins"
command.Connection = conn
command.Parameters.AddWithValue("@name", txtNameAtCat.Text)
command.Parameters.AddWithValue("@desc", txtDescAtCat.Text)
conn.Open()
command.ExecuteNonQuery()
conn.Close()

```

και η function Sqlconnect

```

Protected Function sqlconnect() As SqlConnection
    Dim conn As SqlConnection
    conn = New SqlConnection("Server = marios-PC\sqlexpress;" & "initial Catalog = myEshopDB;" & "Trusted_Connection=yes")
    Return conn
End Function

```

5. Η εφαρμογή μας

5.1 Ecenter.Master

Οι κύριες σελίδες (.master) της ASP.NET μας επιτρέπουν να δημιουργήσουμε μια συνεπή εμφάνιση για τις σελίδες της εφαρμογής μας. Μια ενιαία κεντρική σελίδα προσδιορίζει την εμφάνιση και την αισθητική και την τυπική συμπεριφορά που θέλουμε για όλες τις σελίδες (ή μια ομάδα σελίδων) στην εφαρμογή μας. Μπορούμε έτσι να δημιουργήσουμε ξεχωριστές σελίδες περιεχομένου που περιλαμβάνουν το περιεχόμενο που θέλουμε να εμφανίσουμε. Όταν οι χρήστες ζητούν τις σελίδες περιεχομένου, αυτές συγχωνεύονται με την κύρια σελίδα και παράγουν ως αποτέλεσμα ένα συνδυασμό εμφάνισης της κύριας σελίδας και της σελίδας περιεχομένου.

Στη σελίδα Ecenter.master έχουμε φτιάξει ολόκληρη τη φόρμα της ιστοσελίδας, περιλαμβάνοντας μέσα το λογότυπο, το κουτί αναζήτησης, τα κουτιά προτάσεων, το μενού πλοήγησης, τις πληροφορίες στο footer και στο κέντρο ένα ContentPlaceHolder. Με αυτόν τον τρόπο καταφέρνουμε να έχουμε όλα τα παραπάνω χαρακτηριστικά σε όλες τις σελίδες που κληρονομούν την κύρια σελίδα, και να αλλάζει το περιεχόμενο τους ανάλογα με το περιεχόμενο στο ContentPlaceHolder της κάθε σελίδας.

Επίσης, στη σελίδα Ecenter.master.vb (η οποία τρέχει μαζί με όλες τις σελίδες της εφαρμογής μας) γίνονται όλες οι διεργασίες, όπως να εμφανίζει σε κάθε αλλαγή της σελίδας, ή ανανέωση, καινούριες προτάσεις.

5.2 Default.aspx



Εικόνα 18: Default.aspx

Η σελίδα Default.aspx είναι η αρχική σελίδα και με αυτή ξεκινά η περιήγηση στην εφαρμογή μας. Εδώ ο πελάτης μπορεί να επιλέξει μια από τις κατηγορίες για να βρει ποια προϊόντα τον ενδιαφέρουν. Αν ψάχνει κάτι συγκεκριμένο μπορεί να πληκτρολογήσει το όνομα στο πεδίο κειμένου και να με το κουμπί αναζήτησης να το βρει (αν υπάρχει στο κατάστημα φυσικά). Επίσης, μπορεί να επιλέξει ένα από τα δυο τυχαία εμφανιζόμενα προϊόντα που βρίσκονται στο αριστερό μέρος της σελίδας. Στην μπάρα πλοήγησης μπορεί να πατήσει πάνω στο λογαριασμό, να εισέρθει να δημιουργήσει ένα νέο λογαριασμό (αν δεν έχει) ή αν είναι ήδη συνδεδεμένος να δει τις πληροφορίες του

λογαριασμού του. Ακόμα, στην μπάρα πλοήγησης μπορεί να πατήσει πάνω στο καλάθι αγορών και να δει τα περιεχόμενα του. Για όποιους πελάτες έχουν απορία για το πώς εκπληρώνονται οι πληρωμές των προϊόντων, η αποστολή των προϊόντων και αν είναι ασφαλής η αγορά on-line στο κατάστημα μας μπορεί να βρει πληροφορίες πατώντας τα κουμπιά κάτω από τις προτάσεις. Στο footer υπάρχουν πληροφορίες για την επικοινωνία με το κατάστημα.

Όταν ο χρήστης έχει εισέρθει στο λογαριασμό του τότε κάτω ακριβώς από το κουτί της αναζήτησης εμφανίζεται η διεύθυνση ηλεκτρονικού ταχυδρομείου του και το κουμπί Αποσύνδεση. Μπορεί πατώντας στο e-mail του να μεταφερθεί στη σελίδα AccInfo.aspx και να δει όλες τις πληροφορίες για το λογαριασμό του.



Εικόνα 19: Χρήστης on-line – Αποσύνδεση

5.3 Desktop.aspx



Εικόνα 20: Desktop.aspx

Στη σελίδα Desktop.aspx εμφανίζονται όλα τα προϊόντα που ανήκουν στην κατηγορία των έτοιμων υπολογιστών. Υπάρχει ένας πίνακας στον οποίο φαίνονται όλα τα προϊόντα της κατηγορίας αυτής τοποθετημένα βάση τιμής σε αύξουσα σειρά. Ο πίνακας μπορεί να δείχνει μόνο πέντε προϊόντα στη σελίδα, άρα αν ο πελάτης ψάχνει κάτι άλλο μπορεί πατώντας τους αριθμούς στο κάτω μέρος του πίνακα να βρεθεί στη σελίδα που επιθυμεί. Ο πελάτης μπορεί να επιλέξει ένα προϊόν από τον πίνακα πατώντας πάνω στο όνομα του και να δει περισσότερες λεπτομέρειες για το συγκεκριμένο προϊόν. Ακόμα, μπορεί να βάλει αμέσως ένα προϊόν στο καλάθι αγορών.

Επίσης, μπορεί να περιορίσει την αναζήτηση προϊόντων με τα φίλτρα που βρίσκονται πάνω από τον πίνακα. Για παράδειγμα, μπορεί να επιλέξει υπολογιστές Laptop, με κατασκευαστή Acer, τιμή έως 500 Ευρώ και επεξεργαστή Atom. Τότε στον πίνακα θα εμφανίζονται μόνο οι υπολογιστές με τα παραπάνω χαρακτηριστικά. Το καλό με τα φίλτρα είναι ότι μπορεί να επιλέξει παραπάνω από ένα τη φορά και με οποιοδήποτε συνδυασμό. Δηλαδή μπορεί να επιλέξει από το φίλτρο των σκληρών δίσκων

να βλέπει όλους τους υπολογιστές με σκληρό δίσκο 500 GB και μαζί όλους τους υπολογιστές με σκληρό δίσκο 250 GB. Τα φίλτρα αφορούν τι υπολογιστής είναι (επιτραπέζιος, laptop, netbook ή Mac), τον κατασκευαστή, το εύρος τιμής, τον επεξεργαστή, τη χωρητικότητα σκληρού δίσκου, το μέγεθος μνήμης RAM και το μέγεθος της οθόνης. Τέλος, πατώντας το κουμπί καθαρισμός φίλτρων, επιστρέφει στον αρχικό πίνακα με όλα τα προϊόντα που εμφανίζονταν και την πρώτη φορά που μπήκε ο πελάτης στη σελίδα αυτή.

5.4 Hardware.aspx

The screenshot shows the 'Hardware.aspx' page. At the top, there's a search bar and navigation tabs for 'Υπολογιστές', 'Υακω', 'Λογισμικό', and 'Καλάθι Αγορών'. Below the navigation, there are product categories: 'Cases', 'Psu', 'Cpu', 'Motherboard', 'HardDisk', 'RAM', 'Graphics Cards', 'Sound', 'DVD-Bluray', 'Monitor', and 'Mouse/Keyboard'. A list of products is displayed with columns for 'Εικόνα', 'Όνομα', 'Κατασκευαστής', and 'Τιμή'. The products listed are:

Εικόνα	Όνομα	Κατασκευαστής	Τιμή
	FLOPPY DISK DRIVE 3.5" BLACK	OEM	5,00 €
	KINGSTON KVR1333D3N9-1G DDR3 1GB PC10666 1533MHZ VALUE RAM	KINGSTON	7,22 €
	MICROSOFT COMFORT CURVE KEYBOARD 2000 DSP	MICROSOFT	11,50 €
	KINGSTON KHX1600C9AD3-1G 1GB PC12800 1600MHZ HYPERX	KINGSTON	11,57 €
	CORSAIR SDRAM SO-DIMM 1GB PC3300 (667MHZ) DDR2	CORSAIR	12,90 €

At the bottom of the page, there are logos for 'Εγγύηση - Εμπιστοποίηση - Ασφάλεια', 'Επιμέτρηση Πιστοτήτων', and 'Ασφαλείς Συνλλαγές'.

Εικόνα 21: Hardware.aspx

Η σελίδα Hardware.aspx είναι η ίδια σελίδα με την Desktop.aspx, μόνο που υπάρχουν δυο διαφορές.

Πρώτον, στη σελίδα αυτή βρίσκονται μόνο τα προϊόντα που ανήκουν στην κατηγορία υλικό. Αυτά τα προϊόντα δεν έχουν καμία σχέση με τα προϊόντα στη σελίδα Desktop.aspx. Τα προϊόντα της κατηγορίας χωρίζονται σε έντεκα υποκατηγορίες: Cases, Psu, Cpu, Motherboard, Ram, HardDisk, Graphics Card, Sound Card, DVD-Bluray, Monitor και Mouse/keyboard. Και σε αυτό το σημείο φτάνουμε στη δεύτερη διαφορά από τη σελίδα Desktop.aspx.

Τα φίλτρα πλέον δεν είναι ενεργοποιημένα εξ' αρχής. Για να περιορίσει ένας πελάτης την αναζήτηση του, πρέπει πρώτα να επιλέξει μια υποκατηγορία. Στη συνέχεια εμφανίζονται τα φίλτρα, βάση των χαρακτηριστικών που έχει η κάθε υποκατηγορία. Όπως και στην Desktop.aspx, όταν πατηθεί το κουμπί καθαρισμός φίλτρων όλα επιστρέφουν στην αρχική τους μορφή.

Κατηγορία Hardware:	Cpu
Κατασκευαστής:	AMD INTEL
Εύρος Τιμής:	εώς 50 50-100 100-200 200 και άνω
Chip	Celeron Dual Core Opteron Phenom Core i3 Core i5 Core i7 Athlon
Σύνδεση με motherboard	479 AM3 F LGA775

Εικόνα 22: Hardware.aspx - Φίλτρα CPU

Κατηγορία Hardware:	RAM
Κατασκευαστής:	CORSAIR GEIL KINGSTON SAMSUNG
Εύρος Τιμής:	εώς 50 50-100 100-200 200 και άνω
Τύπος	DDR-2 DDR-3 SO DIMM DDR2 SO DIMM DDR3 SO-DIMM
Μέγεθος	1GB 2GB 4GB

Εικόνα 23: Hardware.aspx - Φίλτρα RAM

5.5 Search.aspx

Όταν πατηθεί το κουμπί της αναζήτησης, σε όποια σελίδα και να βρίσκεται ο πελάτης, μεταφέρεται απ' ευθείας στη σελίδα Search.aspx. Το κείμενο που είχε γράψει στο πεδίο κειμένου αναζήτησης πριν πατήσει το κουμπί αναζήτησης, ελέγχεται αν υπάρχει στο όνομα όλων των προϊόντων που υπάρχουν στη βάση. Αν βρει προϊόντα με τέτοιο κείμενο στο όνομα τους, τότε μας τα εμφανίζει σε ένα πίνακα, στον οποίο μπορεί ο πελάτης είτε να δει περισσότερες λεπτομέρειες πατώντας στο όνομα του προϊόντος είτε να το προσθέσει αμέσως στο καλάθι αγορών. Σε διαφορετική περίπτωση εμφανίζεται το μήνυμα: «Δεν υπάρχουν προϊόντα με αυτά τα δεδομένα»

Για παράδειγμα, αυτό είναι το αποτέλεσμα αν πληκτρολογήσουμε τη λέξη Value στο κουτί αναζήτησης:

The screenshot shows the Ecenter website interface. At the top, there is a search bar and navigation links for 'Υπολογιστές', 'Υλικό', 'Λογαριασμός', and 'Καλάθι Αγορών'. Below the navigation, there are social media icons for Facebook and Twitter. The main content area displays search results in a table format. On the left, there are 'Προτάσεις' (Recommendations) for Intel products. The search results table has columns for 'Εικόνα', 'Όνομα', 'Κατασκευαστής', and 'Τιμή'. Two results are shown: 'Value' by Ecenter for 610,00 € and 'KINGSTON KVR1333D3N9/1G DDR3 1GB PC10666 1333MHZ VALUE RAM' by KINGSTON for 7,22 €. At the bottom, there are service and security logos including 'Εγγύηση - Εξυπηρέτηση - Ασφάλεια', 'Εγγύηση Ποιότητας', 'Εξυπηρέτηση Πελατών', and 'Ασφαλείς Συναλλαγές' with logos for VeriSign Secured, PayPal, MasterCard, and Visa.

Εικόνα 24: Search.aspx

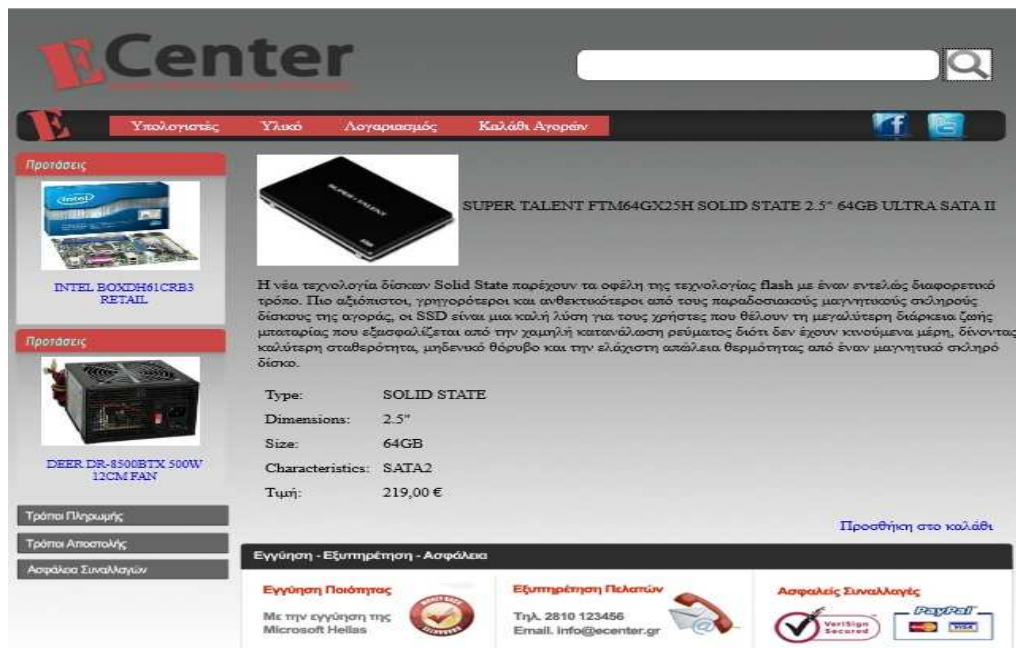
5.6 Product.aspx

Στη σελίδα αυτή εμφανίζεται ένα μόνο προϊόν τη φορά. Αν πατηθεί από οποιαδήποτε σελίδα το όνομα του προϊόντος που βρίσκεται στις προτάσεις, αν πατηθεί το όνομα του προϊόντος από τον πίνακα κατηγορίας (υπολογιστές/υλικό) ή και ακόμα από τον πίνακα της σελίδας αναζήτησης, ο πελάτης μεταβιβάζεται αμέσως σε αυτή τη σελίδα για να μπορέσει να δει λεπτομέρειες για το συγκεκριμένο προϊόν που διάλεξε.

Στο πάνω μέρος φαίνεται η εικόνα σε μεγαλύτερο μέγεθος από τους πίνακες ή τις προτάσεις. Δίπλα εμφανίζεται το όνομα του προϊόντος. Κάτω ακριβώς εμφανίζεται η περιγραφή του προϊόντος

και χαμηλότερα είναι ένας πίνακας με το όνομα των χαρακτηριστικών και δίπλα την τιμή του κάθε χαρακτηριστικού για το συγκεκριμένο προϊόν.

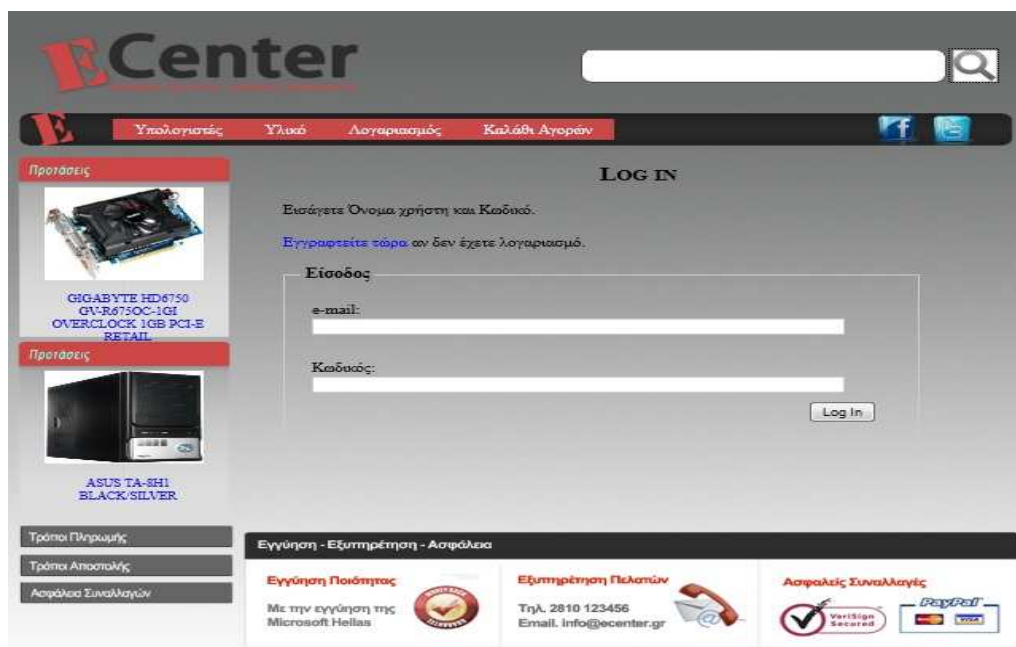
Στην παρακάτω εικόνα βλέπουμε τη σελίδα Product.aspx για το προϊόν «SUPER TALENT FTM64GX25H SOLID STATE 2.5" 64GB ULTRA SATA II»



The screenshot shows the product page for the SUPER TALENT FTM64GX25H SOLID STATE 2.5" 64GB ULTRA SATA II. The page features a search bar at the top, navigation tabs for 'Υπολογιστές', 'Υλικό', 'Λογαριασμός', and 'Καλάθι Αγορών', and social media icons. On the left, there are 'Προτάσεις' (Recommendations) for an Intel motherboard and a power supply. The main product image is a black SSD. Below it, the product name is displayed. A detailed description in Greek explains the benefits of Solid State technology. Technical specifications are listed: Type: SOLID STATE, Dimensions: 2.5", Size: 64GB, Characteristics: SATA2, and Price: 219,00 €. A 'Προσθήκη στο καλάθι' (Add to cart) button is present. At the bottom, there are sections for 'Εγγύηση - Εξυπηρέτηση - Ασφάλεια' (Warranty - Customer Service - Security), 'Εγγύηση Ποιότητας' (Quality Assurance), 'Εξυπηρέτηση Πελατών' (Customer Service), and 'Ασφαλείς Συναλλαγές' (Secure Transactions) with logos for VeriSign, PayPal, and Visa.

Εικόνα 25: Product.aspx

5.7 Login.aspx



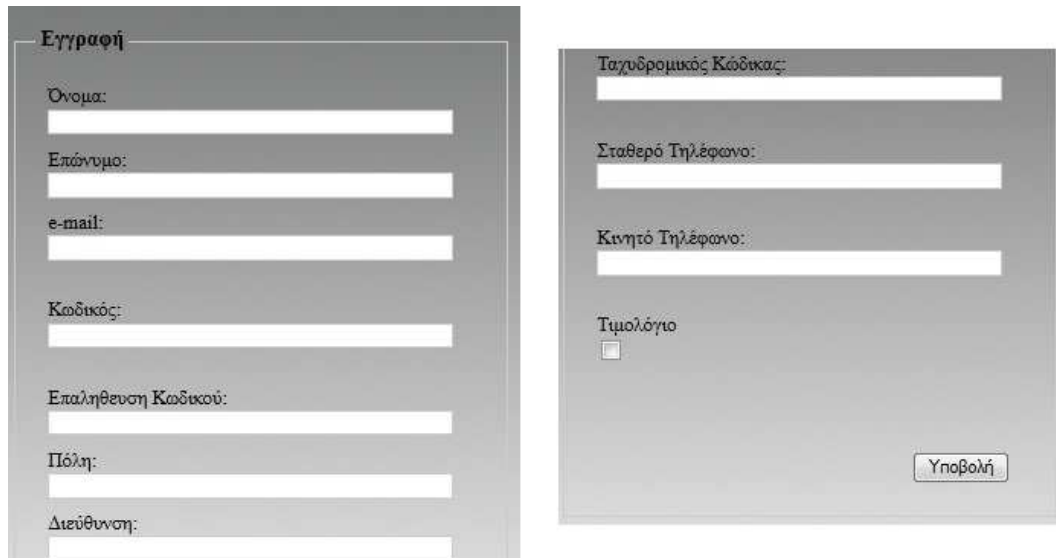
The screenshot shows the login page (Login.aspx) on the BCenter website. The page features a search bar at the top, navigation tabs for 'Υπολογιστές', 'Υλικό', 'Λογαριασμός', and 'Καλάθι Αγορών', and social media icons. On the left, there are 'Προτάσεις' (Recommendations) for a Gigabyte motherboard and an ASUS tower PC. The main content area is titled 'LOG IN' and contains the text: 'Εισάγετε Όνομα χρήστη και Κωδικό.' (Enter Username and Password.) and 'Εγγραφείτε τώρα αν δεν έχετε λογαριασμό.' (Register now if you don't have an account.). Below this, there are two input fields: 'e-mail:' and 'Κωδικός:' (Password:). A 'Log In' button is located at the bottom right of the form. At the bottom, there are sections for 'Εγγύηση - Εξυπηρέτηση - Ασφάλεια' (Warranty - Customer Service - Security), 'Εγγύηση Ποιότητας' (Quality Assurance), 'Εξυπηρέτηση Πελατών' (Customer Service), and 'Ασφαλείς Συναλλαγές' (Secure Transactions) with logos for VeriSign, PayPal, and Visa.

Εικόνα 26: Login.aspx

Η σελίδα αυτή αποτελείται από ένα link, δυο πεδία κειμένου και ένα κουμπί. Όταν ένας πελάτης δεν έχει λογαριασμό και πατήσει το link (εγγραφείτε τώρα) μεταφέρεται στη σελίδα Register.aspx για να δημιουργήσει ένα νέο λογαριασμό. Αλλιώς, βάζει στο πεδίο e-mail τη διεύθυνση του ηλεκτρονικού ταχυδρομείου του, εν συνεχεία τον κωδικό που έχει επιλέξει και πατώντας το κουμπί

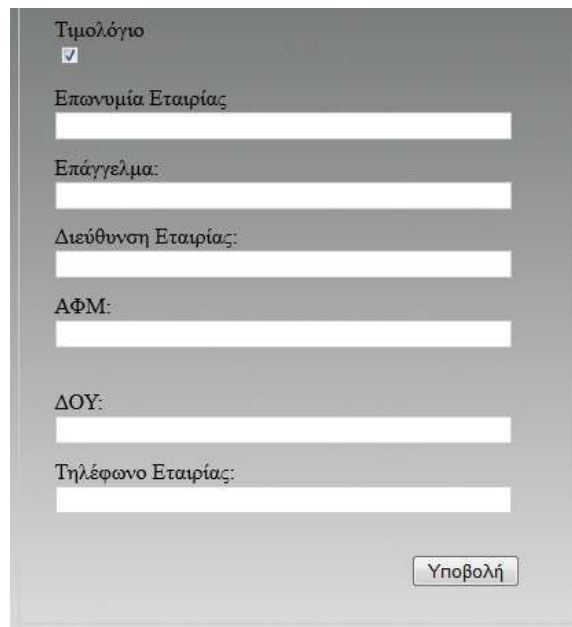
Login αν είναι σωστά τα στοιχεία που έδωσε εισέρχεται στο λογαριασμό του. Σε κάθε πεδίο υπάρχει RequiredFieldValidator με το οποίο υποδεικνύουμε στον πελάτη να συμπληρώσει κάθε πεδίο.

5.8 Register.aspx



Εικόνα 27: Register.aspx - Φόρμα εγγραφής

Η φόρμα αυτή χρειάζεται σε ένα πελάτη για να καταχωρήσει τα στοιχεία του στη βάση δεδομένων του ηλεκτρονικού καταστήματος και να μπορέσει να προβεί σε αγορά των προϊόντων με ασφάλεια.



Εικόνα 28: Register.aspx - Φόρμα Τιμολογίου

