

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: « Εφαρμογή ηλεκτρονικού καταστήματος»

Εισηγητής: κ. Παχουλάκης Ιωάννης

Σπουδάστριες: Κωνσταντίνου Βασιλική Α.Μ.1784

Παπαδημητράκη Κατερίνα Α.Μ.1625

ΠΕΡΙΕΧΟΜΕΝΑ

Πρόλογος

1. Κεφάλαιο – Το λογισμικό πακέτο

1.1 Συνοπτική περιγραφή του συστήματος

1.2 Απαιτήσεις Συστήματος

1.3 Δυναμικές και Στατικές Ιστοσελίδες

1.3.1 Τι είναι ένα στατικό site

1.3.2 Τι είναι ένα δυναμικό site

1.3.3 Η χρήση ενός δυναμικού site μας παρέχει και άλλα πλεονεκτήματα

1.3.4 Επιπλέον χρήσεις ενός δυναμικού site

1.4 Το ASP αναλυτικά

1.5 Αντικείμενα ASP

1.6 Συστατικά ASP

1.7 ASP και Βάσεις Δεδομένων

1.8 Αντικείμενα ADO

1.8.2 Ιδιότητες αντικειμένου Recordset

1.8.1 Μέθοδοι αντικειμένου Recordset

1.9 Συγγραφή κώδικα με την SQL

1.10 CSS

2. Κεφάλαιο – Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης

2.1 Πλατφόρμα .NET

2.2 Χαρακτηριστικά του .NET Framework

2.3 Πλεονεκτήματα .NET

2.4 Μειονεκτήματα .NET (κυρίως στην ανάπτυξη παιχνιδιών)

2.5 Αρχιτεκτονική .NET

2.6 Visual Basic .NET

2.7 Web Forms και ASP .NET

3. Κεφάλαιο – Περιγραφή της Βάσης Δεδομένων

3.1 Βάσεις Δεδομένων και SQL Server 2005

3.2 Τι είναι το SQL Server Management Studio Express:

3.3 Πίνακες της Βάσης Δεδομένων

4. Κεφάλαιο – Ανάλυση Διαδικασιών σε ASP.NET

4.1 Επικύρωση για τις Web φόρμες - ASP.NET Validation Controls

4.1.1 RequiredFieldValidator

4.1.2 RegularExpressionValidator

4.1.3 ASP.NET ΑΡΧΙΤΕΚΤΟΝΙΚΗ

4.2 Ασφάλεια

4.2.1 Ασφάλεια του IIS (Internet Information Server)

4.2.2 Το αρχείο Web.config

4.3 Cookies

4.4 Σύνδεση με Βάση Δεδομένων

5. Κεφάλαιο – Η εφαρμογή μας

5.1 Οι σελίδες αναλυτικά

5.2 Οι σελίδες των ADMINISTRATOR

6. Επίλογος

7. Κεφάλαιο - Βιβλιογραφία – Πηγές

Ευχαριστίες

Θέλω να ευχαριστήσω όλους όσους συμμετείχαν και συνέβαλαν στην ολοκλήρωση της παρούσας πτυχιακής εργασίας. Ειδικότερα τον εισηγητή μου, κ. Παχουλάκη Ιωάννη για την βοήθεια και την καθοδήγησή του. Τα μέλη της εξεταστικής επιτροπής κ. Βαρδιάμπαση και κ. Ταμιωλάκη που μου έκαναν την τιμή να παρακολουθήσουν και να αξιολογήσουν την προσπάθειά μου. Τέλος την οικογένεια μου, που όλα αυτά τα χρόνια στάθηκε δίπλα μου.