Πεδία που είναι απαραίτητα όπως Όνομα έχουν RequiredFieldValidator για να θυμίσουν στον πελάτη ότι δεν πρέπει να παραλειφθούν. Σε μερικά πεδία που η εγκυρότητα των δεδομένων είναι αναγκαία, όπως το e-mail, υπάρχει RegularExpressionValidator, για να αποφευχθούν τυχόν λάθη.

Στο πεδίο Επαλήθευση Κωδικού εκτός από το RequiredFieldValidator, που είναι απαραίτητο γιατί ο πελάτης πρέπει να επαληθεύει τον κωδικό που έχει βάλει επειδή μπορεί να έχει γίνει λάθος που δε

φαίνεται, υπάρχει και το CompareValidator όπου ελέγχει αν ο κωδικός που δόθηκε είναι ίδιος με αυτόν της επαλήθευσης.

Μόλις τελειώσει ο πελάτης με την καταχώρηση των στοιχείων του πατάει στο κουμπί Υποβολή και αν όλα είναι σωστά τότε η εγγραφή του ολοκληρώνεται. Αν δεν είναι σωστά όλα τα πεδία η κάτι λείπει τότε δίπλα από το κουτί με τα στοιχεία εμφανίζεται το ValidationSummary που υποδεικνύει ποια είναι τα λάθη. Μετά τη διόρθωση τους μπορεί ο πελάτης να ξαναπροσπαθήσει να εγγραφεί.

The image shows a registration form titled "Εγγραφή" (Registration) with five input fields, each followed by a red asterisk indicating a required field. To the right of the form is a list of validation requirements in red text:

- Δώστε Όνομα
- Δώστε Επώνυμο
- Δώστε e-mail
- Δώστε Κωδικό
- Επαληθεύστε τον κωδικό σας
- Δώστε Τηλέφωνο

The input fields are labeled as follows:

- Όνομα:
- Επώνυμο:
- e-mail:
- Κωδικός:
- Επαληθευση Κωδικού:

Εικόνα 29: Register.aspx – RequiredFieldValidator

5.9 AccInfo.aspx

The screenshot shows the 'Center' website interface. At the top, there's a navigation bar with 'Υπολογιστές', 'Υλικό', 'Λογαριασμός', and 'Καλάθ. Αγορών'. Below this, the user profile 'marios_gatos@hotmail.com' is visible. The main content area is divided into sections: 'Προτάσεις' (Recommendations) with product images, 'Προσωπικά Στοιχεία' (Personal Data) with fields for name, address, and phone, and 'Στοιχεία Τιμολογίου' (Invoice Details) with company information. A table titled 'Αλλαγή Στοιχείων' (Change Items) lists purchase history with columns for 'Αριθμός Παραγγελίας', 'Ημερομηνία', 'Όνομα Προϊόντος', and 'Τιμή'. At the bottom, there are logos for 'Εγγύηση Ποιότητας', 'Εξυπηρέτηση Πελατών', and 'Ασφαλείς Συναλλαγές'.

Αριθμός Παραγγελίας	Ημερομηνία	Όνομα Προϊόντος	Τιμή
11	23/11/2011 12:00:00 πμ	TOSHIBA SATELLITE L755-S5246 15.5" INTEL B940	500,0000
11	23/11/2011 12:00:00 πμ	ASUS P5G41 T-M LX RETAIL	43,4200

Εικόνα 30: AccInfo.aspx

Στη σελίδα αυτή, κάθε πελάτης που έχει λογαριασμό στο κατάστημα, μπορεί να ελέγξει αν τα στοιχεία που έδωσε είναι σωστά, κι αν βρει κάποιο λάθος να το διορθώσει.

Αποτελείται από δυο κουτιά, ένα για όλα τα προσωπικά στοιχεία του πελάτη και το δεύτερο για τα στοιχεία τιμολογίου του. Για να μπορεί ο πελάτης να έχει πρόσβαση και να τα αλλάξει, υπάρχει το κουμπί «Αλλαγή Στοιχείων», όπου μεταφέρει τον πελάτη στη σελίδα Register.aspx, με τη διαφορά ότι αυτή τη φορά τα πεδία είναι ήδη συμπληρωμένα από τα στοιχεία που είχε εισάγει την πρώτη φορά ο πελάτης.

The screenshot shows a form for updating user details. The fields are pre-filled with the user's information: Ονόμα: Μάριος, Επώνυμο: Φανταουτάκης, e-mail: marios_gatos@hotmail.com, Κωδικός: (empty), Επαληθευση Κωδικού: (empty), Πόλη: Ηράκλειο, Διεύθυνση: Ακρωτηρίου 3, Ταχυδρομικός Κωδικός: 71303.

Εικόνα 31: Register.aspx - Αλλαγή Στοιχείων

Τέλος, κάτω από το κουμπί «Αλλαγή Στοιχείων», υπάρχει ένας πίνακας που κρατάει ιστορικό με τις αγορές του πελάτη. Δείχνει το όνομα του προϊόντος που αγόρασε, την ημερομηνία και την ώρα που το αγόρασε και ποια ήταν η τιμή του προϊόντος.

5.10 ViewCart.aspx

The screenshot shows the 'ViewCart.aspx' page of the ECenter website. At the top, there is a search bar and navigation links for 'Υπολογιστές', 'Υλικό', 'Λογαριασμός', and 'Καλάθι Αγορών'. The main content area features a table with the following items:

Όνομα	Ποσότητα	Τιμή Μονάδας	Σύνολο
CORSAIR VS1GSDS400 SO-DIMM DDR 1GB PC3200 (400MHZ)	4	37,96 €	151,84 €
GIGABYTE GEFORCE 8400GS GV-N84S-512I 512MB PCI-E RETAIL	2	37,50 €	75,00 €
HP MINI 210-2080SV BLUE XK431EA	1	380,00 €	380,00 €
Αξία Προϊόντων: 606,84 €			

Below the table, there is a button for 'Ανανέωση Καλαθιού' and a 'Αγόρασε' button. On the left, there are product recommendations for 'GIGABYTE GA-G41MT-S2PT RETAIL' and 'ASUS P5G41T-M LX RETAIL'. At the bottom, there are sections for 'Εγγύηση - Εξυπηρέτηση - Ασφάλεια', 'Εγγύηση Ποιότητας' (with a Microsoft logo), 'Εξυπηρέτηση Πελατών' (with phone and email icons), and 'Ασφαλείς Συναλλαγές' (with VeriSign and PayPal logos).

Εικόνα 32: Καλάθι Αγορών

Για να βρεθούμε σε αυτή τη σελίδα, είτε πατάμε στο κουμπί «Καλάθι Αγορών» στην μπάρα πλοήγησης, είτε πατώντας προσθήκη στο καλάθι σε κάποιο προϊόν.

Το καλάθι είναι στην ουσία ένας πίνακας, ο οποίος «κρατάει» τα προϊόντα που θέλει ο πελάτης μέχρι να τα αγοράσει όλα μαζί. Ο πίνακας αυτός δείχνει το όνομα του προϊόντος που έχει επιλέξει ο πελάτης, την ποσότητα που επιθυμεί από κάθε προϊόν, την τιμή της μονάδας και τη συνολική τιμή. Η ποσότητα είναι ένα πεδίο κειμένου στο οποίο μπορεί ο πελάτης να βάλει την τιμή που επιθυμεί. Πατώντας μετά το κουμπί κάτω από τον πίνακα «Ανανέωση Καλαθιού» ενημερώνεται αυτόματα και το σύνολο. Στο τέλος του πίνακα υπάρχει και η αξία προϊόντων όπου μας δείχνει πόσο κοστίζουν όλα μαζί τα προϊόντα που βρίσκονται στο καλάθι.

Ο πελάτης μπορεί να συνεχίσει στην αγορά αυτών πατώντας το κουμπί «Αγόρασε».

5.11 HowToPay.aspx-HowToSend.aspx & SecurePay.aspx

Οι σελίδες αυτές αντιστοιχούν στα κουμπιά που βρίσκονται στο κάτω αριστερό μέρος όλων των σελίδων του ηλεκτρονικού μας καταστήματος. Δεν είναι τίποτα άλλο από απλές πληροφορίες που λύνουν απορίες στους πελάτες που δεν έχουν εμπειρία από συναλλαγές στο Internet.

Η σελίδα HowToPay.aspx αναφέρεται στους τρόπους πληρωμής και εξηγεί πως μπορεί κάποιος να πληρώσει τα προϊόντα που θέλει να αγοράσει από το ηλεκτρονικό κατάστημα.

Υπολογιστές **Υλικό** **Λογαριασμός** **Καλάθι Αγορών**

Προτάσεις

Τρόποι Πληρωμής

1. Αντικαταβολή

Για όσους προτιμούν την πιο παραδοσιακή μέθοδο, τα προϊόντα έρχονται μέσω των οδηγών του ecenter ή του courier και εσείς πληρώνετε το αντίτιμο την στιγμή της παραλαβής.

2. Πιστωτική κάρτα

Αν διαθέτετε πιστωτική κάρτα VISA, MasterCard μπορείτε να τη χρησιμοποιήσετε με ασφάλεια μέσα από την φόρμα παραγγελίας μας. Η διαδικασία εξόφλησης μέσω πιστωτικών καρτών καλύπτεται από τις πιο σύγχρονες προδιαγραφές ασφάλειας στον secure server του ecenter με κρυπτογράφηση 128 bit - τη μέγιστη που διατίθεται σήμερα - πιστοποιημένη από την Thawte Consulting (Pty) Ltd. Με χρήση πιστωτικής κάρτα μπορείτε να πληρώσετε έως 24 δόσεις, σύμφωνα με τις δυνατότητες και τους περιορισμούς της κάρτας σας. Η μηνιαία επιβάρυνση είναι περίπου 0.5% που αντιστοιχεί σε ετήσιο επιτόκιο 10.8%. Για παράδειγμα για αγορές € 1000 με 12 δόσεις θα πληρώσετε τελικά €1060 δηλαδή 12 μήνες x 0.5% επιβάρυνση = 6% περισσότερο. Κατά την πρώτη φορά χρήσης της πιστωτικής κάρτας στο κατάστημά μας θα σας ζητηθούν να αποστείλετε ένα αντίγραφο της ταυτότητας καθώς και της πιστωτικής κάρτας (με καλυμμένα τα 4 τελευταία ψηφία) στο logistinio@e-shop.gr ή στο fax 210 68 35 837. Ζητούμε την κατανόησή σας για την πρόσθετη αυτή διαδικασία που αποκλειστικό σκοπό έχει την αποφυγή χρήσης κλεμμένων πιστωτικών καρτών και την προστασία έτσι των πραγματικών ιδιοκτητών τους. Τα στοιχεία αυτά ζητούνται μόνο κατά την πρώτη χρήση της κάρτας, παραμένουν απολύτως εμπιστευτικά και δεν παραχωρούνται σε οποιονδήποτε τρίτο.

Εγγύηση - Εξυπηρέτηση - Ασφάλεια

Εγγύηση Ποιότητας **Εξυπηρέτηση Πελατών** **Ασφαλείς Συναλλαγές**

Με την εγγύηση της Microsoft Hellas Τηλ. 2810 123456 Email: info@securepay.gr

Γρήγορη Πληρωμή **Τρόπος Αποστολής** **Ασφαλείς Συναλλαγές**

Εικόνα 33: Τρόποι Πληρωμής

Η σελίδα HowToSend περιέχει ένα κείμενο στο οποίο αναφέρεται στο πώς τα προϊόντα που θα αγοράσει κάποιος θα σταλούν στη διεύθυνση του και αν θα έχει επιβάρυνση από τα έξοδα αποστολής.

Υπολογιστές **Υλικό** **Λογαριασμός** **Καλάθι Αγορών**

Προτάσεις

Τρόποι Αποστολής

Τα περισσότερα προϊόντα αποστέλλονται την επόμενη εργάσιμη μέρα με δικά μας μέσα, μετά από τηλεφωνική συνεννόηση, ή από κάποια συνεργαζόμενη εταιρεία courier (VFS ή ACS).

Για αγορές οποιονδήποτε προϊόντων ύψους 30 € και άνω η αποστολή γίνεται δωρεάν ενώ για αγορές μικρότερου ύψους το κόστος είναι 3 €, εκτός από δυσπρόσιτες περιοχές όπου υπάρχει επιπλέον χρέωση.

Τα έξοδα αποστολής με Courier θα είναι 5,50 € μέχρι 2 κιλά και 7€ για δέμα μέχρι 8 κιλά. Για κάθε επιπλέον κιλό η χρέωση είναι 1.5 €. Τα έξοδα μπορεί να υπερβούν αυτό το ποσό λόγω του όγκου των προϊόντων ή σε περίπτωση που η περιοχή σας χαρακτηρίζεται δυσπρόσιτη από τις εταιρείες courier.

Όλες οι παραπάνω τιμές περιλαμβάνουν ΦΠΑ

Κατά την υποβολή της παραγγελίας σας επιστρέφεται αυτόματα ένα μήνυμα ηλεκτρονικού ταχυδρομείου που δείχνει τι παραγγείλατε. Εσείς δε χρειάζεται να κάνετε τίποτε άλλο. Η παραγγελία σας έχει πάρει το δρόμο της και μπορείτε να βλέπετε τα στάδια διεκπεραίωσης στην εξέλιξη παραγγελίας. Αν υπάρξει κάποια ανωμαλία θα επικοινωνήσουμε και εμείς μαζί σας.

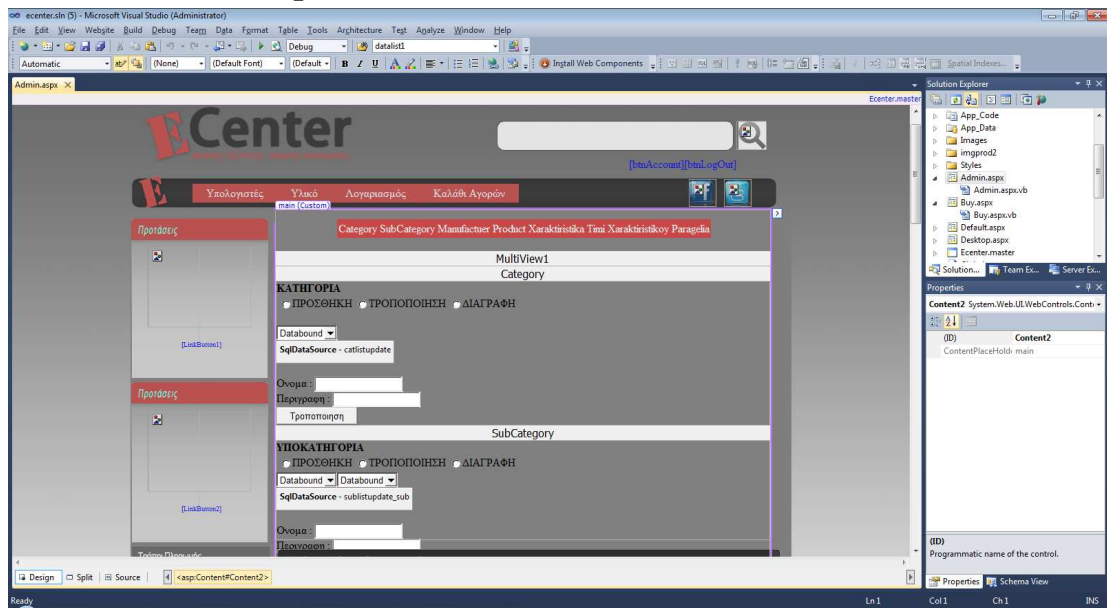
Εικόνα 34: Τρόποι Αποστολής

Τέλος, η σελίδα SecurePay.aspx ενημερώνει τους πελάτες για την ασφάλεια του ηλεκτρονικού καταστήματος, ότι οι κωδικοί τους δεν κινδυνεύουν από υποκλοπές κα.



Εικόνα 35: Ασφάλεια Συναλλαγών

5.12 Admin.aspx



Εικόνα 36: Admin.aspx

Η φόρμα admin.aspx χρησιμοποιείται από τους Διαχειριστές του site και μονό και προσφέρει μια σειρά δυνατοτήτων όπου αυτές είναι

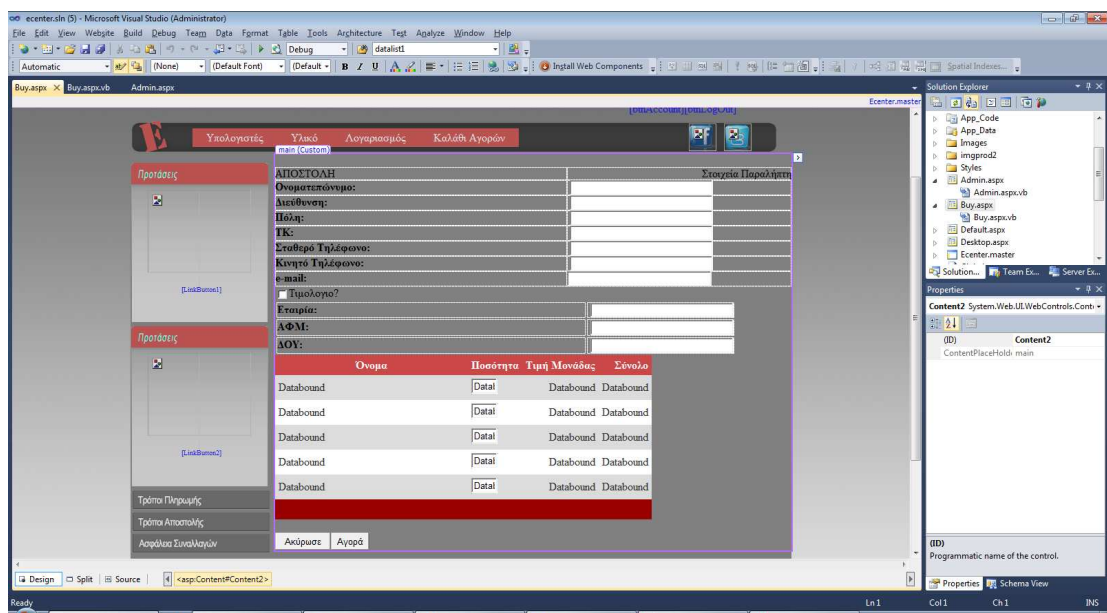
- Δημιουργία/Τροποποίηση/Διαγραφή Κατηγορίας
- Δημιουργία/Τροποποίηση/Διαγραφή Υποκατηγορίας
- Δημιουργία/Τροποποίηση/Διαγραφή Χαρακτηριστικού
- Δημιουργία/Τροποποίηση/Διαγραφή Τιμής Χαρακτηριστικού για προϊόν
- Δημιουργία/Τροποποίηση/Διαγραφή Προϊόντος
- Δημιουργία/Τροποποίηση/Διαγραφή Κατασκευαστή
- Διαγραφή Παραγγελίας

- Και τέλος μια επίβλεψη των ανοιχτών παραγγελιών που δεν έχουν γίνει ακόμα και την δυνατότητα να τις περαιώσουμε.

Η χρήση της φόρμας αυτής γίνεται μονό και εφόσον έχουμε συνδεθεί σαν admin@ecenter.gr δηλαδή σαν διαχειριστής ένας έλεγχος για το id του session εμποδίζει την ανεπιθύμητη είσοδο. Ο διαχειριστής στην κορυφή βλέπει ένα menu με τις διάφορες επιλογές που αναφέραμε προηγουμένως και αφού έχουμε χρησιμοποιήσει multiview βλέπει μονό εκείνο που έχει επιλέξει.

Για την παραγγελία αυτό που βλέπει, είναι ένας πίνακας με τις «ανοιχτές» παραγγελίες, επιλέγοντας μια παραγγελία βλέπει ένα πίνακα με τα προϊόντα και την ποσότητα της παραγγελίας καθώς και την διαθέσιμη ποσότητα που υπάρχει. Πατώντας το button Έλεγχε ελέγχει αν μπορεί να γίνει η παραγγελία δηλαδή έχουμε την ποσότητα που χρειάζεται και αν ναι ενεργοποιεί το κουμπί στείλε το όποιο ολοκληρώνει την διαδικασία της αποστολής.

5.13 Buy.aspx



Εικόνα 37: Buy.aspx

Στην φόρμα ο χρήστης προωθείται αφού πρώτα έχει πατήσει το κουμπί Αγόρασε από την φόρμα ViewCart.aspx. Σε αυτή την φόρμα ο χρήστης βλέπει τα στοιχεία αποστολής (ονοματεπώνυμο, διεύθυνση, πόλη, κτλ) επίσης έχει την δυνατότητα να επιλέξει αν θέλει, ή όχι, να γίνει η παραγγελία με τιμολόγιο το οποίο το έχει εισάγει κατά την εγγραφή του η αργότερα. Επίσης ξαναβλέπει ένα πίνακα με τα προϊόντα που έχει παραγγείλει καθώς και τις ποσότητες και τις τιμές ανάλογα.

Οπότε έχει δυο επιλογές ή θα ακυρώσει την παραγγελία ή θα πατήσει το κουμπί Αγορά και θα καταχωρηθεί η παραγγελία του ως «ανοιχτή» και ο ίδιος θα ανακατευθυνθεί στην message.aspx οπού και του δηλώνεται ότι η παραγγελία του καταχωρήθηκε.

6. Βιβλιογραφία - Πηγές

Βιβλία που χρησιμοποιήθηκαν για τη συγγραφή της πτυχιακής εργασίας:

- Visual Basic .net Προγραμματισμός Εισαγωγή στο .NET και τα web services
- Pro ASP.NET 2.0 Website Programming

Ιστοσελίδες που βρήκαμε χρήσιμες πληροφορίες:

- http://www.cs.uoi.gr/~pitoura/courses/db/MySQL_stored_procedures.html
- http://en.wikipedia.org/wiki/SQL_Server_Management_Studio_Express
- http://el.wikipedia.org/wiki/%CE%92%CE%AC%CF%83%CE%B7_%CE%B4%CE%B5%CE%B4%CE%BF%CE%BC%CE%AD%CE%BD%CF%89%CE%BD
- http://el.wikipedia.org/wiki/Sql_server
- <http://support.microsoft.com/kb/306590/el>
- http://el.wikipedia.org/wiki/HTTP_cookies
- http://www.bloggertips.gr/2011/03/blog-post_05.html
- <http://www.codeproject.com/KB/aspnet/aspnetintro.aspx>
- <http://msdn.microsoft.com/en-us/library/wtxbf3hh.aspx>
- http://en.wikipedia.org/wiki/Cascading_Style_Sheets#Use
- <http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/ad2a22b9-135c-432a-bc9f-c67f074242b7.msp?mfr=true>
- http://el.wikipedia.org/wiki/Visual_Basic
- [http://msdn.microsoft.com/en-us/library/yedba920\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/yedba920(v=vs.71).aspx)

7. Παράρτημα

7.1 Ecenter.master

Στο παρακάτω κομμάτι θα εξετάσουμε τον κώδικα που «τρέχει» στη σελίδα αυτή. Αρχικά βλέπουμε τον κώδικα για το σχεδιασμό της σελίδας μας.

```
<%@ Master Language="VB" CodeFile="Ecenter.master.vb" Inherits="Ecenter" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>e-center</title>
<link href="Styles/Style.css" rel="Stylesheet" type="text/css" />
<link href="Styles/ListStyle.css" rel="Stylesheet" type="text/css" />
<link href="Styles/Fieldset.css" rel="Stylesheet" type="text/css" />
</head>
<body>
<form id="form1" runat="server" defaultbutton="btnSearch">
<div class="page">
<div class="header">
<a href="Default.aspx" runat="server"><asp:Image ID="imgLogo"
runat="server" CssClass="logo" ImageUrl="~/Images/logo.png" /></a>
<div class="search">
<asp:TextBox ID="txtSearch" runat="server" CssClass="txtsearch"
Height="36px"
Width="330px"></asp:TextBox>
<asp:ImageButton ID="btnSearch" runat="server" AlternateText=" "
CssClass="btnsearch" EnableTheming="True" Height="42px"
Width="44px"
ToolTip="Search" />
<asp:LinkButton ID="btnLogOut" runat="server" style="float:right;
margin-top:15px;"></asp:LinkButton>
<asp:LinkButton ID="btnAccount" runat="server" style="float:right;
margin-top:15px;"></asp:LinkButton>
</div>
</div>
<div class="menu">
<asp:Menu ID="NavigationMenu" runat="server" CssClass="lista"
Orientation="Horizontal" EnableViewState="false" IncludeStyleBlock="false">
<Items>
<asp:MenuItem Text="Υπολογιστές" Value="Desktop"
NavigateUrl="Desktop.aspx"></asp:MenuItem>
<asp:MenuItem Text="Υλικό" Value="Hardware"
NavigateUrl="Hardware.aspx"></asp:MenuItem>
<asp:MenuItem Text="Λογαριασμός" Value="Account">
<asp:MenuItem Text="Είσοδος" Value="Login"
NavigateUrl="~/Account/Login.aspx"></asp:MenuItem>
<asp:MenuItem Text="Δημιουργία Λογαριασμού"
Value="Δημιουργία Λογαριασμού" NavigateUrl="~/Account/Register.aspx"></asp:MenuItem>
<asp:MenuItem Text="Πληροφορίες" Value="Info"
NavigateUrl="~/Account/AccInfo.aspx"></asp:MenuItem>
</asp:MenuItem>
<asp:MenuItem Text="Καλάθι Αγορών" Value="Cart"
NavigateUrl="~/ViewCart.aspx"></asp:MenuItem>
</Items>
</asp:Menu>
```

```

        <asp:ImageButton ID="btntw" runat="server" AlternateText=" "
EnableTheming="true" height="40px" Width="40px" ToolTip="Twitter" style="float:right;
margin:-34px 58px 0px 0px;" />
        <asp:ImageButton ID="btnfb" runat="server" AlternateText=" "
EnableTheming="true" height="40px" Width="40px" ToolTip="Facebook" style="float:right;
margin:-34px 11px 0px 0px;" />
    </div>
    <div class="main">
        <div class="left">
            <div class="leftbutton" style="height:238px; float:left;
background-image: url('/ecenter/Images/protaseis.png'); background-repeat: no-
repeat;">
                <asp:Image ID="Image1" runat="server" Height="120px"
ImageAlign="Middle" style="margin-left: 25px; margin-top:
41px"
                    Width="150px"/>
                <p style="text-align:center">
                    <asp:LinkButton ID="LinkButton1" runat="server"
Width="165px" Font-Size="Smaller"></asp:LinkButton>
                </p>
            </div>
            <div class="leftbutton" style="height:238px; float:left;
background-image: url('/ecenter/Images/protaseis.png'); background-repeat: no-
repeat;">
                <asp:Image ID="Image2" runat="server" Height="120px"
ImageAlign="Middle" style="margin-left: 25px; margin-top:
41px"
                    Width="150px"/>
                <p style="text-align:center">
                    <asp:LinkButton ID="LinkButton2" runat="server"
Width="165px" Font-Size="Smaller"></asp:LinkButton>
                </p>
            </div>
            <a href="http://www.paypal.com"><asp:Image ID="imgPayMethod"
CssClass="leftbutton" runat="server" ImageUrl="Images/tropoipliromis.png" /></a>
            <a href="http://www.acscourier.gr"><asp:Image ID="imgDeliv"
CssClass="leftbutton" runat="server" ImageUrl="Images/tropoiapostolis.png" /></a>
            <a href="http://www.paypal.com"><asp:Image ID="imgSecure"
CssClass="leftbutton" runat="server" ImageUrl="Images/asfaleiasinalaggon.png" /></a>
        </div>
        <div style="width:750px;float:right;">
            <asp:ContentPlaceHolder id="main" runat="server">
            </asp:ContentPlaceHolder>
        </div>
        <div class="footer">
            <asp:Image ID="imgGuan" CssClass="footbox" runat="server"
ImageUrl="~/Images/eggiisipoiotitas.png" />
            <asp:Image ID="imgSupp" CssClass="footbox" runat="server"
ImageUrl="~/Images/eksipiretisipelaton.png" />
            <asp:Image ID="imgSec" CssClass="footbox" runat="server"
ImageUrl="~/Images/asfalissinalages.png" />
        </div>
    </div>
</div>
</form>
</body>
</html>

```

Στη συνέχεια γυρνάμε στη σελίδα Ecenter.master.vb για να εκμεταλλευτούμε όσα βάλουμε στη σελίδα μας.

Με την υπορουτίνα btnSearch_Click αποθηκεύουμε τις λέξεις που βρίσκονται στο textbox Αναζήτησης σε ένα Session, κόβουμε τα επιπλέον κενά και μεταφέρουμε τον πελάτη στη σελίδα Search.aspx για να δει τα αποτελέσματα.

```
Protected Sub btnSearch_Click(ByVal sender As Object, ByVal e As
System.Web.UI.ImageClickEventArgs) Handles btnSearch.Click
    Session("SEARCH") = txtSearch.Text.Trim
    Response.Redirect("~/Search.aspx", False)
End Sub
```

Μόλις φορτώσει η σελίδα ανοίγει σύνδεση με τη βάση από όπου παίρνουμε δυο τυχαία προϊόντα για να τα εμφανίσουμε στην αριστερή στήλη της ιστοσελίδας. Έπειτα, ελέγχουμε αν είναι συνδεδεμένος ο χρήστης. Αν είναι, εμφανίζουμε στην οθόνη το e-mail του και ένα κουμπί αποσύνδεσης δίπλα. Όταν πατηθεί το κουμπί αποσύνδεσης μπαίνουμε στην υπορουτίνα btnLogout_Click απ' όπου αδειάζουμε όλα τα sessions και επιστρέφει τον χρήστη στην αρχική σελίδα.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    Dim con As New SqlConnection
    (System.Configuration.ConfigurationManager.AppSettings("CONNECTION_STRING"))
    Dim cmd As New SqlCommand
    Dim dr As SqlDataReader
    If Not IsPostBack Then
        Try
            cmd = New SqlCommand("SELECT TOP 2 ProductID, Name, Image FROM
tblProduct ORDER BY NEWID()", con)
            con.Open()
            dr = cmd.ExecuteReader()
            While dr.Read()
                LinkButton1.Text = dr("Name").ToString
                Image1.ImageUrl = dr("Image").ToString
                dr.Read()
                LinkButton2.Text = dr("Name").ToString
                Image2.ImageUrl = dr("Image").ToString
            End While
            dr.Close()
        Catch ex As Exception
            Session("MESSAGE") = ex.ToString
            Response.Redirect("~/Message.aspx", False)
        Finally
            cmd.Dispose()
            con.Close()
        End Try
    End If

    If Session("CustomerID") > 0 Then
        Dim comd As New SqlCommand
        Dim drEmail As SqlDataReader
        comd.CommandType = Data.CommandType.StoredProcedure
        comd.CommandText = "customer_sel"
        comd.Connection = con
        comd.Parameters.AddWithValue("@CustomerID", Session("CustomerID"))
        Try
            con.Open()
            drEmail = comd.ExecuteReader
            If drEmail.Read() Then
                btnAccount.Text = drEmail("Email")
            End If
            drEmail.Close()
        Catch ex As Exception
```

```

        MsgBox(ex.ToString)
    Finally
        cmd.Dispose()
        con.Close()
    End Try
    btnLogout.Text = ", Αποσύνδεση"
End If
End Sub

Protected Sub btnLogout_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnLogout.Click
    Session.RemoveAll()
    Session.Abandon()
    Response.Redirect("~/Default.aspx", True)
End Sub

```

Παρακάτω βλέπουμε τι γίνεται όταν πατηθεί ένα από τα δυο τυχαία προϊόντα που φαίνονται στη σελίδα. Αποθηκεύουμε σε ένα session τον αύξοντα αριθμό του προϊόντος και στη συνέχεια στέλνουμε τον χρήστη στη σελίδα Product.aspx για να δει τις λεπτομέρειες του προϊόντος.

```

Protected Sub LinkButton1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LinkButton1.Click
    Session("Product") = GetProdID(LinkButton1.Text)
    Response.Redirect("~/Product.aspx", False)
End Sub

```

```

Protected Sub LinkButton2_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LinkButton2.Click
    Session("Product") = GetProdID(LinkButton2.Text)
    Response.Redirect("~/Product.aspx", False)
End Sub

```

```

Protected Function GetProdID(ByVal name As String) As Integer
    Dim con As New SqlConnection
    (ConfigurationManager.AppSettings("CONNECTION_STRING"))
    Dim cmd As New SqlCommand("Select ProductID From tblProduct Where Name = '"
& name & "'", con)
    Dim dr As SqlDataReader
    Dim _id As Integer
    Try
        con.Open()
        dr = cmd.ExecuteReader
        If dr.Read() Then
            _id = dr("ProductID")
        End If
        dr.Close()
    Catch ex As Exception
        Session("MESSAGE") = ex.ToString
        Response.Redirect("~/Message", True)
    Finally
        cmd.Dispose()
        con.Close()
    End Try
    Return _id
End Function

```

Τέλος, όταν πατήσει ένας συνδεδεμένος χρήστης στο εμφανιζόμενο e-mail του τότε μεταφέρεται στη σελίδα AccInfo.aspx για περισσότερες πληροφορίες του λογαριασμού του.

```

Protected Sub btnAccount_Click(sender As Object, e As System.EventArgs) Handles
btnAccount.Click

```

```

        Response.Redirect("~/Account/AccInfo.aspx", False)
    End Sub
End Class

```

7.2 Default.aspx

Στην aspx σελίδα περνάμε όλα τα στοιχεία που θέλουμε να συγγενεύσουμε στη σελίδα master μέσω του ContentPlaceHolder.

```

<asp:Content ID="BodyContent" ContentPlaceHolderID="main" Runat="Server">
    <div class="center"></div>
        <div class="right" id="adv1" style="margin:10px 10px 0px 0px; background-
image: url('/ecenter/Images/advert1.png'); background-repeat:no-repeat"></div>
        <div class="right" id="adv2" style="background-image:
url('/ecenter/Images/advert2.png'); background-repeat:no-repeat"></div>
        <div class="box" id="box1"
            style="background-image: url('/ecenter/Images/metaforika.png');
background-repeat: no-repeat"></div>
        <div class="box" id="box2"
            style="background-image: url('/ecenter/Images/epistrofi.png');
background-repeat: no-repeat"></div>
        <div class="box" id="box3"
            style="background-image: url('/ecenter/Images/topothetisi.png');
background-repeat: no-repeat"></div>
</asp:Content>

```

7.3 Desktop.aspx

Το περιεχόμενο της σελίδας φαίνεται από το παρακάτω κομμάτι κώδικα.

```

<asp:Content ID="mainContent" ContentPlaceHolderID="main" Runat="Server">
    <asp:Table ID="Table1" runat="server">
        <asp:TableRow>
            <asp:TableCell>Υπολογιστές:</asp:TableCell>
            <asp:TableCell>
                <asp:Menu ID="SubCategoryFilter" runat="server" CssClass="
filterlist" Orientation="Horizontal" EnableViewState="true"
IncludeStyleBlock="false"></asp:Menu>
            </asp:TableCell>
        </asp:TableRow>
        <asp:TableRow>
            <asp:TableCell>Κατασκευαστής:</asp:TableCell>
            <asp:TableCell>
                <asp:Menu ID="ManufacturerFilter" runat="server" CssClass="
filterlist" Orientation="Horizontal" EnableViewState="true"
IncludeStyleBlock="false"></asp:Menu>
            </asp:TableCell>
        </asp:TableRow>
        <asp:TableRow>
            <asp:TableCell>Εύρος Τιμής:</asp:TableCell>
            <asp:TableCell>
                <asp:Menu ID="PriceFilter" runat="server" CssClass=" filterlist"
Orientation="Horizontal" EnableViewState="true" IncludeStyleBlock="false">
                    <Items>
                        <asp:MenuItem Text="έως 500" Value="500"></asp:MenuItem>
                        <asp:MenuItem Text="500-800" Value="500-
800"></asp:MenuItem>
                        <asp:MenuItem Text="800-1000" Value="800-
1000"></asp:MenuItem>
                    </Items>
                </asp:Menu>
            </asp:TableCell>
        </asp:TableRow>
    </asp:Table>