Σας ευχαριστώ πολύ,

Κωνσταντίνου Βασιλική

Θέλω να ευχαριστήσω όλους όσους συμμετείχαν και συνέβαλαν στην ολοκλήρωση της παρούσας πτυχιακής εργασίας. Ειδικότερα τον εισηγητή μου, κ. Παχουλάκη Ιωάννη για την βοήθεια και την καθοδήγησή του. Τα μέλη της εξεταστικής επιτροπής κ. Βαρδιάμπαση και κ. Ταμιωλάκη που μου έκαναν την τιμή να παρακολουθήσουν και να αξιολογήσουν την προσπάθειά μου. Τέλος την οικογένεια μου, που όλα αυτά τα χρόνια στάθηκε δίπλα μου.

Σας ευχαριστώ πολύ,

Παπαδημητράκη Κατερίνα

Πρόλογος

Το θέμα της πτυχιακής εργασίας είναι η δημιουργία ενός ηλεκτρονικού καταστήματος όπου οι χρήστες μπορούν να πραγματοποιήσουν αγορές και οι διαχειριστές μπορούν να διαχειρίζονται την βάση του καταστήματος μόνοι τους.

Έχουμε επικεντρωθεί στην διευκόλυνση των χρηστών αλλά και περισσότερο των διαχειριστών.

1. Κεφάλαιο – Το λογισμικό πακέτο

1.1 Συνοπτική περιγραφή του συστήματος

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας εφαρμογής ηλεκτρονικού καταστήματος, στο οποίο ο πελάτης μπορεί να πραγματοποιήσει τις αγορές του μέσω του διαδικτύου καθώς επίσης να ανακαλύψει τις δυνατότητες που του παρέχει το κατάστημα και από την άλλη μεριά ο διαχειριστής μπορεί να διαχειρίζεται την βάση του καταστήματος μόνος του, χωρίς να έχει ανάγκη έναν ειδικό προγραμματιστή, καθώς το σύστημα διαχείρισης είναι φιλικό προς τον χρήστη.

Η εφαρμογή έχει αναπτυχθεί σε τεχνολογία ASP.NET, με το πρόγραμμα Visual Studio 2008, με βάση δεδομένων σε SQL Server 2005. Η βάση δεδομένων περιλαμβάνει όλα τα συστατικά που είναι απαραίτητα για την υλοποίηση ενός ηλεκτρονικού καταστήματος.

Ισχυρότεροι αλγόριθμοι κρυπτογράφησης είναι το επόμενο στάδιο εξέλιξης για μεγαλύτερη ασφάλεια των χρηστών και των συναλλαγών τους.

1.2 Απαιτήσεις Συστήματος

Οι απαιτούμενες τεχνικές προδιαγραφές της πλατφόρμας εγκατάστασης είναι οι ακόλουθες:

➤ Για την πλατφόρμα .NET Framework απαιτείται ένα από τα παρακάτω λειτουργικά συστήματα:

- Microsoft Windows 2000 Professional with Service Pack 2.0
- Microsoft Windows 2000 Server with Service Pack 2.0
- Microsoft Windows 2000 Advanced Server with Service Pack 2.0
- Microsoft Windows 2000 Datacenter Server with Service Pack 2.0
- Microsoft Windows XP Professional
- Microsoft Vista

➤ Για την πλατφόρμα του SQL Server 2005 ένα από τα παρακάτω:

- Microsoft Windows NT Server 4.0
- Microsoft Windows NT Server Enterprise Edition 4.0
- Microsoft Windows 2005 Server
- Microsoft Windows 2005 Advanced Server
- Microsoft Windows 2005 Data Center Server
- Microsoft Windows Server 2005
- Microsoft Windows XP Professional
- Microsoft Vista

➤ Κοινές προδιαγραφές:

- Μνήμη: 128MB τουλάχιστον (προτεινόμενη 512MB)
- Επεξεργαστής: Pentium 133 MHz ή γρηγορότερος