```



```

                <asp:MenuItem Text="1000 και άνω"
Value="1000"></asp:MenuItem>
            </Items>
        </asp:Menu>
    </asp:TableCell>
</asp:TableRow>
<asp:TableRow>
    <asp:TableCell>Επεξεργαστής:</asp:TableCell>
    <asp:TableCell>
        <asp:Menu ID="CPUFilter" runat="server" CssClass=" filterlist"
Orientation="Horizontal" EnableViewState="true"
IncludeStyleBlock="false">
            <Items>
                <asp:MenuItem Text="Atom" Value="Atom"></asp:MenuItem>
                <asp:MenuItem Text="Celeron"
Value="Celeron"></asp:MenuItem>
                <asp:MenuItem Text="Pentium"
Value="Pentium"></asp:MenuItem>
                <asp:MenuItem Text="Core 2 Duo" Value="Duo"></asp:MenuItem>
                <asp:MenuItem Text="Core i3" Value="i3"></asp:MenuItem>
                <asp:MenuItem Text="Core i5" Value="i5"></asp:MenuItem>
                <asp:MenuItem Text="Core i7" Value="i7"></asp:MenuItem>
            </Items>
        </asp:Menu>
    </asp:TableCell>
</asp:TableRow>
<asp:TableRow>
    <asp:TableCell>Σκληρός Δίσκος:</asp:TableCell>
    <asp:TableCell>
        <asp:Menu ID="HDDFilter" runat="server" CssClass=" filterlist"
Orientation="Horizontal" EnableViewState="true"
IncludeStyleBlock="false"></asp:Menu>
    </asp:TableCell>
</asp:TableRow>
<asp:TableRow>
    <asp:TableCell>Μνήμη:</asp:TableCell>
    <asp:TableCell>
        <asp:Menu ID="RAMFilter" runat="server" CssClass=" filterlist"
Orientation="Horizontal" EnableViewState="true"
IncludeStyleBlock="false"></asp:Menu>
    </asp:TableCell>
</asp:TableRow>
<asp:TableRow>
    <asp:TableCell>Οθόνη:</asp:TableCell>
    <asp:TableCell>
        <asp:Menu ID="MonitorFilter" runat="server" CssClass=" filterlist"
Orientation="Horizontal" EnableViewState="true"
IncludeStyleBlock="false"></asp:Menu>
    </asp:TableCell>
</asp:TableRow>
    <asp:TableCell><asp:Label ID="lblChoose" runat="server"
Visible="false">Έχετε επιλέξει:</asp:Label></asp:TableCell>
    <asp:TableCell><asp:Menu ID="chooseMenu" runat="server"
CssClass="filterlist" Orientation="Horizontal" EnableViewState="true"
IncludeStyleBlock="false" Visible="false"></asp:Menu></asp:TableCell>
</asp:TableRow>
</asp:Table>
<asp:LinkButton ID="lbtnClear" runat="server">Καθαρισμός
Φίλτρων</asp:LinkButton>
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"

```

```

AutoGenerateColumns="False" EmptyDataText="Δεν υπάρχουν προϊόντα με αυτά τα
δεδομένα"
PageSize="5" CellPadding="4" ForeColor="#333333" GridLines="None"
DataSourceID = "SqlDataSource1" DataKeyNames="ProductID"
OnPageIndexChanging="GridView1_PageIndexChanging">
<AlternatingRowStyle BackColor="White" />
<Columns>
    <asp:ImageField DataImageUrlField="Image" HeaderText="Εικόνα">
        <ControlStyle Height="65px" Width="80px" />
    </asp:ImageField>
    <asp:ButtonField DataTextField="Name" HeaderText="Όνομα"
SortExpression="Name"
        CommandName="ProductName">
    <ItemStyle HorizontalAlign="Center" Width="280px" />
    </asp:ButtonField>
    <asp:BoundField DataField="Manufacturer" HeaderText="Κατασκευαστής"
SortExpression="Manufacturer" >
    <ItemStyle HorizontalAlign="Center" Width="105px" />
    </asp:BoundField>
    <asp:BoundField DataField="Price" HeaderText="Τιμή"
SortExpression="Price" DataFormatString="{0:C}" >
    <ItemStyle HorizontalAlign="Center" Width="75px" />
    </asp:BoundField>
    <asp:ButtonField CommandName="AddtoCart" Text="Προσθήκη στο Καλάθι" />
</Columns>
<FooterStyle BackColor="#990000" Font-Bold="True" ForeColor="White" />
<HeaderStyle BackColor="#CB4848" Font-Bold="True" ForeColor="White" />
<PagerStyle BackColor="#DC5050" ForeColor="#333333"HorizontalAlign="Center"
/>
<RowStyle BackColor="#DCDCDC" ForeColor="#333333" />
<SelectedRowStyle BackColor="#acacac" Font-Bold="True" ForeColor="Navy" />
<SortedAscendingCellStyle BackColor="#acacac" />
<SortedAscendingHeaderStyle BackColor="#4D0000" />
<SortedDescendingCellStyle BackColor="#acacac" />
<SortedDescendingHeaderStyle BackColor="#820000" />
</asp:GridView>
</asp:Content>

```

Στη σελίδα Desktop.aspx.vb περιέχεται ο κώδικας με τον οποίο λειτουργούν τα φίλτρα και το gridview για την εμφάνιση των προϊόντων. Αρχικά δηλώνουμε ένα string με SQL query που θα χρησιμοποιείται σε όλη τη σελίδα για τις ανάγκες κάθε φίλτρου. Από αυτήν την πρόταση λείπει το τέλος. Σε κάθε φίλτρο που θα συναντάμε θα προσθέτουμε και την κατάλληλη έκφραση έτσι ώστε να έχουμε το επιθυμητό αποτέλεσμα.

```

Public menuString As String = "SELECT DISTINCT tblProduct.ProductID,
tblProduct.Name, tblProduct.Price, tblProduct.Image, tblManufacturer.Name AS
Manufacturer
FROM tblProduct
INNER JOIN tblManufacturer ON tblProduct.ManufacturerID =
tblManufacturer.ManufacturerID
INNER JOIN tblSubCategory ON tblProduct.SubCategoryID =
tblSubCategory.SubCategoryID
INNER JOIN tblCategory ON tblSubCategory.CategoryID = tblCategory.CategoryID
INNER JOIN tblSubCategoryCharacteristics ON tblSubCategory.SubCategoryID =
tblSubCategoryCharacteristics.SubCategoryID
INNER JOIN tblProductCharacteristics ON tblProduct.ProductID =
tblProductCharacteristics.ProductID AND
tblSubCategoryCharacteristics.CharacteristicsID =
tblProductCharacteristics.CharacteristicsID"

```

Όταν φορτώνει πρώτη φορά η σελίδα θέλουμε να εμφανίζονται όλα τα προϊόντα της κατηγορίας Desktop. Άρα η πρόταση επιλογής είναι το παραπάνω string με κλείσιμο της πρότασης τον περιορισμό WHERE (tblCategory.Name = 'Pc'). Αποθηκεύουμε το τελικό string σε ένα session και το βάζουμε στο DataSource του gridView μας με ταξινόμηση σύμφωνα με την τιμή των προϊόντων.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    If Not IsPostBack Then
        Dim con As New SqlConnection
        (ConfigurationManager.AppSettings("CONNECTION_STRING"))
        Dim cmd, cmdMan, cmdHdd, cmdRam, cmdScreen As New SqlCommand
        Dim dr As SqlDataReader
        Session("SubCatFilter") = Nothing
        Session("ManFilter") = Nothing
        Session("PriceFilter") = Nothing
        Session("Filter1") = Nothing
        Session("Filter2") = Nothing
        Session("Filter3") = Nothing
        Session("Filter4") = Nothing
        Try
            con.Open()
            Session("CatFilter") = menuString & "WHERE (tblCategory.Name =
'Pc')"
```

```
            Session("FILTER") = Session("CatFilter")
            SqlDataAdapter1.SelectCommand = Session("FILTER") & " ORDER BY
tblProduct.Price"
```

Οι λίστες των φίλτρων γεμίζουν δυναμικά. Δηλαδή όταν τρέξει η σελίδα, θα πάει στη βάση να ελέγξει ποιες υποκατηγορίες βρίσκονται σε αυτή την κατηγορία, ποιои κατασκευαστές και ποιες τιμές χαρακτηριστικών υπάρχουν στα προϊόντα. Αφού τα βρει, επιστρέφει τα αποτελέσματα στις λίστες μέσα στον πίνακα φίλτρων.

```
cmd.CommandType = CommandType.StoredProcedure
cmd.CommandText = "category_sel"
cmd.Connection = con
cmd.Parameters.AddWithValue("@CategoryParam", "Pc")
dr = cmd.ExecuteReader
SubCategoryFilter.Items.Clear()
While dr.Read()
    Dim itemi As New MenuItem
    itemi.Text = dr("Name")
    SubCategoryFilter.Items.Add(itemi)
End While
dr.Close()

cmdMan.CommandType = CommandType.StoredProcedure
cmdMan.CommandText = "manufacturer_sel"
cmdMan.Connection = con
cmdMan.Parameters.AddWithValue("@CategoryParam", "Pc")
dr = cmdMan.ExecuteReader
ManufacturerFilter.Items.Clear()
While dr.Read()
    Dim itemi As New MenuItem
    itemi.Text = dr("Name")
    ManufacturerFilter.Items.Add(itemi)
End While
dr.Close()

cmdHdd.CommandType = CommandType.StoredProcedure
cmdHdd.CommandText = "subcategoryChar_sel"
cmdHdd.Connection = con
```

```

cmdHdd.Parameters.AddWithValue("@CategoryParam", "Pc")
cmdHdd.Parameters.AddWithValue("@SubCategoryParam", "HDD")
dr = cmdHdd.ExecuteReader
HDDFilter.Items.Clear()
While dr.Read()
    Dim itemi As New MenuItem
    itemi.Text = dr("Value")
    HDDFilter.Items.Add(itemi)
End While
dr.Close()

cmdRam.CommandType = CommandType.StoredProcedure
cmdRam.CommandText = "subcategoryChar_sel"
cmdRam.Connection = con
cmdRam.Parameters.AddWithValue("@CategoryParam", "Pc")
cmdRam.Parameters.AddWithValue("@SubCategoryParam", "RAM")
dr = cmdRam.ExecuteReader
RAMFilter.Items.Clear()
While dr.Read()
    Dim itemi As New MenuItem
    itemi.Text = dr("Value")
    RAMFilter.Items.Add(itemi)
End While
dr.Close()

cmdScreen.CommandType = CommandType.StoredProcedure
cmdScreen.CommandText = "subcategoryChar_sel"
cmdScreen.Connection = con
cmdScreen.Parameters.AddWithValue("@CategoryParam", "Pc")
cmdScreen.Parameters.AddWithValue("@SubCategoryParam", "Screen
Size")

dr = cmdScreen.ExecuteReader
MonitorFilter.Items.Clear()
While dr.Read()
    Dim itemi As New MenuItem
    itemi.Text = dr("Value")
    MonitorFilter.Items.Add(itemi)
End While
dr.Close()
Catch ex As Exception
    Session("MESSAGE") = ex.ToString
    Response.Redirect("~/Message.aspx", False)
Finally
    cmd.Dispose()
    cmdHdd.Dispose()
    cmdMan.Dispose()
    cmdRam.Dispose()
    cmdScreen.Dispose()
    con.Close()
End Try
End If
End Sub

```

Τι γίνεται όταν θέλουμε να περιορίσουμε την αναζήτηση μας; Οι λίστες φίλτρων τώρα έχουν γεμίσει κι έτσι μπορούμε να δούμε ποιες υποκατηγορίες υπάρχουν στην κατηγορία Desktop. Όταν πατηθεί ένα αντικείμενο μιας λίστας, ελέγχουμε πρώτα απ' όλα αν έχει ξαναπατηθεί από τη συγκεκριμένη λίστα ένα άλλο αντικείμενο. Όταν δεν έχει χρησιμοποιηθεί η λίστα τότε προσθέτουμε στο αρχικό query την τομή όλης της κατηγορίας με το αντικείμενο που επιλέξαμε. Αν έχει χρησιμοποιηθεί αντικείμενο από αυτή τη λίστα οι επιλογές είναι δυο. Να είναι το ίδιο επιλεγμένο αντικείμενο, όποτε το αγνοούμε, ή να είναι ένα διαφορετικό όποτε στο προηγούμενο string θα προσθέσουμε τον όρο OR για να βλέπουμε προϊόντα από δυο η και παραπάνω υποκατηγορίες. Το

τελικό string αποθηκεύεται σε session ξεχωριστό από αυτό που θα χρησιμοποιηθεί για το DataSource του gridView. Τέλος, στο session που θα χρησιμοποιηθεί για πηγή βάσης προσθέτουμε όλα τα φίλτρα (υποκατηγορίας, κατασκευαστή, μνήμης RAM κτλ).

Ο κώδικας για τα φίλτρα είναι όμοιος για κάθε κατηγορία φίλτρου, με εξαίρεση αυτού του εύρους τιμής όπου έχουμε προκαθορισμένες τιμές, και έχει ως εξής:

```
Protected Sub SubCategoryFilter_MenuItemClick(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.MenuEventArgs) Handles SubCategoryFilter.MenuItemClick
    If Session("SubCatFilter") = Nothing Then
        Session("SubCatFilter") = " INTERSECT " & menuString & " WHERE (tblSubCategory.Name = '" & SubCategoryFilter.SelectedItem.Text & "')"
    Else
        Session("SubCatFilter") = Session("SubCatFilter") & " OR (tblSubCategory.Name = '" & SubCategoryFilter.SelectedItem.Text & "')"
    End If
    Session("FILTER") = Session("CatFilter") & Session("SubCatFilter") & Session("ManFilter") & Session("PriceFilter") & session("Filter1") & session("Filter2") & session("Filter3") & session("Filter4")
    SqlDataSource1.SelectCommand = Session("FILTER") & " ORDER BY tblProduct.Price"
    GridView1.PageIndex = 0
    If lblChoose.Visible = False Then
        lblChoose.Visible = True
        chooseMenu.Visible = True
    End If
    Dim item As New MenuItem
    item.Text = SubCategoryFilter.SelectedItem.Text
    chooseMenu.Items.Add(item)
End Sub
```

Όταν θέλουμε να καθαρίσουμε τα φίλτρα και να επιστρέψουμε στο αρχικό, όπου φαίνονται όλα τα προϊόντα της κατηγορίας, πατάμε το κουμπί Καθαρισμός Φίλτρων και εκτελείται ο παρακάτω κώδικας.

```
Protected Sub lbtnClear_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles lbtnClear.Click
    Response.Redirect("~/Desktop.aspx", False)
    GridView1.PageIndex = 0
    Session("SubCatFilter") = Nothing
    Session("ManFilter") = Nothing
    Session("PriceFilter") = Nothing
    session("Filter1") = Nothing
    session("Filter2") = Nothing
    session("Filter3") = Nothing
    session("Filter4") = Nothing
    lblChoose.Visible = False
    chooseMenu.Items.Clear()
    chooseMenu.Visible = False
End Sub
```

Επειδή δεν μπορούμε να δείχνουμε όλα τα προϊόντα σε μια σελίδα τα παρουσιάζουμε πέντε προϊόντα ανά σελίδα στο gridView. Η ταξινόμηση γίνεται με βάση την τιμή του προϊόντος. Κάθε φορά που αλλάζει η σελίδα του gridView πρέπει να ενημερώνεται η πηγή της βάσης και να ξέρει σε ποια σελίδα βρίσκεται. Αυτό το καταφέρνουμε με το παρακάτω κομμάτι.

```
Protected Sub GridView1_PageIndexChanging(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.GridViewPageEventArgs) Handles GridView1.PageIndexChanging
    SqlDataSource1.SelectCommand = Session("FILTER") & " ORDER BY tblProduct.Price"
```

```

GridView1.PageIndex = e.NewPageIndex
End Sub

```

Σε κάθε σειρά στο gridView έχουμε δυο εντολές. Μια για όταν επιλέγουμε ένα προϊόν, να πηγαίνει στη σελίδα Product.aspx και να βλέπουμε τις λεπτομέρειες κάθε προϊόντος και μια για να προσθέτουμε το προϊόν στο καλάθι αγοράς. Ο τρόπος για την επίτευξη του σκοπού αυτού γίνεται μέσω του παρακάτω κώδικα.

```

Protected Sub GridView1_RowCommand(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.GridViewCommandEventArgs) Handles GridView1.RowCommand
    Dim currentCommand As String = e.CommandName
    Dim currentRowIndex As Integer = Int32.Parse(e.CommandArgument)
    If currentCommand = "ProductName" Then
        Session("Product") = GridView1.DataKeys(currentRowIndex).Value
        Response.Redirect("~/Product.aspx", False)
    ElseIf currentCommand = "AddtoCart" Then
        Session("Product") = GridView1.DataKeys(currentRowIndex).Value
        ShoppingCart.Instance.AddItem(Session("Product"))
        Response.Redirect("~/ViewCart.aspx", False)
    End If
End Sub

```

7.4 Hardware.aspx

Μόλις επιλεγεί μια υποκατηγορία γίνεται η αναζήτηση των κατασκευαστών, των χαρακτηριστικών και των τιμών των χαρακτηριστικών της υποκατηγορίας αυτής. Στη συνέχεια γεμίζονται οι λίστες για να μπορούμε να επιλέξουμε επιπλέον φίλτρα για τον περιορισμό της αναζήτησης μας. Όλα γίνονται μέσα στην υπορουτίνα επιλογής υποκατηγορίας SubCategoryFilter_MenuItemClick.

```

Protected Sub SubCategoryFilter_MenuItemClick(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.MenuEventArgs) Handles SubCategoryFilter.MenuItemClick
    If Session("SubCatFilter") = Nothing Then
        Session("SubCatFilter") = " INTERSECT " & menuString & " WHERE
(tblSubCategory.Name = ' " & SubCategoryFilter.SelectedItem.Text & " )"
    Else
        Session("SubCatFilter") = Session("SubCatFilter") & " OR
(tblSubCategory.Name = ' " & SubCategoryFilter.SelectedItem.Text & " )"
    End If
    Session("FILTER") = Session("CatFilter") & Session("SubCatFilter") &
Session("ManFilter") & Session("PriceFilter") & Session("Filter1") &
Session("Filter2") _
        & Session("Filter3") & Session("Filter4")
    SqlDataSource1.SelectCommand = Session("FILTER") & " ORDER BY
tblProduct.Price"
    GridView1.PageIndex = 0
    SubCategoryFilter.Enabled = False
    Select Case SubCategoryFilter.SelectedItem.Text
        Case "Cases"
            Label1.Text = "Τύπος"
            Label1.Visible = True
            Label2.Visible = False
            Label3.Visible = False
            Label4.Visible = False

            Fill1("Cases", "Type")
            Filter1.Visible = True
            Filter2.Visible = False
            Filter3.Visible = False
            Filter4.Visible = False

```

```
Case "Psu"
    Label1.Text = "Τύπος"
    Label1.Visible = True
    Label2.Visible = False
    Label3.Visible = False
    Label4.Visible = False

    Fill1("Psu", "Type")
    Filter1.Visible = True
    Filter2.Visible = False
    Filter3.Visible = False
    Filter4.Visible = False

Case "Cpu"
    Label1.Text = "Chip"
    Label2.Text = "Σύνδεση με motherboard"
    Label1.Visible = True
    Label2.Visible = True
    Label3.Visible = False
    Label4.Visible = False

    Fill1("Cpu", "Chip")
    Filter1.Visible = True
    Fill2("Cpu", "Socket")
    Filter2.Visible = True
    Filter3.Visible = False
    Filter4.Visible = False

Case "Motherboard"
    Label1.Text = "Chipset"
    Label2.Text = "Σύνδεση με CPU"
    Label3.Text = "Σύνδεση με μνήμη RAM"
    Label4.Text = "Size"
    Label1.Visible = True
    Label2.Visible = True
    Label3.Visible = True
    Label4.Visible = True

    Fill1("Motherboard", "Chipset")
    Filter1.Visible = True
    Fill2("Motherboard", "Cpu Socket")
    Filter2.Visible = True
    Fill3("Motherboard", "Ram Socket")
    Filter3.Visible = True
    Fill4("Motherboard", "Size")
    Filter4.Visible = True

Case "HardDisk"
    Label1.Text = "Τύπος"
    Label2.Text = "Διαστάσεις"
    Label3.Text = "Μέγεθος"
    Label4.Text = "Επιπλέον χαρακτηριστικά"
    Label1.Visible = True
    Label2.Visible = True
    Label3.Visible = True
    Label4.Visible = True

    Fill1("HardDisk", "Type")
    Filter1.Visible = True
    Fill2("HardDisk", "Dimensions")
    Filter2.Visible = True
    Fill3("HardDisk", "Size")
    Filter3.Visible = True
    Fill4("HardDisk", "Characteristics")
    Filter4.Visible = True
```

```
Case "RAM"
Label1.Text = "Τύπος"
Label2.Text = "Μέγεθος"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = False
Label4.Visible = False

Fill1("RAM", "Type")
Filter1.Visible = True
Fill2("RAM", "Size")
Filter2.Visible = True
Filter3.Visible = False
Filter4.Visible = False
Case "Graphics Cards"
Label1.Text = "Chipset"
Label2.Text = "Σύνδεση"
Label3.Text = "Χωρητικότητα Μνήμης"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
Label4.Visible = False

Fill1("Graphics Cards", "Chipset")
Filter1.Visible = True
Fill2("Graphics Cards", "Connection")
Filter2.Visible = True
Fill3("Graphics Cards", "Capacity")
Filter3.Visible = True
Filter4.Visible = False
Case "Sound"
Label1.Text = "Σύνδεση"
Label1.Visible = True
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False

Fill1("Sound", "Connection")
Filter1.Visible = True
Filter2.Visible = False
Filter3.Visible = False
Filter4.Visible = False
Case "DVD-Bluray"
Label1.Text = "Τύπος"
Label1.Visible = True
Label2.Visible = False
Label3.Visible = False
Label4.Visible = False

Fill1("DVD-Bluray", "Type")
Filter1.Visible = True
Filter2.Visible = False
Filter3.Visible = False
Filter4.Visible = False
Case "Monitor"
Label1.Text = "Μέγεθος"
Label2.Text = "Μέγιστη ανάλυση"
Label3.Text = "Συνδεσιμότητα"
Label4.Text = "Χαρακτηριστικά"
Label1.Visible = True
Label2.Visible = True
Label3.Visible = True
```



```

        Label4.Visible = True

        Fill1("Monitor", "Size")
        Filter1.Visible = True
        Fill2("Monitor", "Pixels")
        Filter2.Visible = True
        Fill3("Monitor", "Connection")
        Filter3.Visible = True
        Fill4("Monitor", "Characteristics")
        Filter4.Visible = True
    Case "Mouse/Keyboard"
        Label1.Text = "Σύνδεση"
        Label2.Text = "Χαρακτηριστικά"
        Label1.Visible = True
        Label2.Visible = True
        Label3.Visible = False
        Label4.Visible = False

        Fill1("Mouse/Keyboard", "Connection")
        Filter1.Visible = True
        Fill2("Mouse/Keyboard", "Characteristics")
        Filter2.Visible = True
        Filter3.Visible = False
        Filter4.Visible = False
    End Select
Dim con As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STRING"))
Dim cmdMan As New SqlCommand
Dim dr As SqlDataReader
Try
    con.Open()
    cmdMan.CommandType = CommandType.StoredProcedure
    cmdMan.CommandText = "manufacturer_subcat_sel"
    cmdMan.Connection = con
    cmdMan.Parameters.AddWithValue("@CategoryParam", "Hardware")
    cmdMan.Parameters.AddWithValue("@SubCategoryParam",
SubCategoryFilter.SelectedItem.Text)
    dr = cmdMan.ExecuteReader
    ManufacturerFilter.Items.Clear()
    While dr.Read()
        Dim item1 As New MenuItem
        item1.Text = dr("Name")
        ManufacturerFilter.Items.Add(item1)
    End While
    dr.Close()
Catch ex As Exception
    Session("MESSAGE") = ex.ToString
    Response.Redirect("~/Message.aspx", True)
Finally
    cmdMan.Dispose()
    con.Close()
End Try
Dim item, item2 As New MenuItem
item.Text = SubCategoryFilter.SelectedItem.Text
item2.Text = SubCategoryFilter.SelectedItem.Text
SubCategoryFilter.Items.Clear()
SubCategoryFilter.Items.Add(item)
chooseMenu.Items.Add(item2)
lblManufacturer.Visible = True
lblChoose.Visible = True
lblPrice.Visible = True
PriceFilter.Visible = True

```

```

        chooseMenu.Visible = True
    End Sub

```

Τα χαρακτηριστικά κάθε προϊόντος είναι το πολύ τέσσερα, οπότε έχουμε φτιάξει τέσσερις διαφορετικές συναρτήσεις για να γεμίσουν την κάθε λίστα για τα φίλτρα. Όλες οι λίστες γεμίζουν δυναμικά, με εξαίρεση αυτή του επεξεργαστή. Στην πρώτη λίστα του επεξεργαστή δίνουμε προεπιλεγμένες τιμές για το chipset του επεξεργαστή.

```

Protected Sub Fill1(ByVal SubCatName As String, ByVal CharName As String)
    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim dr As SqlDataReader
    If SubCatName = "Cpu" Then
        Filter1.Items.Clear()
        Dim item1 As New MenuItem
        item1.Text = "Celeron"
        Filter1.Items.Add(item1)
        Dim item2 As New MenuItem
        item2.Text = "Dual Core"
        Filter1.Items.Add(item2)
        Dim item3 As New MenuItem
        item3.Text = "Opteron"
        Filter1.Items.Add(item3)
        Dim item4 As New MenuItem
        item4.Text = "Phenom"
        Filter1.Items.Add(item4)
        Dim item5 As New MenuItem
        item5.Text = "Core i3"
        Filter1.Items.Add(item5)
        Dim item6 As New MenuItem
        item6.Text = "Core i5"
        Filter1.Items.Add(item6)
        Dim item7 As New MenuItem
        item7.Text = "Core i7"
        Filter1.Items.Add(item7)
        Dim item8 As New MenuItem
        item8.Text = "Athlon"
        Filter1.Items.Add(item8)
    Else
        Try
            con.ConnectionString =
ConfigurationManager.AppSettings("CONNECTION_STRING")
            con.Open()
            cmd.CommandType = CommandType.StoredProcedure
            cmd.CommandText = "productvalue_sel"
            cmd.Connection = con
            With cmd
                .Parameters.AddWithValue("@CategoryParam", "Hardware")
                .Parameters.AddWithValue("@SubCategoryParam", SubCatName)
                .Parameters.AddWithValue("@SubCatCharacteristic", CharName)
            End With
            dr = cmd.ExecuteReader
            Filter1.Items.Clear()
            While dr.Read()
                Dim itemi As New MenuItem
                itemi.Text = dr("Value")
                Filter1.Items.Add(itemi)
            End While
            dr.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
        Finally

```

```

        cmd.Dispose()
        con.Close()
    End Try
End If
End Sub

```

Όταν ο χρήστης επιλέξει ένα από τα αντικείμενα που βρίσκονται στις λίστες εκτελείται παρόμοιος κώδικας με τη σελίδα Desktop.aspx.vb. Η διαφορά είναι πάλι στον επεξεργαστή όπου μέσα στον πίνακα έκτος από την τιμή που έχει για την ταχύτητα έχουμε βάλει και το όνομα του chipset. Εφ' όσων επιλέξουμε το όνομα ενός επεξεργαστή πρέπει στην αναζήτηση του να χρησιμοποιήσουμε την έκφραση SQL "like". Με αυτήν την έκφραση δε δημιουργείται πρόβλημα στις υπόλοιπες υποκατηγορίες, οπότε χρησιμοποιούμε τον ίδιο κώδικα για όλα τα φίλτρα.

```

Protected Sub Filter1_MenuItemClick(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.MenuEventArgs) Handles Filter1.MenuItemClick
    If Session("Filter1") = Nothing Then
        Session("Filter1") = " INTERSECT " & menuString & " WHERE
(tblProductCharacteristics.Value like '%" & Filter1.SelectedItem.Text & "%')
    Else
        Session("Filter1") = Session("Filter1") & " OR
(tblProductCharacteristics.Value like '%" & Filter1.SelectedItem.Text & "%')
    End If
    Session("FILTER") = Session("CatFilter") & Session("SubCatFilter") &
Session("ManFilter") & Session("PriceFilter") & Session("Filter1") &
Session("Filter2") _
        & Session("Filter3") & Session("Filter4")
    SqlDataSource1.SelectCommand = Session("FILTER") & " ORDER BY
tblProduct.Price"
    GridView1.PageIndex = 0
    Dim item As New MenuItem
    item.Text = Filter1.SelectedItem.Text
    chooseMenu.Items.Add(item)
End Sub

```

7.5 Search.aspx

Σχεδιαστικά αυτό που θέλουμε από αυτή τη σελίδα είναι το gridView, όπου θα βλέπουμε τα αποτελέσματα της αναζήτησής μας. Το τοποθετούμε σε ένα panel και χρησιμοποιούμε τις ίδιες ακριβώς εντολές γραμμής που είδαμε και στο gridView της Desktop.aspx.

```

<asp:Content ID="Content2" ContentPlaceHolderID="main" Runat="Server">
<div style="width: 750px; margin-top:15px">
    <asp:Panel ID="Panel1" runat="server" Width="750px" ScrollBars="None"
Height="460px" >
        <asp:GridView ID="GridView1" runat="server" AllowPaging="true"
AutoGenerateColumns="False" EmptyDataText="Δεν υπάρχουν προϊόντα με αυτά τα
δεδομένα"
        PageSize="5" CellPadding="4" ForeColor="#333333" GridLines="None"
        DataSourceID = "SqlDataSource1" DataKeyNames="ProductID">
        <AlternatingRowStyle BackColor="White" />
        <Columns>
            <asp:ImageField DataImageUrlField="Image" HeaderText="Εικόνα">
                <ControlStyle Height="65px" Width="80px" />
            </asp:ImageField>
            <asp:ButtonField DataTextField="Name" HeaderText="Όνομα"
SortExpression="Name"
                CommandName="ProductName">
            <ItemStyle HorizontalAlign="Center" Width="280px" />
            </asp:ButtonField>
            <asp:BoundField DataField="Manufacturer" HeaderText="Κατασκευαστής"

```

```

        SortExpression="Manufacturer" >
        <ItemStyle HorizontalAlign="Center" Width="105px" />
        </asp:BoundField>
        <asp:BoundField DataField="Price" HeaderText="Τιμή"
        SortExpression="Price" DataFormatString="{0:C}">
        <ItemStyle HorizontalAlign="Center" Width="75px" />
        </asp:BoundField>
        <asp:ButtonField CommandName="AddtoCart" Text="Προσθήκη στο Καλάθι" >
        <ItemStyle HorizontalAlign="Right" Width="150px" />
        </asp:ButtonField>
    </Columns>
    <FooterStyle BackColor="#990000" Font-Bold="True" ForeColor="White" />
    <HeaderStyle BackColor="#CB4848" Font-Bold="True" ForeColor="White" />
    <PagerStyle BackColor="#DC5050" ForeColor="#333333"HorizontalAlign="Center"
/>

    <RowStyle BackColor="#DCDCDC" ForeColor="#333333" />
    <SelectedRowStyle BackColor="#acacac" Font-Bold="True" ForeColor="Navy" />
    <SortedAscendingCellStyle BackColor="#acacac" />
    <SortedAscendingHeaderStyle BackColor="#4D0000" />
    <SortedDescendingCellStyle BackColor="#acacac" />
    <SortedDescendingHeaderStyle BackColor="#820000" />
</asp:GridView>
</asp:Panel>
</div>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
    ConnectionString="<%$ ConnectionStrings:myEshopDBConnectionString %>"
    ProviderName="<%$ ConnectionStrings:myEshopDBConnectionString.ProviderName
%>">
    </asp:SqlDataSource>
</asp:Content>

```

Στον κώδικα μας ενδιαφέρει η αναζήτηση και η προβολή όλων των προϊόντων που έχουν στο όνομα τους λέξεις ή και γράμματα από την λέξη που έχει δοθεί από το χρήστη. Μόλις φορτώσει η σελίδα Search.aspx παίρνει τη λέξη που βρίσκεται στο Session("SEARCH") και ελέγχει ποια προϊόντα έχουν αυτή τη λέξη στο όνομα τους. Το αποτέλεσμα γίνεται η πηγή της βάσης του gridview και προβάλλεται στην οθόνη.

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    SqlDataSource1.SelectCommand = "SELECT tblProduct.ProductID,
tblProduct.Name, tblProduct.Price, tblProduct.Image, tblManufacturer.Name AS
Manufacturer FROM tblProduct
INNER JOIN tblManufacturer ON tblProduct.ManufacturerID =
tblManufacturer.ManufacturerID
WHERE (tblProduct.Name like '%" & Session("SEARCH") & "%')"
```

Σε αυτό το gridview δεν είναι αναγκαίο να κρατάμε το page index και να κάνουμε bind το κομμάτι της βάσης που θέλουμε, γιατί απλά δεν αλλάζει η πηγή και οι σελίδες του gridview ενημερώνονται αυτόματα.

7.6Product.aspx

Για το σχεδιασμό της σελίδας αυτής χρησιμοποιούμε ένα ImageBox για την εικόνα του προϊόντος, και δίπλα ακριβώς, ένα Label όπου αναγράφεται το όνομα του. Κάτω από την εικόνα έχουμε βάλει πάλι Labels, για την περιγραφή και για τα χαρακτηριστικά του προϊόντος. Για καλύτερη ταξινόμηση στη σελίδα τα Labels των χαρακτηριστικών είναι τοποθετημένα σε πίνακα. Τέλος, υπάρχει στο κάτω μέρος το κουμπί προσθήκη στο καλάθι.

```

<asp:Content ID="BodyContent" ContentPlaceHolderID="main" Runat="Server">
<div style="width: 720px; padding-left:20px; margin-top:15px">
  <asp:Panel ID="Panel1" runat="server" Width="720px" ScrollBars="None"
Height="460px" >
  <asp:Image ID="Image1" runat="server" Height="150px" ImageAlign="Middle"
Width="190px" />
  <asp:Label ID="lblName" runat="server"></asp:Label>
  <p>
  <asp:Label ID="lblDesc" runat="server"></asp:Label>
  </p>
  <asp:Table ID="Table1" runat="server" CellPadding="5">
  <asp:TableRow>
  <asp:TableCell>
  <asp:Label ID="lblChar1" runat="server" ></asp:Label>
  </asp:TableCell>
  <asp:TableCell>
  <asp:Label ID="lblData1" runat="server" ></asp:Label>
  </asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
  <asp:TableCell>
  <asp:Label ID="lblChar2" runat="server" ></asp:Label>
  </asp:TableCell>
  <asp:TableCell>
  <asp:Label ID="lblData2" runat="server" ></asp:Label>
  </asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
  <asp:TableCell>
  <asp:Label ID="lblChar3" runat="server" ></asp:Label>
  </asp:TableCell>
  <asp:TableCell>
  <asp:Label ID="lblData3" runat="server" ></asp:Label>
  </asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
  <asp:TableCell>
  <asp:Label ID="lblChar4" runat="server" ></asp:Label>
  </asp:TableCell>
  <asp:TableCell>
  <asp:Label ID="lblData4" runat="server" ></asp:Label>
  </asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
  <asp:TableCell>
  Τιμή:
  </asp:TableCell>
  <asp:TableCell>
  <asp:Label ID="lblPrice" runat="server"></asp:Label>
  </asp:TableCell>
  </asp:TableRow>
  </asp:Table>
  <p>
  <asp:LinkButton ID="LinkButton1" runat="server" style="float:right; margin-
right:25px;">Προσθήκη στο καλάθι</asp:LinkButton>
  </p>
  </asp:Panel>
</div>
</asp:Content>

```

Όσον αφορά το προγραμματιστικό κομμάτι της σελίδας αυτής, αυτό που κάνουμε είναι να αναζητήσουμε το προϊόν που έχει ζητήσει από άλλη σελίδα ο χρήστης και απλά να προβάλουμε μέσω

εικόνων και ταμπελών τις πληροφορίες του προϊόντος. Αρχικά ανοίγουμε σύνδεση με τη βάση και στη συνέχεια αναζητάμε το προϊόν με τον αύξοντα αριθμό του που βρίσκεται στο Session("Product"). Όταν βρεθεί τοποθετούμε το όνομα, την περιγραφή, την τιμή και την εικόνα στα κατάλληλα σημεία. Για τα χαρακτηριστικά αποθηκεύουμε πρώτα το όνομα των χαρακτηριστικών σε ένα πίνακα και την τιμή σε ένα δεύτερο πίνακα. Μετα από τους πίνακες παίρνουμε τις τιμές και τις βάζουμε στα σωστά labels. Αυτό γίνεται γιατί ο αριθμός των χαρακτηριστικών διαφέρει από το ένα προϊόν στο άλλο.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
```

```
Dim con As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STRING"))
Dim cmd As New SqlCommand
Dim dr As SqlDataReader
Dim pin1(4), pin2(4) As String
Dim i As Integer = 0

lblChar1.Text = ""
lblData1.Text = ""
lblChar2.Text = ""
lblData2.Text = ""
lblChar3.Text = ""
lblData3.Text = ""
lblChar4.Text = ""
lblData4.Text = ""

cmd.CommandType = Data.CommandType.StoredProcedure
cmd.CommandText = "product_sel"
cmd.Connection = con
cmd.Parameters.AddWithValue("@Product", Session("Product"))
```

```
Try
```

```
con.Open()
dr = cmd.ExecuteReader
While dr.Read()
    lblName.Text = dr("Name")
    lblDesc.Text = dr("Description")
    lblPrice.Text = Convert.ToDecimal(dr("Price")).ToString("C")
    pin1(i) = dr("CharName")
    pin2(i) = dr("Value")
    Image1.ImageUrl = dr("Image")
    i = i + 1
End While
dr.Close()
Catch ex As Exception
    MsgBox(ex.ToString)
Finally
    cmd.Dispose()
    con.Close()
End Try
If Not pin1(0) = Nothing Then
    lblChar1.Text = pin1(0) & ": "
    lblData1.Text = pin2(0)
End If
If Not pin1(1) = Nothing Then
    lblChar2.Text = pin1(1) & ": "
    lblData2.Text = pin2(1)
End If
If Not pin1(2) = Nothing Then
    lblChar3.Text = pin1(2) & ": "
    lblData3.Text = pin2(2)
End If
```

```

    If Not pin1(3) = Nothing Then
        lblChar4.Text = pin1(3) & ": "
        lblData4.Text = pin2(3)
    End If
End Sub

```

Τέλος, για το κουμπί «προσθήκη στο καλάθι», καλούμε τη συνάρτηση που προσθέτει το προϊόν στο καλάθι (θα δούμε αργότερα πως) και μεταβιβάζει τον χρήστη αμέσως στη σελίδα του καλαθιού.