- Web server: Microsoft Internet Information Services (IIS) 5.0+

1.3 Δυναμικές και Στατικές Ιστοσελίδες

Πολλές φορές, πολλοί δυσκολεύονται να αποφασίσουν τι “χρειάζεται”, ώστε να δημιουργήσουν ένα χρήσιμο και χρηστικό website, το οποίο θα ανανεώνεται συνεχώς προκειμένου να παρέχει αξιόπιστη ενημέρωση προς τους επισκέπτες του. Ο πιο παρεξηγημένος παράγοντας απόφασης εστιάζεται στην επιλογή μεταξύ ενός δυναμικού site που περιέχει βάση δεδομένων για την συνεχή ενημέρωση και ενός στατικού site που περιέχει απλές ιστοσελίδες (στην ουσία μία συλλογή πληροφοριών). Καλό είναι λοιπόν πριν πάρουμε οποιαδήποτε απόφαση, να σκεφθούμε σοβαρά τις δυνατότητες αυτές.

1.3.1 Τι είναι ένα στατικό site

Στατικό site είναι μία συλλογή ιστοσελίδων αμετάβλητων κειμένων, χρησιμοποιώντας οποιοδήποτε πρόγραμμα επεξεργασίας. Οι σελίδες αυτές δύνανται να περιλαμβάνουν εκτενείς πληροφορίες (όπως στοιχεία επικοινωνίας, τιμές, προϊόντα / υπηρεσίες, ακόμα και γραφικά), αλλά δεν μπορούν να τροποποιούνται εύκολα, γρήγορα και με χαμηλό κόστος.

Για παράδειγμα, αναφέρουμε ένα μικρό site εμπορικής εταιρίας με κατάλογο προϊόντων. Αν η εταιρία ασχολείται αποκλειστικά με την πώληση μόνο 5 ειδών προϊόντων, ίσως πρέπει να δημιουργηθούν 5 διαφορετικές σελίδες αναγράφοντας σε κάθε σελίδα τις λεπτομέρειες του κάθε προϊόντος. Θα χρειαστεί επίσης να δημιουργηθούν και σελίδες όπως «Παραγγελίες» και «Η Εταιρία». Σε ένα στατικό site, όλες αυτές οι σελίδες δημιουργούνται μία φορά και «δημοσιεύονται» στον web server. Οι επισκέπτες του site μπορούν να έχουν πρόσβαση στις σελίδες του site και να ενημερωθούν για τα προϊόντα ενώ στη συνέχεια να κάνουν την παραγγελία τους. Η παραγγελία αυτή δεν γίνεται βέβαια on-line αλλά μέσω e-mail.

1.3.2 Τι είναι ένα δυναμικό site

Η λέξη «δυναμικό» είναι πολυδιάστατη. Μπορεί να αναφέρεται στον αυτόματο τρόπο με τον οποίο δημιουργούνται οι web σελίδες. Ο δυναμικός μηχανισμός, δημιουργεί τις σελίδες αυτές, διαβάζοντας πληροφορίες που προέρχονται από μία βάση δεδομένων, εισάγοντας τις πληροφορίες (data) σε ένα “Template” και στέλνοντας τη «σελίδα» πίσω στο web browser. Μπορεί επίσης να αφορά την εισαγωγή/ ανανέωση καθώς και διαγραφή πληροφοριών στις ήδη υπάρχουσες σελίδες χρησιμοποιώντας σελίδες διαχείρισης.

Σύμφωνα λοιπόν με το προηγούμενο παράδειγμα και ενώ η εταιρία έχει μεγαλώσει, διαθέτει πάνω από 100 είδη προϊόντων. Σε αυτή την περίπτωση, οι στατικές σελίδες μπορεί να αποδειχθούν εξαιρετικά δύσχρηστες. Κάθε φορά που χρειάζεται να γίνει μία αλλαγή στην τιμή, να προστεθεί κάποια λεπτομέρεια, να ανανεωθεί το εικαστικό κομμάτι του site, κλπ., πρέπει να παρέμβει κάποιος σε 100 σελίδες. Ένα επιπλέον δύσχρηστο σημείο στις στατικές σελίδες εντοπίζεται στο «index» των σελίδων. Εφ' όσον υπάρχουν 100 σελίδες (μία για κάθε προϊόν), με δυσκολία πλέον θα βρεθεί η συγκεκριμένη σελίδα στο back end του site ώστε να γίνει μία αλλαγή.