```

Protected Sub LinkButton1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LinkButton1.Click
    ShoppingCart.Instance.AddItem(Session("Product"))
    Response.Redirect("~/ViewCart.aspx", False)
End Sub

```

7.7 Login.aspx

Ο κώδικας για την εμφάνιση της σελίδας:

```

<asp:Content ID="body" ContentPlaceHolderID="main" runat="server">
    <h2 style="font-variant: small-caps; text-align: center">Log in</h2>

    <p style="padding-left:40px">
        Εισάγετε Όνομα χρήστη και Κωδικό.<br /><br />
        <a href="Register.aspx">Εγγραφείτε τώρα</a> αν δεν έχετε λογαριασμό.
    </p>
    <div style="width: 600px; padding-left:40px">
        <asp:Panel ID="Panel1" runat="server" Width="600px" ScrollBars="None"
Height="335px" DefaultButton="LoginButton">
            <fieldset class="login">
                <legend><b>Είσοδος</b></legend>
                <p>
                    <asp:Label ID="lblUserNameLog" runat="server">e-
mail:</asp:Label><br/>
                    <asp:TextBox ID="UserName" runat="server" CssClass="textEntry"
Width="500px" TabIndex="0"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="UserNameRequired"
runat="server"
                        ControlToValidate="UserName"
                        CssClass="failureNotification"
                        ErrorMessage="User Name is
required.">*</asp:RequiredFieldValidator>
                </p>
                <br />
                <p>
                    <asp:Label ID="lblPass"
runat="server">Κωδικός:</asp:Label><br/>
                    <asp:TextBox ID="Password" runat="server"
                        CssClass="passwordEntry"
                        TextMode="Password" Width="500px"
                        TabIndex="1"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="PasswordRequired"
runat="server"
                        ControlToValidate="Password"
                        CssClass="failureNotification"
                        ErrorMessage="Password is
required.">*</asp:RequiredFieldValidator>
                </p>
                <p class="btnSubmit">

```

```

        <asp:Button ID="LoginButton" runat="server" CommandName="Login"
Text="Log In" TabIndex="2"/>
    </p>
    <asp:ValidationSummary ID="LoginSummary" runat="server"
CssClass="failureNotification" />
</fieldset>
</asp:Panel>
</div>
</asp:Content>

```

Αν υπάρχει ήδη συνδεδεμένος χρήστης, αυτόματα μας μεταφέρει στη σελίδα Message.aspx όπου μας αναφέρει πως κάποιος είναι ήδη συνδεδεμένος.

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    ClientScript.RegisterHiddenField("__EVENTTARGET", "LoginButton")
    If Session("CustomerID") > 0 Then
        Session("MESSAGE") = "Ο χρήστης είναι ήδη συνδεδεμένος"
        Response.Redirect("~/Message.aspx", True)
    End If
End Sub

```

Αλλιώς, ο χρήστης βάζει κανονικά τα στοιχεία του και μόλις πατήσει το κουμπί για να εισέρθει στο λογαριασμό του ελέγχεται αν υπάρχει ο λογαριασμός αυτός αν ο κωδικός που έδωσε είναι σωστός και ανάλογα είτε γίνεται η είσοδος είτε πηγαίνει στη σελίδα Message.aspx και αναφέρει το λάθος.

```

Protected Sub LoginButton_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles LoginButton.Click
    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim dr As SqlDataReader

    Try
        con.ConnectionString =
ConfigurationManager.AppSettings("CONNECTION_STRING")
        con.Open()
        cmd = New SqlCommand("SELECT CustomerID FROM tblCustomer " & _
& "WHERE Email = '" & UserName.Text & "' " & _
& "AND Password = '" & Password.Text & "'", con)
        dr = cmd.ExecuteReader(Data.CommandBehavior.SequentialAccess)
        If dr.Read Then
            If UserName.Text = "admin@ecenter.gr" Then
                Response.Redirect("~/Admin.aspx", False)
            Else
                Response.Redirect("~/Default.aspx", False)
            End If
            Session("CustomerID") = dr.GetValue(0).ToString.Trim
        Else
            Session("MESSAGE") = "Λάθος Κωδικός ή e-mail"
            Response.Redirect("~/Message.aspx", False)
        End If
        dr.Close()
    Catch ex As Exception
        Session("MESSAGE") = ex.ToString
        Response.Redirect("~/Message.aspx", False)
    Finally
        cmd.Dispose()
        con.Close()
    End Try
End Sub

```


7.8 Register.aspx

Η φόρμα εγγραφής αποτελείται από τα απαραίτητα στοιχεία που χρειάζονται για να σταλούν τα προϊόντα στη διεύθυνση του πελάτη. Τα στοιχεία αυτά είναι: όνομα, επώνυμο, e-mail, κωδικός, πόλη, διεύθυνση, ταχυδρομικός κώδικας σταθερό και κινητό τηλέφωνο. Όποιος πελάτης θέλει να κάνει αγορές για την επιχείρηση του και θέλει να κοπεί τιμολόγιο αντί απόδειξης, πατάει στο κουτί του τιμολογίου, εμφανίζονται κι άλλα πεδία για συμπλήρωση, που αυτή τη φορά έχουν να κάνουν με την εταιρία. Για την επίτευξη της παραπάνω εικόνας γράψαμε τον παρακάτω κώδικα:

```
<asp:Content ID="body" ContentPlaceHolderID="main" runat="server">
  <div style="width: 600px; padding-left:40px">
    <asp:Panel runat="server" Width="680px" ScrollBars="Vertical"
Height="475px" DefaultButton="btnSignUp">
      <div style="width: 405px;">
        <fieldset class="login" style="float:left;">
          <legend><b>Εγγραφή</b></legend>
          <p>
            <asp:Label ID="lblFName" runat="server">Όνομα:</asp:Label>
            <asp:TextBox ID="txtFName" runat="server"
CssClass="textEntry"></asp:TextBox>
            <asp:RequiredFieldValidator ID="FName" runat="server"
CssClass="failureNotification" ErrorMessage="Δώστε Όνομα"
ControlToValidate="txtFName">*</asp:RequiredFieldValidator>
          </p>
          <p>
            <asp:Label ID="lblLName" runat="server">Επώνυμο:</asp:Label>
            <asp:TextBox ID="txtLName" runat="server"
CssClass="textEntry"></asp:TextBox>
            <asp:RequiredFieldValidator ID="LName" runat="server"
CssClass="failureNotification" ErrorMessage="Δώστε Επώνυμο"
ControlToValidate="txtLName">*</asp:RequiredFieldValidator>
          </p>
          <p>
            <asp:Label ID="lblEmail" runat="server">e-mail:</asp:Label>
            <asp:TextBox ID="txtemail" runat="server"
CssClass="textEntry"></asp:TextBox>
            <asp:RequiredFieldValidator ID="Email" runat="server"
CssClass="failureNotification" ErrorMessage="Δώστε e-mail"
ControlToValidate="txtemail">*</asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator Id="vldRejex" RunAt="server"
ControlToValidate="txtemail" ErrorMessage="Εισάγετε μια έγκυρη διεύθυνση e-mail"
ValidationExpression="^([a-zA-Z0-9_-\.\+])@[a-z0-9-]+\.(\[a-z0-9-]+\)*(\.[a-z]{2,3})$">
            </asp:RegularExpressionValidator>
          </p>
          <p>
            <asp:Label ID="lblPass" runat="server">Κωδικός:</asp:Label>
            <asp:TextBox ID="txtPass" runat="server" CssClass="passwordEntry"
TextMode="Password"></asp:TextBox>
            <asp:RequiredFieldValidator runat="server"
CssClass="failureNotification" ErrorMessage="Δώστε Κωδικό"
ControlToValidate="txtPass">*</asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator Id="passVal" RunAt="server"
ControlToValidate="txtPass" ErrorMessage="Ο κωδικός πρέπει να έχει 6-15 ψηφία"
ValidationExpression="^.{6,15}$">
            </asp:RegularExpressionValidator>
          </p>
          <p>
            <asp:Label ID="lblPassConf" runat="server">Επαληθευση
Κωδικού:</asp:Label>
            <asp:TextBox ID="ConfirmPassword" runat="server"
CssClass="passwordEntry" TextMode="Password"></asp:TextBox>
```

```

        <asp:RequiredFieldValidator ControlToValidate="ConfirmPassword"
        CssClass="failureNotification" Display="Dynamic" ErrorMessage="Επαληθεύστε τον κωδικό
        σας" ID="ConfirmPasswordRequired" runat="server">*</asp:RequiredFieldValidator>
        <asp:CompareValidator runat="server" ControlToCompare="txtPass"
        ControlToValidate="ConfirmPassword" CssClass="failureNotification" Display="Dynamic"
        ErrorMessage="Λάθος Κωδικός">*</asp:CompareValidator>
    </p>
    <p>
        <asp:Label ID="lblCity" runat="server">Πόλη:</asp:Label>
        <asp:TextBox ID="txtCity" runat="server"
        CssClass="textEntry"></asp:TextBox>
    </p>
    <p>
        <asp:Label ID="lblAddress" runat="server">Διεύθυνση:</asp:Label>
        <asp:TextBox ID="txtAddress" runat="server"
        CssClass="textEntry"></asp:TextBox>
    </p>
    <p>
        <asp:Label ID="lblPostCode" runat="server">Ταχυδρομικός
        Κώδικας:</asp:Label>
        <asp:TextBox ID="txtPostCode" runat="server"
        CssClass="textEntry"></asp:TextBox>
        <asp:RegularExpressionValidator Id="postVal" RunAt="server"
        ControlToValidate="txtPostCode" ErrorMessage="Εισάγετε έγκυρο TK"
        ValidationExpression="^[0-9]{5}$">
        </asp:RegularExpressionValidator>
    </p>
    <p>
        <asp:Label ID="lblPhone" runat="server">Σταθερό
        Τηλέφωνο:</asp:Label>
        <asp:TextBox ID="txtPhone" runat="server"
        CssClass="textEntry"></asp:TextBox>
        <asp:RequiredFieldValidator ID="Phone" runat="server"
        CssClass="failureNotification" ErrorMessage="Δώστε Τηλέφωνο"
        ControlToValidate="txtPhone">*</asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator Id="phoneVal" RunAt="server"
        ControlToValidate="txtPhone" ErrorMessage="Εισάγετε ένα έγκυρο αριθμό"
        ValidationExpression="^[0-9]{10}$">
        </asp:RegularExpressionValidator>
    </p>
    <p>
        <asp:Label ID="lblCellPhone" runat="server">Κινητό
        Τηλέφωνο:</asp:Label>
        <asp:TextBox ID="txtCellPhone" runat="server"
        CssClass="textEntry"></asp:TextBox>
        <asp:RegularExpressionValidator Id="mobileVal" RunAt="server"
        ControlToValidate="txtCellPhone" ErrorMessage="Εισάγετε ένα έγκυρο αριθμό"
        ValidationExpression="^[0-9]{10}$">
        </asp:RegularExpressionValidator>
    </p>
    <p>
        <asp:CheckBox ID="checkInvoice" runat="server" text="Τιμολόγιο"
        TextAlign="Left" AutoPostBack="True" />
    </p>
    <p>
        <asp:Label ID="lblCompany" runat="server" Visible="False">Επωνυμία
        Εταιρίας</asp:Label>
        <asp:TextBox ID="txtCompany" runat="server" CssClass="textEntry"
        Visible="False"></asp:TextBox>
    </p>
    <p>

```

```

                <asp:Label ID="lblProffession" runat="server"
Visible="False">Επάγγελμα:</asp:Label>
                <asp:TextBox ID="txtProffession" runat="server"
CssClass="textEntry"
                Visible="False"></asp:TextBox>
            </p>
            <p>
                <asp:Label ID="lblCompAddress" runat="server"
Visible="False">Διεύθυνση Εταιρίας:</asp:Label>
                <asp:TextBox ID="txtCompAddress" runat="server"
CssClass="textEntry"
                Visible="False"></asp:TextBox>
            </p>
            <p>
                <asp:Label ID="lblAFM" runat="server"
Visible="False">ΑΦΜ:</asp:Label>
                <asp:TextBox ID="txtAFM" runat="server" CssClass="textEntry"
Visible="False"></asp:TextBox>
                <asp:RegularExpressionValidator Id="afmVal" RunAt="server"
ControlToValidate="txtAFM" ErrorMessage="Εισάγετε έγκυρο ΑΦΜ"
ValidationExpression="^[0-9]{9}$">
                </asp:RegularExpressionValidator>
            </p>
            <p>
                <asp:Label ID="lblDOY" runat="server"
Visible="False">ΔΟΥ:</asp:Label>
                <asp:TextBox ID="txtDOY" runat="server" CssClass="textEntry"
Visible="False"></asp:TextBox>
            </p>
            <p>
                <asp:Label ID="lblCompPhone" runat="server"
Visible="False">Τηλέφωνο Εταιρίας:</asp:Label>
                <asp:TextBox ID="txtCompPhone" runat="server" CssClass="textEntry"
                Visible="False"></asp:TextBox>
                <asp:RegularExpressionValidator Id="compPhoneVal" RunAt="server"
ControlToValidate="txtCompPhone" ErrorMessage="Εισάγετε ένα έγκυρο αριθμό"
ValidationExpression="^[0-9]{10}$">
                </asp:RegularExpressionValidator>
            </p>
            <p class="btnSubmit">
                <asp:Button ID="btnSignUp" runat="server" Text="Υποβολή" /><br />
                <asp:Label ID="Label1" runat="server"></asp:Label>
            </p>
        </fieldset>
    </div>
    <asp:ValidationSummary ID="Summary" style="float:left;" runat="server"
CssClass="failureNotification" />
</asp:Panel>
</div>
</asp:Content>

```

Για την καταχώρηση των στοιχείων και τη δημιουργία ενός λογαριασμού χρησιμοποιείται ο παρακάτω κώδικας:

```

Protected Sub btnSignUp_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnSignUp.Click
    Dim con As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STRING"))
    Dim cmd, cmd2 As New SqlCommand

    cmd.CommandType = CommandType.StoredProcedure
    cmd2.CommandType = CommandType.StoredProcedure

```

```

If Session("CustomerID") > 0 Then
    cmd.CommandText = "customer_upd"
    Try
        con.Open()
        If checkInvoice.Checked AndAlso Exists(con, "SELECT * FROM
tblInvoice WHERE CustomerID = '" & Session("CustomerID") & "' ") = False Then
            cmd2.CommandText = "invoice_ins"
        ElseIf checkInvoice.Checked AndAlso Exists(con, "SELECT * FROM
tblInvoice WHERE CustomerID = '" & Session("CustomerID") & "' ") Then
            cmd2.CommandText = "invoice_upd"
        End If
    Catch ex As Exception
        MsgBox(ex.ToString)
    Finally
        con.Close()
    End Try
Else
    cmd.CommandText = "customer_ins"
    cmd2.CommandText = "invoice_ins"
End If
cmd.Connection = con
cmd2.Connection = con
With cmd
    If Session("CustomerID") > 0 Then
        .Parameters.AddWithValue("@CustomerID", Session("CustomerID"))
    End If
    .Parameters.AddWithValue("@FirstName", txtFName.Text.Trim)
    .Parameters.AddWithValue("@LastName", txtLName.Text.Trim)
    .Parameters.AddWithValue("@Email", txtemail.Text.Trim)
    .Parameters.AddWithValue("@Password", txtPass.Text.Trim)
    .Parameters.AddWithValue("@Address", txtAddress.Text.Trim)
    .Parameters.AddWithValue("@Town", txtCity.Text.Trim)
    .Parameters.AddWithValue("@PostalCode", txtPostCode.Text.Trim)
    .Parameters.AddWithValue("@HomePhone", txtPhone.Text.Trim)
    .Parameters.AddWithValue("@MobilePhone", txtCellPhone.Text.Trim)
End With
Try
    con.Open()
    If con IsNot Nothing AndAlso con.State = ConnectionState.Open Then
        If Exists(con, "SELECT * FROM tblCustomer WHERE Email = '" &
txtemail.Text.Trim & "' ") AndAlso Session("CustomerID") = Nothing Then
            Session("MESSAGE") = "Ο χρήστης έχει ήδη λογαριασμό"
            Response.Redirect("~/Message.aspx", False)
        Else
            Dim dr As SqlDataReader = cmd.ExecuteReader()
            While dr.Read()
                Session("CustomerID") = dr("CustomerID")
            End While
            dr.Close()

            If checkInvoice.Checked Then
                With cmd2
                    .Parameters.AddWithValue("@CustomerID",
Session("CustomerID"))
                    .Parameters.AddWithValue("@CompanyName",
txtCompany.Text.Trim)
                    .Parameters.AddWithValue("@Profession",
txtProffession.Text.Trim)
                    .Parameters.AddWithValue("@Address",
txtCompAddress.Text.Trim)
                    .Parameters.AddWithValue("@AFM", txtAFM.Text.Trim)
                    .Parameters.AddWithValue("@DOY", txtDOY.Text.Trim)
                End With
            End If
        End If
    End If
End Try

```

```

        .Parameters.AddWithValue("@CompanyPhone",
txtCompPhone.Text.Trim)
        End With
        cmd2.ExecuteNonQuery()
    End If
    Label1.Text = "Η εγγραφή ολοκληρώθηκε με επιτυχία"
End If
End If
Catch ex As Exception
    Session("MESSAGE") = ex.ToString
    Response.Redirect("~/Message.aspx", True)
Finally
    cmd.Dispose()
    cmd2.Dispose()
    con.Close()
End Try
End Sub

```

Επειδη όμως η σελίδα αυτή χρησιμοποιείται από ήδη συνδεδεμενους χρηστες για την αλλαγη στοιχειων, όταν φορτωνει τρεχει ο εξης κωδικας:

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    If Session("CustomerID") > 0 AndAlso Not IsPostBack Then
        Dim con As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STRING"))
        Dim cmd, cmd2 As New SqlCommand
        Dim dr As SqlDataReader

        txtemail.Enabled = False
        cmd.CommandType = CommandType.StoredProcedure
        cmd.CommandText = "customer_sel"
        cmd.Connection = con
        cmd.Parameters.AddWithValue("@CustomerID", Session("CustomerID"))
        Try
            con.Open()
            dr = cmd.ExecuteReader
            While dr.Read()
                txtFName.Text = dr("FirstName").ToString.Trim
                txtLName.Text = dr("LastName").ToString.Trim
                txtemail.Text = dr("Email").ToString.Trim
                txtAddress.Text = dr("Address").ToString.Trim
                txtCity.Text = dr("Town").ToString.Trim
                txtPostCode.Text = dr("PostalCode").ToString.Trim
                txtPhone.Text = dr("HomePhone").ToString.Trim
                txtCellPhone.Text = dr("MobilePhone").ToString.Trim
            End While
            dr.Close()
            If Exists(con, "SELECT * FROM tblInvoice WHERE CustomerID = '" &
Session("CustomerID") & "' ") Then
                cmd2.CommandType = CommandType.StoredProcedure
                cmd2.CommandText = "invoice_sel"
                cmd2.Connection = con
                cmd2.Parameters.AddWithValue("@CustomerID",
Session("CustomerID"))
                dr = cmd2.ExecuteReader
                While dr.Read()
                    txtCompany.Text = dr("CompanyName").ToString.Trim
                    txtProffession.Text = dr("Proffession").ToString.Trim
                    txtCompAddress.Text = dr("Address").ToString.Trim
                    txtAFM.Text = dr("AFM").ToString.Trim
                    txtDOY.Text = dr("DOY").ToString.Trim

```

```

        txtCompPhone.Text = dr("CompanyPhone").ToString.Trim
    End While
    dr.Close()
End If
Catch ex As Exception
    Session("MESSAGE") = ex.ToString
    Response.Redirect("~/Message.aspx", True)
Finally
    cmd.Dispose()
    cmd2.Dispose()
    con.Close()
End Try
End If

    btnSignUp.Text = "Υποβολή"
End Sub

```

7.9 AccInfo.aspx

```

<asp:Content ID="body" ContentPlaceHolderID="main" runat="server">
    <asp:Panel ID="Panel1" runat="server" Width="650px" ScrollBars="None">
        <fieldset class="login" style="float:left;">
            <legend><b>Προσωπικά Στοιχεία</b></legend>
            <asp:Table ID="tblPersonal" runat="server" Width="600px">
                <asp:TableRow>
                    <asp:TableCell>Όνομα:</asp:TableCell>
                    <asp:TableCell><asp:Label ID="lblFName"
runat="server"></asp:Label></asp:TableCell>
                </asp:TableRow>
                <asp:TableRow>
                    <asp:TableCell>Επώνυμο:</asp:TableCell>
                    <asp:TableCell><asp:Label ID="lblLName"
runat="server"></asp:Label></asp:TableCell>
                </asp:TableRow>
                <asp:TableRow>
                    <asp:TableCell>Πόλη:</asp:TableCell>
                    <asp:TableCell><asp:Label ID="lblTown"
runat="server"></asp:Label></asp:TableCell>
                </asp:TableRow>
                <asp:TableRow>
                    <asp:TableCell>Διεύθυνση:</asp:TableCell>
                    <asp:TableCell><asp:Label ID="lblAddress"
runat="server"></asp:Label></asp:TableCell>
                </asp:TableRow>
                <asp:TableRow>
                    <asp:TableCell>Ταχυδρομικός Κώδικας:</asp:TableCell>
                    <asp:TableCell><asp:Label ID="lblPostCode"
runat="server"></asp:Label></asp:TableCell>
                </asp:TableRow>
                <asp:TableRow>
                    <asp:TableCell>Σταθερό Τηλέφωνο:</asp:TableCell>
                    <asp:TableCell><asp:Label ID="lblPhone"
runat="server"></asp:Label></asp:TableCell>
                </asp:TableRow>
                <asp:TableRow>
                    <asp:TableCell>Κινητό Τηλέφωνο:</asp:TableCell>
                    <asp:TableCell><asp:Label ID="lblCellPhone"
runat="server"></asp:Label></asp:TableCell>
                </asp:TableRow>
            </asp:Table>
        </fieldset>
        <fieldset id="fInvoice" class="login" style="float:left;">

```