Απεναντίας, ένα δυναμικό site θα μας επέτρεπε να κάνουμε τα ακόλουθα:

- Να αλλάζουμε με ευκολία κάποια πληροφορία σε όλες τις σελίδες του site, κάνοντας μόνο μία αλλαγή στη βάση δεδομένων ή στο template και η οποία θα επηρεάζει όλες τις σελίδες.

Να αλλάζουμε εύκολα πληροφορίες σχετικά με τα προϊόντα ή τις υπηρεσίες που παρέχει. Για παράδειγμα, επιθυμούμε να κάνουμε αλλαγές στον τιμοκατάλογο που αφορά μόνο σε προϊόντα π.χ. της περυσινής χρονιάς. Θα προστεθεί ένα επιπλέον πεδίο στη βάση δεδομένων που θα απεικονίζει την αλλαγή αυτή.

- Να δίνουμε την δυνατότητα να επιλέγει ο επισκέπτης τον τρόπο με τον οποίο θέλει να βλέπει τα προϊόντα καθώς έχει τη δυνατότητα να προσωποποιεί τις σελίδες χρησιμοποιώντας διάφορα κριτήρια δικής του επιλογής που υπάρχουν στη βάση δεδομένων. Πχ. μέγεθος, χρώμα, είδος, τιμή κλπ.

1.3.3 Η χρήση ενός δυναμικού site μας παρέχει και άλλα πλεονεκτήματα:

- Αποθήκευση πληροφοριών σχετικά με προτιμήσεις των πελατών και προσωποποίηση σελίδων σύμφωνα με τις ανάγκες κάθε πελάτη.
- Δυνατότητα δημιουργίας στατιστικών αναφορών για λήψη αποφάσεων.

Παράδειγμα 1:

Η ταξινόμηση των πωλήσεων του τελευταίου τετράμηνου σύμφωνα με το είδος προκειμένου να δείτε αν τα συγκεκριμένα προϊόντα κάνουν επιτυχημένες πωλήσεις.

Παράδειγμα 2:

Σχετική αναφορά με τα προϊόντα που έχουν επιλέξει κατά καιρούς οι επισκέπτες του site, τα έχουν βάλει στο καλάθι τους αλλά δεν έχουν φτάσει ποτέ στην παραγγελία.

Παράδειγμα 3:

Ποια ήταν η τελευταία σελίδα που επισκέφτηκε ο χρήστης πριν την παραγγελία του, ώστε να προσδιοριστούν οι επιπρόσθετες πληροφορίες που χρειάζεται κάποιος προκειμένου να φτάσει σε αυτήν.

- Σύνδεση βάσης δεδομένων με την αποθήκη του φυσικού καταστήματος ώστε να προειδοποιείται ο πελάτης αν επιλέξει κάποιο προϊόν που δεν είναι διαθέσιμο.

1.3.4 Επιπλέον χρήσεις ενός δυναμικού site

- Χρήση τεχνολογιών όπως «message boards», όπου χρειάζεται να αποθηκεύονται πολλές πληροφορίες για μεγάλο διάστημα.
- Συντήρηση εκτενούς περιεχομένου που αλλάζει πολύ συχνά

Παράδειγμα:

Δελτία Τύπου, ισολογισμοί ή βάση δεδομένων για τεχνική υποστήριξη. Γρηγορότερες και αποτελεσματικότερες αναφορές στην αναζήτηση των περιεχομένων.

Είναι πολύ πιο γρήγορο να αναζητήσεις κάποια πληροφορία από μία βάση δεδομένων, δηλώνοντας key word στη μηχανή αναζήτησης και περιμένοντας μερικά δευτερόλεπτα τα αποτελέσματα, από το να ψάχνεις 100 ή 1000 web σελίδες μία προς μία.

- Αποθήκευση πληροφοριών στη βάση δεδομένων γενικού περιεχομένου

Παράδειγμα:

Η βάση δεδομένων μπορεί να σχεδιαστεί ώστε να καταχωρούνται οι βάρδιες των υπαλλήλων.

Οι δυνατότητες ενός δυναμικού site είναι πολλές και πρακτικά, τίποτα από όλα τα παραπάνω δεν είναι εφικτό χρησιμοποιώντας ένα στατικό web site. Σημαντικό είναι και το γεγονός ότι τα εργαλεία για την ανάπτυξη δυναμικών web site εξελίσσονται και βελτιώνονται συνεχώς, πράγμα που μειώνει σημαντικά το κόστος ανάπτυξης. Αν θέλετε λοιπόν να επιλέξετε μία λύση που να καλύπτει τις μακροπρόθεσμες ανάγκες σας με περισσότερα πλεονεκτήματα και πολύ μικρή διαφορά κόστους αυτή θα πρέπει να είναι δυναμική!

Όπως και οι στατικές σελίδες, είναι οικονομικές στην κατασκευή (μικρό κόστος ανά σελίδα), και επιτρέπουν εύκολη συντήρηση και παρακολούθηση (για μεγάλους δικτυακούς τόπους, με μεγάλο αριθμό προϊόντων και ειδών). Δεν είναι όμως τόσο εύκολο να δημιουργηθούν από έναν άπειρο χρήστη, καθώς απαιτούνται κάποιες βασικές γνώσεις προγραμματισμού και χρήσης εφαρμογών και βάσεων δεδομένων (SQL, PHP, ASP, Perl, κ.λπ.).

1.4 Το ASP αναλυτικά

Ένα από τα εργαλεία για την δημιουργία δυναμικών ιστοσελίδων είναι το ASP (Active Server Pages). Μέσα από ένα σύνολο συστατικών, αντικειμένων, και script γλωσσών που μας παρέχει, η δημιουργία δυναμικών δικτυακών τόπων γίνεται εύκολα, γρήγορα, με χαμηλό κόστος και κυρίως με αξιοπιστία και σιγουριά.

Η επίδραση του ASP στην ανάπτυξη ενός Web Site οφείλεται στην δυνατότητα του να παράγει υψηλό βαθμό αλληλεπίδρασης. Αυτό σημαίνει ότι τα πράγματα συμβαίνουν σε πραγματικό χρόνο(κατά την διάρκεια που ο χρήστης χρησιμοποιεί ενεργά το site), και με κάθε ανταλλαγή πληροφοριών ή δεδομένων μεταξύ του χρήστη και του Web Site η ανταπόκριση του ενός ή του άλλου προσαρμόζεται ανάλογα με τις τρέχουσες συνθήκες.

Δεν υπάρχει πραγματικό όριο στο ποσό της επεξεργασίας που μπορεί να εκτελέσει ένα site ανταποκρινόμενο στην είσοδο που λαμβάνει από τον χρήστη, όταν χρησιμοποιείται το ASP και άλλες συμβατές με αυτό τεχνολογίες. Τα καλύτερα σημερινά Web Sites αναγνωρίζουν τους επισκέπτες τους και τους χαιρετούν με το όνομα τους όταν αυτοί επιστρέφουν στο ίδιο site. Τα καλύτερα εμπορικά Web Sites εκτελούν συνήθως "εξατομικευμένο" marketing, βασιζόμενα σε προφίλ που διατηρούν για τους χρήστες και στις ήδη καταγεγραμμένες αγοραστικές τους συνήθειες.