```

<legend><b>Στοιχεία Τιμολογίου</b></legend>
<asp:Table ID="tblInvoice" runat="server" Width="600px">
  <asp:TableRow>
    <asp:TableCell>Επωνυμία Εταιρίας:</asp:TableCell>
    <asp:TableCell><asp:Label ID="lblCompany"
runat="server"></asp:Label></asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
    <asp:TableCell>Επάγγελμα:</asp:TableCell>
    <asp:TableCell><asp:Label ID="lblProfession"
runat="server"></asp:Label></asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
    <asp:TableCell>Διεύθυνση Εταιρίας:</asp:TableCell>
    <asp:TableCell><asp:Label ID="lblCompAddress"
runat="server"></asp:Label></asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
    <asp:TableCell>ΑΦΜ:</asp:TableCell>
    <asp:TableCell><asp:Label ID="lblAFM"
runat="server"></asp:Label></asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
    <asp:TableCell>ΔΟΥ:</asp:TableCell>
    <asp:TableCell><asp:Label ID="lblDOY"
runat="server"></asp:Label></asp:TableCell>
  </asp:TableRow>
  <asp:TableRow>
    <asp:TableCell>Τηλέφωνο Εταιρίας:</asp:TableCell>
    <asp:TableCell><asp:Label ID="lblCompPhone"
runat="server"></asp:Label></asp:TableCell>
  </asp:TableRow>
</asp:Table>
</fieldset>
<asp:LinkButton ID="LinkButton1" runat="server">Αλλαγή
Στοιχείων</asp:LinkButton>
</asp:Panel>
<asp:GridView ID="GridView1" runat="server" AllowPaging="True"
AutoGenerateColumns="False" EmptyDataText="Δεν υπάρχουν δεδομένα για
αγορές"
  PageSize="6" CellPadding="4" ForeColor="#333333" GridLines="None"
  DataSourceID = "SqlDataSource1" Width = "725px" DataKeyNames="OrderID">
  <AlternatingRowStyle BackColor="White" />
  <Columns>
    <asp:BoundField HeaderText="Αριθμός Παραγγελίας" DataField="OrderID"
InsertVisible="False"
  ReadOnly="True" SortExpression="OrderID">
      <ItemStyle HorizontalAlign="Center" Width="100px" />
    </asp:BoundField>
    <asp:BoundField HeaderText="Ημερομηνία" DataField="Date"
SortExpression="Date">
      <ItemStyle HorizontalAlign="Center" Width="100px" />
    </asp:BoundField>
    <asp:BoundField HeaderText="Όνομα Προϊόντος" DataField="Name"
SortExpression="Name">
      <ItemStyle HorizontalAlign="Center" Width="425px" />
    </asp:BoundField>
    <asp:BoundField HeaderText="Τιμή" DataField="Price"
SortExpression="Price">
      <ItemStyle HorizontalAlign="Center" Width="100px" />
    </asp:BoundField>
  </Columns>

```

```

<FooterStyle BackColor="#990000" Font-Bold="True" ForeColor="White" />
<HeaderStyle BackColor="#CB4848" Font-Bold="True" ForeColor="White" />
<PagerStyle BackColor="#DC5050" ForeColor="#333333"
HorizontalAlign="Center"/>
<RowStyle BackColor="#DCDCDC" ForeColor="#333333" />
<SelectedRowStyle BackColor="#acacac" Font-Bold="True" ForeColor="Navy" />
<SortedAscendingCellStyle BackColor="#acacac" />
<SortedAscendingHeaderStyle BackColor="#4D0000" />
<SortedDescendingCellStyle BackColor="#acacac" />
<SortedDescendingHeaderStyle BackColor="#820000" />
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:myEshopDBConnectionString %>"
ProviderName="<%$ ConnectionStrings:myEshopDBConnectionString.ProviderName
%>"
SelectCommand="productorder_sel" SelectCommandType="StoredProcedure">
<SelectParameters>
<asp:SessionParameter Name="CustomerID" SessionField="CustomerID"
Type="Int32" />
</SelectParameters>
</asp:SqlDataSource>
</asp:Content>

```

Προγραμματιστικά για την προβολή των στοιχείων του πελάτη ακολουθούμε την ίδια λογική όπως και στην προβολή των στοιχείων ενός προϊόντος.

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load

```

```

    If Not Session("CustomerID") > 0 Then
        Session("MESSAGE") = "Πρέπει να συνδεθείτε πρώτα"
        Response.Redirect("~/Message.aspx", True)
    End If
    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim cmd2 As New SqlCommand
    Dim dr As SqlDataReader

    con.ConnectionString =
ConfigurationManager.AppSettings("CONNECTION_STRING")
    cmd.CommandType = CommandType.StoredProcedure
    cmd.CommandText = "customer_sel"
    cmd.Connection = con
    cmd.Parameters.AddWithValue("@CustomerID", Session("CustomerID"))
    Try
        con.Open()
        dr = cmd.ExecuteReader
        While dr.Read()
            lblFName.Text = dr("FirstName")
            lblLName.Text = dr("LastName")
            lblTown.Text = dr("Town")
            lblAddress.Text = dr("Address")
            lblPostCode.Text = dr("PostalCode")
            lblPhone.Text = dr("HomePhone")
            lblCellPhone.Text = dr("MobilePhone")
        End While
        dr.Close()
        If Exists(con, "SELECT * FROM tblInvoice WHERE CustomerID = '" &
Session("CustomerID") & "' ") Then
            cmd2.CommandType = CommandType.StoredProcedure
            cmd2.CommandText = "invoice_sel"
            cmd2.Connection = con

```



```

cmd2.Parameters.AddWithValue("@CustomerID", Session("CustomerID"))
dr = cmd2.ExecuteReader
While dr.Read()
    lblCompany.Text = dr("CompanyName")
    lblProfession.Text = dr("Profession")
    lblCompAddress.Text = dr("Address")
    lblAFM.Text = dr("AFM")
    lblDOY.Text = dr("DOY")
    lblCompPhone.Text = dr("CompanyPhone")
End While
End If
Catch ex As Exception
    MsgBox(ex.ToString)
Finally
    cmd.Dispose()
    cmd2.Dispose()
    con.Close()
End Try
End Sub

```

7.10 ViewCart.aspx

Η δημιουργία της σελίδας αυτής δίνεται από τον παρακάτω κώδικα. Η διαφορά με τα υπόλοιπα gridview που έχουμε δει μέχρι τώρα είναι το template field, όπου συνδυάζουμε σε ένα κελί ένα textbox και ένα κουμπί.

```

<asp:Content ID="Content2" ContentPlaceHolderID="main" Runat="Server">
<div style="width: 750px; margin-top:15px">
    <asp:Panel ID="Panel1" runat="server" Width="725px" ScrollBars="Vertical"
Height="460px" >
        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
EmptyDataText="Δεν υπάρχουν προϊόντα στο καλάθι"
CellPadding="5" ForeColor="#333333" GridLines="None"
DataKeyNames="ProductID" ShowFooter="True"
OnRowDataBound="GridView1_RowDataBound"
OnRowCommand="GridView1_RowCommand">
            <AlternatingRowStyle BackColor="White" />
            <Columns>
                <asp:BoundField DataField="Name" HeaderText="Όνομα" >
                    <ItemStyle Width="270px" />
                </asp:BoundField>
                <asp:TemplateField HeaderText="Ποσότητα">
                    <ItemTemplate>
                        <asp:TextBox runat="server" ID="txtQuantity" Columns="5"
Text='<%# Eval("Quantity") %>'></asp:TextBox>
                        <asp:LinkButton runat="server" ID="btnRemove" Text="Διαγραφή"
CommandName="Remove" CommandArgument='<%# Eval("ProductId") %>' style="font-
size:12px"></asp:LinkButton>
                    </ItemTemplate>
                </asp:TemplateField>
                <asp:BoundField DataField="UnitPrice" HeaderText="Τιμή Μονάδας"
ItemStyle-HorizontalAlign="Right" HeaderStyle-
HorizontalAlign="Right"
DataFormatString="{0:C}" >
            <HeaderStyle HorizontalAlign="Right"></HeaderStyle>
            <ItemStyle HorizontalAlign="Right"></ItemStyle>
                </asp:BoundField>
                <asp:BoundField DataField="TotalPrice" HeaderText="Σύνολο"

```

```

                ItemStyle-HorizontalAlign="Right" HeaderStyle-
HorizontalAlign="Right"
                DataFormatString="{0:C}" >
        <HeaderStyle HorizontalAlign="Right"></HeaderStyle>

        <ItemStyle HorizontalAlign="Right"></ItemStyle>
                </asp:BoundField>
        </Columns>
        <FooterStyle HorizontalAlign="Right" BackColor="#990000" Font-Bold="True"
ForeColor="White" />
        <HeaderStyle BackColor="#CB4848" Font-Bold="True" ForeColor="White" />
        <RowStyle BackColor="#DCDCDC" ForeColor="#333333" />
        <SelectedRowStyle BackColor="#acacac" Font-Bold="True" ForeColor="Navy" />
</asp:GridView>
<br />
        <asp:Button runat="server" ID="btnUpdateCart" Text="Ανανέωση Καλαθιού"
OnClick="btnUpdateCart_Click" />
        <asp:Button ID="Buy" runat="server" style="float:right" Text="Αγόρασε"
Width="163px" />
</asp:Panel>
</div>
</asp:Content>

```

Για να δούμε πως δουλεύει ο κώδικας στο καλάθι αγοράς πρέπει να κοιτάξουμε τις κλάσεις που φτιάξαμε για να κρατάμε τα αντικείμενα.

- CartItem.vb

```

Public Class CartItem
    Implements IEquatable(Of CartItem)

    Private _quantity As Integer
    Public Property Quantity() As Integer
        Get
            Return _quantity
        End Get
        Set(ByVal value As Integer)
            _quantity = value
        End Set
    End Property

    Private _productId As Integer
    Public Property ProductId() As Integer
        Get
            Return _productId
        End Get
        Set(ByVal value As Integer)
            _prod = Nothing
            _productId = value
        End Set
    End Property

    Private _prod As Product = Nothing
    Public ReadOnly Property Prod() As Product
        Get
            If _prod Is Nothing Then
                _prod = New Product(ProductId)
            End If
            Return _prod
        End Get
    End Property

    Public ReadOnly Property Name() As String
        Get

```

```

        Return Prod.Name
    End Get
End Property

Public ReadOnly Property UnitPrice As Decimal
    Get
        Return Prod.Price
    End Get
End Property

Public ReadOnly Property TotalPrice() As Decimal
    Get
        Return UnitPrice * Quantity
    End Get
End Property

Public Sub New(ByVal productId As Integer)
    Me.ProductId = productId
End Sub

Public Overloads Function Equals(ByVal item As CartItem) As Boolean Implements
System.IEquatable(Of CartItem).Equals
    Return item.ProductId = Me.ProductId
End Function
End Class

```

- Product.vb

```

Public Class Product

    Private _id As Integer
    Public Property Id() As Integer
        Get
            Return _id
        End Get
        Set(ByVal value As Integer)
            _id = value
        End Set
    End Property

    Private _price As Decimal
    Public Property Price() As Decimal
        Get
            Return _price
        End Get
        Set(ByVal value As Decimal)
            _price = value
        End Set
    End Property

    Private _name As String
    Public Property Name() As String
        Get
            Return _name
        End Get
        Set(ByVal value As String)
            _name = value
        End Set
    End Property

    Public Sub New(ByVal id As Integer)
        Me.Id = id
    End Sub

```

```

        Dim con As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STRING"))
        Dim cmd As New SqlCommand
        Dim dr As SqlDataReader
        cmd.CommandType = Data.CommandType.StoredProcedure
        cmd.CommandText = "product_sel"
        cmd.Parameters.AddWithValue("@Product", id)
        cmd.Connection = con

        Try
            con.Open()
            dr = cmd.ExecuteReader
            If dr.Read() Then
                Me.Price = dr("Price")
                Me.Name = dr("Name")
            End If
            dr.Close()
        Catch ex As Exception
            MsgBox(ex.ToString)
        Finally
            cmd.Dispose()
            con.Close()
        End Try

    End Sub
End Class

```

- ShoppingCart.vb

```

Public Class ShoppingCart

    Private _items As List(Of CartItem)
    Public Property Items() As List(Of CartItem)
        Get
            Return _items
        End Get
        Set(ByVal value As List(Of CartItem))
            _items = value
        End Set
    End Property

    Public Shared ReadOnly Instance As ShoppingCart

    Shared Sub New()
        If HttpContext.Current.Session("ASPNETShoppingCart") Is Nothing Then
            Instance = New ShoppingCart()
            Instance.Items = New List(Of CartItem)
            HttpContext.Current.Session("ASPNETShoppingCart") = Instance
        Else
            Instance = CType(HttpContext.Current.Session("ASPNETShoppingCart"),
ShoppingCart)
        End If
    End Sub

    Protected Sub New()

    End Sub

    Public Sub AddItem(ByVal productId As Integer)
        Dim newItem = New CartItem(productId)

        If Items.Contains(newItem) Then
            For Each item As CartItem In Items

```

```

        If item.Equals(newitem) Then
            item.Quantity += 1
            Return
        End If
    Next
Else
    newitem.Quantity = 1
    Items.Add(newitem)
End If
End Sub

Public Sub SetItemQuantity(ByVal productId As Integer, ByVal quantity As
Integer)
    If quantity = 0 Then
        removeItem(productId)
        Return
    End If

    Dim updatedItem = New CartItem(productId)
    For Each item As CartItem In Items
        If item.Equals(updatedItem) Then
            item.Quantity = quantity
            Return
        End If
    Next
End Sub

Public Sub RemoveItem(ByVal productId As Integer)
    Dim removedItem = New CartItem(productId)

    For Each item As CartItem In Items
        If item.Equals(removedItem) Then
            Items.Remove(item)
            Return
        End If
    Next
End Sub

Public Function GetSubTotal() As Decimal
    Dim subTotal As Decimal = 0
    For Each item As CartItem In Items
        subTotal += item.TotalPrice
    Next
    Return subTotal
End Function
End Class

```

Γνωρίζοντας τις κλάσεις αυτές και τις συναρτήσεις τους, στη σελίδα ViewCart.aspx.vb χρησιμοποιούμε κάποιο κώδικα για να τις εκμεταλλευτούμε.

- Διαγραφή ενός προϊόντος από το καλάθι.

```

Protected Sub GridView1_RowCommand(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.GridViewCommandEventArgs) Handles GridView1.RowCommand
    If e.CommandName = "Remove" Then
        Dim productId = Convert.ToInt32(e.CommandArgument)
        ShoppingCart.Instance.RemoveItem(productId)
    End If
    BindData()
End Sub

```

- Εμφάνιση συνολικής αξίας των προϊόντων που βρίσκονται στο καλάθι.

```
Protected Sub GridView1_RowDataBound(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.GridViewRowEventArgs) Handles GridView1.RowDataBound
    If e.Row.RowType = DataControlRowType.Footer Then
        e.Row.Cells(3).Text = "Αξία Προϊόντων: " &
ShoppingCart.Instance.GetSubTotal().ToString("C")
    End If
End Sub
```

- Ανανέωση καλαθιού για καταχώρηση ποσότητας ενός προϊόντος.

```
Protected Sub btnUpdateCart_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnUpdateCart.Click
    For Each row As GridViewRow In GridView1.Rows
        If row.RowType = DataControlRowType.DataRow Then
            Try
                Dim productId =
Convert.ToInt32(GridView1.DataKeys(row.RowIndex).Value)
                Dim quantity =
Integer.Parse(CType(row.Cells(1).FindControl("txtQuantity"), TextBox).Text)
                ShoppingCart.Instance.SetItemQuantity(productId, quantity)
            Catch ex As Exception
                Session("MESSAGE") = ex.ToString
                Response.Redirect("~/Message.aspx", True)
            End Try
        End If
    Next
    BindData()
End Sub
```

- Συνέχιση στην αγορά των προϊόντων που βρίσκονται στο καλάθι.

```
Protected Sub Buy_Click(sender As Object, e As System.EventArgs) Handles Buy.Click
    Response.Redirect("~/buy.aspx", True)
End Sub
```

7.11 Admin.aspx

Πάμε να δούμε λοιπόν πιο αναλυτικά πως είναι δομημένη η σελίδα. Καταρχήν το πρώτο πράγμα που βλέπουμε είναι ένα Menu με επιλογές για το τι θέλουμε να επεξεργαστούμε πχ. Category, SubCategory, Products κτλ απο αυτό το μενου ενεργοποιούνται τα views που είναι μέσα στο multiview και έτσι κάθε φορά βλέπουμε αυτό που έχουμε επιλέξει. Παρακάτω φαίνεται ο κώδικας που επιλέγει πιο view θα δείχνει στο πάτημα ενός στοιχείου του μενού.

```
Protected Sub Menu1_MenuItemClick(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.MenuEventArgs) Handles Menu1.MenuItemClick

    Dim epilogi As String
    epilogi = Menu1.SelectedValue.ToString

    Select Case epilogi
        Case "Category"
            MultiView1.SetActiveView(Category)
        Case "SubCategory"
            MultiView1.SetActiveView(SubCategory)
        Case "Manufacturer"
            MultiView1.SetActiveView(Manufacturer)
        Case "Product"
            MultiView1.SetActiveView(Product)
```

```

    Case "characteristic"
        MultiView1.SetActiveView(Characteristic)
    Case "Char_Val"
        MultiView1.SetActiveView(Char_Val)
    Case "Order"
        MultiView1.SetActiveView(Order)
End Select

```

```
End Sub
```

Βλέπουμε λοιπόν ότι έχουμε μια Case η οποία παίρνει την τιμή του κουμπιού που έχουμε επιλέξει και ανάλογα ενεργοποιεί το κατάλληλο view από το Multiview1. Όταν πατήσουμε το κουμπί Category θα δούμε καταρχήν μια radio list που εκεί επιλέγουμε τι θέλουμε να κάνουμε στο Category. Υπάρχουν 3 επιλογές α) να κάνουμε προσθήκη κατηγορίας, β) να κάνουμε επεξεργασία μιας κατηγορίας, γ) να κάνουμε διαγραφή μιας κατηγορίας. Ανάλογα τώρα τι θα επιλέξουμε εμφανίζονται και τα κατάλληλα πεδία. Παρακάτω φαίνεται ο κώδικας που ενεργοποιείται στο πάτημα ενός radio button από το list.

```

Protected Sub rd1Cat_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles rd1Cat.SelectedIndexChanged

```

```

    If rd1Cat.SelectedIndex = 0 Then
        lstCatAtCat.Visible = False
        txtNameAtCat.Text = ""
        txtDescAtCat.Text = ""
    ElseIf rd1Cat.SelectedIndex = 1 Then

        lstCatAtCat.Visible = True
        txtNameAtCat.Text = ""
        txtDescAtCat.Text = ""
    ElseIf rd1Cat.SelectedIndex = 2 Then
        lstCatAtCat.Visible = True
        txtNameAtCat.Text = ""
        txtDescAtCat.Text = ""
    End If

```

```
End Sub
```

Όπως βλέπουμε λοιπόν στο παράδειγμα μας για την κατηγορία αν πατήσουμε το πρώτο κουμπί που είναι αυτό της δημιουργίας κατηγορίας δεν θα μας εμφανίσει την λίστα με τις κατηγορίες καθώς δεν την χρειαζόμαστε και σβήνει και τα περιεχόμενα που υπάρχουν αν είχαμε κάνει κάτι πιο πριν από τα textbox txtnameAtCat και txtDescAtCat. Οπότε εμείς τώρα γράφοντας ένα όνομα και μια περιγραφή στα αντίστοιχα textbox το μόνο που θα χρειαστεί είναι να πατήσουμε το κουμπί από κάτω. Αν θέλουμε να τροποποιήσουμε μια κατηγορία και πατώντας το 2^ο κουμπί που λέει «τροποποίηση» μας ενεργοποιεί μια λίστα με όλες τις κατηγορίες που υπάρχουν στην βάση μας αυτή την στιγμή. Όταν επιλέξουμε ποια κατηγορία θέλουμε να τροποποιήσουμε, τότε τρέχει ο παρακάτω κώδικας ο οποίος μας γεμίζει τα πεδία txtNameAtCat και txtDescAtCat ώστε εμείς μετά να πάμε και να τα τροποποιήσουμε και πατώντας το κουμπί από κάτω να πάρει τα στοιχεία και να τα αλλάξει στην βάση μας. Παρακάτω φαίνεται ο κώδικας ο οποίος μας γεμίζει τα textboxes ανάλογα με την επιλογή της λίστας.

```

Protected Sub lstCatAtCat_SelectedIndexChanged(ByVal sender As Object, ByVal e As
System.EventArgs) Handles lstCatAtCat.SelectedIndexChanged
    txtNameAtCat.Text = lstCatAtCat.SelectedItem.ToString
    Dim conn As New SqlConnection()
    conn = sqlconnect()
    Dim command As New SqlCommand
    command.CommandType = CommandType.StoredProcedure
    command.CommandText = "tblCategory_sel"

```

```

command.Connection = conn
command.Parameters.AddWithValue("@id", lstCatAtCat.SelectedValue)
conn.Open()
Dim reader As SqlDataReader
reader = command.ExecuteReader()
While (reader.Read)
    txtDescAtCat.Text = reader("Description").ToString
End While
reader.Close()
conn.Close()
End Sub

```

```

tblCategory_sel
(
    @id int
)
as
select tblCategory.Description
from tblCategory
where CategoryID=@id

```

Καταρχήν να εξηγήσω πρώτα ότι έχω μια function αποθηκευμένη στο project για να μου δίνει αυτόματα τα στοιχεία σύνδεσης. Παρακάτω φαίνεται η sqlconnect().

```

Protected Function sqlconnect() As SqlConnection
    Dim conn As SqlConnection
    conn = New SqlConnection("Server = marios-PC\squlexpress;" & "initial
Catalog = myEshopDB;" & " Trusted_Connection=yes")
    Return conn
End Function