Επί του παρόντος, δύο είναι οι βασικές πλατφόρμες που χρησιμοποιούνται για την συντριπτική πλειοψηφία των Web Sites - είτε κάποια παραλλαγή του UNIX, είτε τα Windows NT, 2000, 2003. Στην πλευρά του UNIX θα ακούτε πράγματα όπως το Common Gateway Interface(CGI) και η γλώσσα Perl(Practical Extraction Reporting Language) σε συζητήσεις για τα χαρακτηριστικά "διαλογικότητας" και την αλληλεπίδραση του χρήστη με το site. Στην πλευρά των Windows NT θα ακούτε για ASP και Javascript / VBscript. Και οι δύο τεχνολογίες εκτελούν σε ένα μεγάλο ποσοστό τις ίδιες λειτουργίες. Ωστόσο για μια ποικιλία λόγων το ASP δείχνει να είναι ευκολότερο, φθηνότερο και πιο αποτελεσματικό.

1.5 Αντικείμενα ASP

Με την αυστηρή έννοια του όρου, το ASP δεν είναι μία γλώσσα προγραμματισμού. Μοιάζει περισσότερο με μία συγκολλητική ουσία η οποία κρατάει ενωμένα πολλαπλά επιμέρους συστατικά - τα scripts, τα αντικείμενα και την αλληλεπίδραση με τον Web Server. Από τεχνικής απόψεως το ASP διαθέτει αντικείμενα τα οποία καλούνται από την VBscript ή την Javascript για την εκτέλεση συγκεκριμένων, εξαιρετικά χρησιμων λειτουργιών, όπως η συλλογή των δεδομένων που στέλνουν οι χρήστες, η ανταπόκριση στην

είσοδο των χρηστών, η διαχείριση εφαρμογών και συνοδών εργασίας και ο χειρισμός του Server.

Η χρήση μιας εφαρμογής Browser είναι σχετικά εύκολη υπόθεση: Τοποθετούμε απλώς τον δείκτη του ποντικιού πάνω σε μια σύνδεση και κάνουμε κλικ για να μεταβούμε στον προορισμό της, ή συμπληρώνουμε τα πεδία μιας φόρμας και κάνουμε κλικ στο κουμπί «υποβολής» (Submit) για να στείλουμε τα δεδομένα στον server. Ωστόσο κάτω από αυτή την ευκολία και απλότητα κρύβεται μια σχετικά πολύπλοκη συναλλαγή - μια μορφή αμφίδρομης επικοινωνίας μεταξύ του Web server και του client συστήματος. Υπεύθυνα για αυτή την διαδικασία είναι τα αντικείμενα που καλούνται από το ASP.

Μερικά από τα αντικείμενα αυτά τα οποία είναι και ευρέως γνωστά στον κόσμο του ASP παρουσιάζονται και αναλύονται παρακάτω :

Το αντικείμενο **i) Request (αίτηση)** και **ii) Response (απάντηση)**

Τα αντικείμενα αυτά χρησιμοποιούνται για την συλλογή των εισερχόμενων δεδομένων από τον χρήστη και την εμφάνιση μιας απάντησης σε αυτόν, στέλνοντας επεξεργασμένο κείμενο, HTML κώδικα και άλλα δεδομένα.

Αναλυτικά:

i) Όταν ο χρήστης κάνει κλικ σε μια σύνδεση ή υποβάλει μια φόρμα, το αντικείμενο **Request** συλλέγει όλα τα δεδομένα της αίτησης, συμπεριλαμβανομένων των μεταβλητών HTTP, των cookies, των ερωτημάτων (query strings), των πιστοποιητικών ασφαλείας (security certificates) και συγκεκριμένων ιδιοτήτων (properties) του ίδιου του αντικειμένου Request, οι οποίες μπορούν να χρησιμοποιηθούν για την διαχείριση της επικοινωνίας. Όλα αυτά τα δεδομένα γίνονται άμεσα διαθέσιμα στο ASP script που χρησιμοποιείται.