```

Η οποία μου γυρίζει το connection που θα χρειαστώ για να συνδεθώ στην βάση δεδομένων αφού λοιπόν κάνω το connection ενεργοποιώ μια stored Procedure που είναι αποθηκευμένη στην βάση μου και λέγεται tblCategory_sel η οποία παίρνει σαν παράμετρο ένα id το οποίο της το δίνει το επιλεγμένο αντικείμενο από την λίστα που έχουμε και στην συνέχεια επιστρέφουμε με ένα reader τα περιεχόμενα του Description γιατί το όνομα της κατηγορίας, όπως φαίνεται και στην πρώτη γραμμή του κώδικα, το παίρνουμε από το αντίστοιχο αντικείμενο της λίστας και τα βάζουμε στα αντίστοιχα textboxes . Αφού λοιπόν έχουμε κάνει ό,τι τροποποίηση θέλαμε στα δυο κουτιά πατάμε το κουμπί «τροποποίηση» και τα αλλάζει στην βάση μας. Το ίδιο συμβαίνει και στην περίπτωση που θέλουμε να κάνουμε διαγραφή μιας κατηγορίας αλλά δεν έχει νόημα τι θα αλλάξουμε στα κουτιά γιατί θα διαγραφεί η κατηγορία που έχουμε στην λίστα . παρακάτω φαίνεται όλος ο κώδικας που ενεργοποιείται όταν πατήσουμε το κουμπί «τροποποίηση».

```

Protected Sub btnCat_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnCat.Click
    Try
        If rd1Cat.SelectedIndex = 0 Then
            Dim conn As New SqlConnection()
            conn = sqlconnect()
            Dim command As New SqlCommand
            command.CommandType = CommandType.StoredProcedure
            command.CommandText = "tblCategory_ins"
            command.Connection = conn
            command.Parameters.AddWithValue("@name", txtNameAtCat.Text)
            command.Parameters.AddWithValue("@desc", txtDescAtCat.Text)
            conn.Open()
            command.ExecuteNonQuery()
            conn.Close()
            txtNameAtCat.Text = ""

```



```

        txtDescAtCat.Text = ""
        lstCatAtCat.DataBind()
    ElseIf rd1Cat.SelectedIndex = 1 Then
        Dim conn As New SqlConnection()
        conn = sqlconnect()
        Dim command As New SqlCommand
        command.CommandType = CommandType.StoredProcedure
        command.CommandText = "tblCategory_upd"
        command.Connection = conn
        command.Parameters.AddWithValue("@name", txtNameAtCat.Text)
        command.Parameters.AddWithValue("@desc", txtDescAtCat.Text)
        command.Parameters.AddWithValue("@id", lstCatAtCat.SelectedValue)
        conn.Open()
        command.ExecuteNonQuery()
        conn.Close()
        txtNameAtCat.Text = ""
        txtDescAtCat.Text = ""
        lstCatAtCat.DataBind()
    ElseIf rd1Cat.SelectedIndex = 2 Then
        Dim conn As New SqlConnection()
        conn = sqlconnect()
        Dim command As New SqlCommand
        command.CommandType = CommandType.StoredProcedure
        command.CommandText = "tblCategory_del"
        command.Connection = conn
        command.Parameters.AddWithValue("@id", lstCatAtCat.SelectedValue)
        conn.Open()
        command.ExecuteNonQuery()
        conn.Close()
        txtNameAtCat.Text = ""
        txtDescAtCat.Text = ""
        lstCatAtCat.DataBind()
    End If
Catch ex As Exception
    MsgBox("Error!")
End Try
End Sub

```

tblCategory_ins

```

ALTER procedure [dbo].[tblCategory_ins]
(
    @name nvarchar(max),
    @desc nvarchar(max)
)
as
insert into tblCategory ([Name],[Description])
values (@name,@desc)

```

tblCategory_upd

```

ALTER procedure [dbo].[tblCategory_upd]
(
    @name nvarchar(max),
    @desc nvarchar(max),
    @id int
)
as
update tblCategory
set Name =@name, Description = @desc
where CategoryID=@id

```

```
tblCategory_del
ALTER procedure [dbo].[tblCategory_del]
(
    @id int
)
as
delete from tblCategory
where CategoryID=@id
```

Βλέπουμε παραπάνω ότι κατά περίπτωση το τι έχουμε επιλέξει στο radio list μας κάνει και τις ανάλογες ενέργειες. Δηλαδή αν έχουμε πατήσει εισαγωγή το οποίο είναι η επιλογή 0 στο radio list τότε δημιουργεί μια σύνδεση με την βάση μέσω της sqlconnect και καλεί άλλη μια stored procedure από την βάση που λέγεται tblCategory_ins η οποία δέχεται σαν παραμέτρους ένα όνομα και μια περιγραφή και με αυτά δημιουργεί μια νέα εγγραφή στον πίνακα tblCategory. Στη συνέχεια απαλείφει τα περιεχόμενα από τα κουτιά για πιθανή είσοδο νέων στοιχείων για νέα κατηγορία. Αν έχουμε επιλέξει τροποποίηση είναι το 1 στο radio list και το οποίο δημιουργεί μια σύνδεση με την βάση και καλεί μια stored procedure την tblCategory_upd η οποία παίρνει σαν παραμέτρους το όνομα και την περιγραφή από τα κουτιά τα αντίστοιχα και το id της κατηγορίας που θέλουμε να αλλάξουμε από το επιλεγμένο αντικείμενο στην list και εν συνεχεία τροποποιεί την βάση με αυτά τα δεδομένα και στην συνέχεια σβήνει πάλι τα κουτιά όνομα και περιγραφή. Αν έχουμε επιλέξει διαγραφή, τότε καλεί την stored procedure tblCategory_del, η οποία παίρνει σαν παράμετρο το id της κατηγορίας και το διαγράφει από την βάση. Επίσης όλο αυτό είναι μέσα σε μια try-catch για πιθανά σφάλματα κατά την διαδικασία. Παρόμοια διαδικασία ακολουθείται και στις άλλες επιλογές του μενού με εξαίρεση αυτό της παραγγελίας το οποίο θα αναλύσω τώρα.

Όταν επιλέξουμε από το μενού Παραγγελία εκτός από τα 2 textbox που εμφανίζει και που εισάγουμε id Παραγγελίας και id προϊόντος και μετά πατώντας το κουτί μας διαγράφει από την παραγγελία το συγκεκριμένο προϊόν όπως ο κώδικας φαίνεται παρακάτω.

```
Protected Sub btnDelAtOrd_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnDelAtOrd.Click
    Try
        Dim conn As New SqlConnection()
        conn = sqlconnect()
        Dim command As New SqlCommand
        command.CommandType = CommandType.StoredProcedure
        command.CommandText = "tblProductOrder_del"
        command.Connection = conn
        command.Parameters.AddWithValue("@orderid", txtOrdAtOrd.Text)
        command.Parameters.AddWithValue("@productid", txtProdAtOrd.Text)
        conn.Open()
        command.ExecuteNonQuery()
        conn.Close()
    Catch ex As Exception
        MsgBox("Error!")
    End Try

End Sub
```

```
tblProductOrder_del
ALTER procedure [dbo].[tblProduct_del]
(
    @prodid int
)
as
DELETE FROM tblProduct
WHERE ProductID=@prodid
```

Όπως βλέπουμε από πάνω στην συνάρτηση αυτή αφού δημιουργήσει ένα connection με την βάση καλεί την stored procedure tblProductOrder_del και αφού πάρει σαν παραμέτρους τα id από τα αντίστοιχα textboxes, διαγράφει στην βάση την εγγραφή εκείνη στον πίνακα tblOrder. Όλο αυτό μέσα σε μια try catch για να μην μας δημιουργήσει κάποιο ξαφνικό πρόβλημα. Από κάτω λοιπόν από αυτά θα συναντήσουμε ένα list box με τις παραγγελίες που δεν έχουν αποσταλεί ακόμα και επιλέγοντας μια θα μας δείξει από κάτω ένα πίνακα με το προϊόν που έχει παραγγελθεί, την ποσότητα που ζητάει ο πελάτης και αυτή που διαθέτει το κατάστημα. Κάτω από αυτή την λίστα θα δούμε 2 κουμπιά ένα ενεργοποιημένο το «έλεγε» και ένα κλειστό από την αρχή το «στείλε». Στην αρχή μόνο το έλεγε είναι ανοιχτό και όταν το πατήσουμε ενεργοποιεί ένα κομμάτι κώδικα και ελέγχει αν υπάρχει διαθέσιμη ποσότητα για την παραγγελία και αν υπάρχει τότε ενεργοποιεί το στείλε για να τελειώσει η παραγγελία. Παρακάτω βλέπουμε το κομμάτι αυτό κώδικα

```
Protected Sub btnChk_Click(sender As Object, e As System.EventArgs) Handles
btnChk.Click

    Dim row1 As Integer = 0
    For Each row As GridViewRow In GridView1.Rows
        If row.RowType = DataControlRowType.DataRow Then
            Try
                Dim qEc = Convert.ToInt32(GridView1.Rows(row1).Cells(1).Text)
                Dim qCus = Convert.ToInt32(GridView1.Rows(row1).Cells(2).Text)
                If (qEc >= qCus) Then
                    btnSend.Enabled = True
                Else
                    MsgBox("πρέπει να προμηθευτείτε προϊόντα γιατί δεν υπάρχουν
αρκετά για να γίνει η παραγγελία")
                End If
            Catch ex As Exception
            End Try
        End If
        row1 += 1
    Next
End Sub
```

Όπως βλέπουμε πάει στο gridView1 και ελέγχει μια μια τις σειρές για τις ποσότητες που ζητά ο χρήστης και αυτές που υπάρχουν στην αποθήκη. Αν δεν υπάρχει αρκετή ποσότητα τότε μας βγάζει μήνυμα ότι πρέπει να αγοράσουμε και άλλο stock για να τελειώσει η παραγγελία αλλιώς αν υπάρχει stock τότε ενεργοποιείται το κουμπί «στείλε». Ο κώδικας που εκτελείται από το κουμπί στείλε φαίνεται παρακάτω.

```
Protected Sub btnSend_Click(sender As Object, e As System.EventArgs) Handles
btnSend.Click

    Dim conn As New SqlConnection()
    conn = sqlconnect()
    Dim comm As New SqlCommand("UPDATE tblOrder SET Sent=" & True & " where
OrderID=" & lstOrderAtOrd.Selected.Value, conn)
    comm.ExecuteNonQuery()
    comm.Dispose()
    conn.Close()
    Dim row1 As Integer = 0
    For Each row As GridViewRow In GridView1.Rows
        If row.RowType = DataControlRowType.DataRow Then
            Try
                Dim Prodid =
Convert.ToInt32(GridView1.Rows(row1).Cells(0).Text)
                Dim qEc = Convert.ToInt32(GridView1.Rows(row1).Cells(1).Text)
                Dim qCus = Convert.ToInt32(GridView1.Rows(row1).Cells(2).Text)
                qEc = qEc - qCus
            End Try
        End If
    Next
End Sub
```

```

        Dim conn1 As New SqlConnection()
        conn = sqlconnect()
        conn.Open()
        Dim comm1 As New SqlCommand("UPDATE tblProduct SET Quantity=" &
qEc & " where ProductId=" & Prodid, conn)
        comm.ExecuteNonQuery()
        comm.Dispose()
    Catch ex As Exception
        MsgBox(ex.ToString)
    Finally
        conn.Close()
    End Try
End If
row1 += 1
Next
btnSend.Enabled = False
End Sub

```

Παραπάνω φαίνεται ο κώδικας που εκτελείται από το κουμπί «στείλε», ο οποίος στην αρχή δημιουργεί το connection, αλλά τώρα δεν πηγαίνει σε stored procedures, αλλά η εντολή δίνεται αμέσως η οποία πάει στον πίνακα tblOrder, βρίσκει το orderid και αλλάζει το πεδίο Sent του πίνακα σε true σε πρώτη φάση. Στην συνέχεια πρέπει να ενημερώσει και την βάση ότι αφαιρέθηκαν τόσα προϊόντα. Όποτε παίρνει πάλι με επανάληψη κάθε σειρά του gridView1 και αφαιρεί την ποσότητα που ζητείται από αυτήν που έχουμε εμείς και το καινούργιο νούμερο το αποθηκεύει στον tblProduct στο Quantity του αντιστοίχου βέβαια Productid.

7.12 Buy.aspx

Στο buy.aspx όταν φορτώνει για πρώτη φορά η σελίδα (όταν πατηθεί το κουμπί από την προηγούμενη σελίδα) τότε εκτελείται ο παρακάτω κώδικας στην αρχή της σελίδας την ώρα που φορτώνει.

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
    tblinvatbuy.Visible = False
    Try
        If Session("CustomerID") > 0 Then
            Dim reader As SqlDataReader
            Dim conn As New
SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STRING"))
            Dim command As New SqlCommand
            Dim cmd As New SqlCommand
            command.CommandType = CommandType.StoredProcedure
            cmd.CommandType = CommandType.StoredProcedure
            command.CommandText = "customer_sel"
            command.Connection = conn
            command.Parameters.AddWithValue("@CustomerID",
Session("CustomerID"))
            conn.Open()
            reader = command.ExecuteReader()

            While (reader.Read)
                Try
                    Dim name As String
                    name = reader("FirstName").ToString & " " &
reader("LastName").ToString
                    txtNameAtbuy.Text = name
                    txtAddAtbuy.Text = reader("Address").ToString
                    txtEmailAtbuy.Text = reader("Email").ToString
                    txtMobileAtbuy.Text = reader("MobilePhone").ToString

```

```

        txtPhoneAtbuy.Text = reader("HomePhone").ToString
        txtTKAtbuy.Text = reader("PostalCode").ToString
        txtTownAtbuy.Text = reader("Town").ToString
    Catch ex As Exception
    End Try
End While

reader.Close()
If Exists(conn, "Select * from tblInvoice where CustomerID=" &
Session("CustomerID")) Then
    cmd.CommandText = "invoice_sel"
    cmd.Connection = conn
    cmd.Parameters.AddWithValue("@CustomerID",
Session("CustomerID"))
    reader = cmd.ExecuteReader

    While reader.Read
        txtCompAtbuy.Text = reader("CompanyName").ToString
        txtAfmAtbuy.Text = reader("AFM").ToString
        txtdayAtbuy.Text = reader("DOY").ToString
    End While
    reader.Close()
    cmd.Dispose()
    tblinvatbuy.Visible = True
    chkInvoice.Checked = True
Else
    chkInvoice.Visible = False
End If
conn.Close()
conn.Dispose()
Else
    Session("MESSAGE") = " Δεν εχετε κανει login"
    Response.Redirect("~/Message.aspx", True)
End If
GridView1.DataSource = ShoppingCart.Instance.Items
GridView1.DataBind()

Catch ex As Exception
    Response.Redirect("~/Account/Login.aspx", True)
End Try

End Sub

```

Customer_sel

```

ALTER procedure [dbo].[customer_sel]
(
    @CustomerID int
)
AS
SELECT * FROM tblCustomer WHERE (CustomerID=@CustomerID)

```

Invoice_sel

```

ALTER procedure [dbo].[invoice_sel]
(
    @CustomerID int
)
AS
SELECT * FROM tblInvoice WHERE CustomerID = @CustomerID

```

Στην αρχή ελέγχει αν ο χρήστης είναι ακόμα συνδεδεμένος ή απλά έρχεται πρώτη φορά στη σελίδα για να αγοράσει προϊόντα και δεν έχει λογαριασμό ή δεν έχει δημιουργήσει ακόμα. Αν συμβαίνει αυτό τότε ο χρήστης αυτόματα κατευθύνεται στην σελίδα εγγραφής και μετά μπορεί να ξαναγυρίσει. Στην συνέχεια αν έχει περάσει αυτόν τον έλεγχο, συνδέεται με την βάση και παίρνει τα στοιχεία του χρήστη και τα εκτυπώνει στην οθόνη στα κατάλληλα textboxes. Αν ο χρήστης έχει δηλώσει ότι θέλει τιμολόγιο και όχι απλή απόδειξη τότε υπάρχει ένα κουμπί που το ενεργοποιεί και αυτόματα εμφανίζεται και ένας δεύτερος πίνακας με τα στοιχεία του τιμολογίου που έχει αποθηκεύσει ο χρήστης.

```
Protected Sub chkInvoice_CheckedChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles chkInvoice.CheckedChanged
```

```
    If chkInvoice.Checked = True Then
        tblinvatbuy.Visible = True
    ElseIf chkInvoice.Checked = False Then
        tblinvatbuy.Visible = False
    End If
```

```
End Sub
```

Αν δεν έχει αποθηκεύσει τα στοιχεία τιμολογίου θα παραπεμφθεί στην σελίδα εγγραφής για να το δηλώσει εκεί. Εφόσον περάσει τους πίνακες αυτούς, βλέπει και ένα 3^ο πίνακα ο οποίος περιέχει τα προϊόντα που έχει επιλέξει ο χρήστης για να αγοράσει, για ένα τελευταίο έλεγχο. Και στο τέλος δυο κουμπιά ένα cancel και ένα συνέχισε. Αν πατηθεί το cancel θα τρέξει ο κώδικας:

```
Protected Sub btnCancelAtbuy_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnCancelAtbuy.Click
    Response.Redirect("~/Default.aspx", True)
End Sub
```

Και θα τον γυρίσει στην αρχική σελίδα. Αν πατήσει αγορά θα τρέξει ο κώδικας:

```
Protected Sub btnBuyAtBuy_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnBuyAtBuy.Click
    Dim conn As New SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STRING"))
    Dim command As New SqlCommand

    command.CommandType = CommandType.StoredProcedure
    command.CommandText = "tblOrder_ins"
    command.Connection = conn
    command.Parameters.AddWithValue("@customerID", Session("CustomerID"))
    command.Parameters.AddWithValue("@date", Date.UtcNow)
    command.Parameters.AddWithValue("@price", ShoppingCart.Instance.GetSubTotal)
    command.Parameters.AddWithValue("@sent", False)
    conn.Open()

    Dim reader As SqlDataReader
    reader = command.ExecuteReader()

    While (reader.Read)
        Dim con As New SqlConnection(ConfigurationManager.AppSettings("CONNECTION_STRING"))

        For Each row As GridViewRow In GridView1.Rows
            If row.RowType = DataControlRowType.DataRow Then
```

```

        Try
            Dim productId =
Convert.ToInt32(GridView1.DataKeys(row.RowIndex).Value)
            Dim quantity =
Integer.Parse(CType(row.Cells(1).FindControl("txtQuantity"), TextBox).Text)
            Dim comm As New SqlCommand
            comm.CommandType = CommandType.StoredProcedure
            comm.CommandText = "tblProductOrder_ins"
            comm.Connection = con
            comm.Parameters.AddWithValue("@ordid", reader("OrderID"))
            comm.Parameters.AddWithValue("@prodid", productId)
            comm.Parameters.AddWithValue("@quant", quantity)
            con.Open()
            comm.ExecuteNonQuery()
            con.Close()
            comm.Dispose()

        Catch ex As Exception
            Session("MESSAGE") = ex.ToString
            Response.Redirect("~/Message.aspx", True)
        End Try
    End If
Next
End While

    reader.Close()
    conn.Close()
    conn.Dispose()

    Session("Message") = " Η ΑΓΟΡΑ ΟΛΟΚΛΗΡΩΘΗΚΕ"
    Response.Redirect("~/Message.aspx")

End Sub

```

Ο παραπάνω κώδικας συνδέεται με την βάση, καλεί την stored procedure tblOrder_ins και αφού περάσει σαν παραμέτρους το Id του πελάτη, την ημερομηνία της παραγγελίας, την τελική τιμή όλης της παραγγελίας και τέλος ένα default bit ελέγχου για την παραγγελία, τότε κάνει μια νέα εγγραφή στον πίνακα tblOrder. Στη συνέχεια έχει περάσει ένα ένα τα προϊόντα στην λίστα και την ποσότητα και τα έχει περάσει στον πίνακα tblProductOrder για να είναι εύκολο να παρακολουθήσουμε στη συνέχεια ποια προϊόντα έχει αγοράσει ο χρήστης. Παρακάτω φαίνονται οι κώδικες των store procedure tblOrder_ins και tblProductOrder_ins

```

tblOrder_ins
(
    @customerid int,
    @date date,
    @price money,
    @sent bit
)
as
insert into tblOrder ([CustomerID] , [Date] , [Price], [Sent] )
values (@customerid , @date, @price, @sent)
select tblOrder.OrderID from tblOrder where CustomerID=@customerid and
Price=@price and Date=@date

tblProductOrd_ins
(
    @prodid int,
    @ordid int,
    @quant int
)

```

```
as
insert into tblProductOrder
([OrderID], [ProductID] ,[Quantity] )
values (@ordid , @prodid , @quant )
```