ii) Το αντικείμενο **Response** είναι το εργαλείο που έχουμε στην διάθεση μας για να ανταποκρινόμαστε στις αιτήσεις των χρηστών, αλλά εξυπηρετεί επίσης και άλλες χρήσιμες λειτουργίες. Για παράδειγμα μπορούμε να κατευθύνουμε επιλεκτικά τους χρήστες σε άλλα URL ή σελίδες, χρησιμοποιώντας την μέθοδο Redirect αυτού του αντικειμένου. Υποστηρίζει επίσης την λειτουργία προσωρινής ενδιάμεσης αποθήκευσης (buffering), η οποία αποθηκεύει το περιεχόμενο μιας επιστρεφόμενης σελίδας μέχρι να ολοκληρωθεί όλη η επεξεργασία. Αυτό σημαίνει ότι μπορούμε να στέλνουμε στον χρήστη τα τμήματα που είναι ήδη έτοιμα, κατά την διάρκεια που περιμένει να ολοκληρωθεί η υπόλοιπη επεξεργασία.

Επειδή στο αντικείμενο αυτό βασίζονται πολλές εργασίες περιλαμβάνει πολλές χρήσιμες μεθόδους. Αναφορικά οι πιο γνωστές και πιο συχνά χρησιμοποιούμενες είναι: Write(“αλφαριθμητικό ή άλλη τιμή”), Redirect(“url”) , End() , BinaryWrite (“ασφαλής_διάταξη”).

Επίσης :

Το **iii) Application (εφαρμογή)** και **iv) Session (σύνοδος)** τα οποία δίνουν στον δημιουργό την δυνατότητα να καθορίσει την λειτουργικότητα του Web site σαν να ήταν μια εφαρμογή της οποίας οι μεταβλητές είναι ορατές από το σύνολο των ιστοσελίδων που χρησιμοποιούνται, ή σαν μία σύνοδο της οποίας οι μεταβλητές είναι ορατές μόνο από τις σελίδες που χρησιμοποιεί ένας μεμονωμένος χρήστης.

Αναλυτικά:

iii) Το αντικείμενο **Application** αποτελεί έναν γενικό χώρο αποθήκευσης για μεταβλητές (όλες τύπου Variant, με διαφορετικούς δευτερεύοντες τύπους για κάθε συγκεκριμένο τύπο δεδομένων) οι οποίες μπορούν να «στεγάζουν» κείμενο, αριθμητικές τιμές, ημερομηνίες, διατάξεις (arrays) και δείκτες προς αντικείμενα COM.

iv) Όμοια με το αντικείμενο Application, το αντικείμενο Session διαθέτει αρκετά συμβάντα τα οποία μας επιτρέπουν να εκκινήσουμε ενέργειες κατά την έναρξη ή τον τερματισμό ενός Session. Το Session ξεκινάει όταν ζητείται η πρώτη σελίδα από έναν χρήστη και τερματίζεται όταν **α)** το Session λήξει (Session Timeout) **β)** καλείται η μέθοδος Session.Abandon (Τερματίζουμε το Session του χρήστη στην σελίδα που έχουμε καθορίσει). Μπορούμε επίσης να ορίσουμε το χρονικό διάστημα για την λήξη του Session χρησιμοποιώντας το παράθυρο διαλόγου Application Configuration. Εναλλακτικά μπορούμε να ορίσουμε αυτό το διάστημα για μεμονωμένες σελίδες χρησιμοποιώντας όπως είπαμε πιο πάνω την ιδιότητα Session.Timeout. Σε κάθε περίπτωση, αυτή η δυνατότητα μας παρέχει έναν βολικό τρόπο για να διαχειριζόμαστε τα Session των χρηστών.

1.6 Συστατικά ASP

Τα συστατικά (components) είναι πολύτιμα επειδή μας παρέχουν έτοιμη λειτουργικότητα η οποία θα απαιτούσε πολύ χρόνο για να την υλοποιήσουμε μόνοι μας. Τα συστατικά που είναι ενσωματωμένα στο ASP είναι ουσιαστικά δωρεάν, ενώ το κόστος των περισσότερων συστατικών που προέρχονται από τρίτους κατασκευαστές είναι σχετικά χαμηλό. Οι μηχανές διερμηνεύσης script (scripting engines) μας παρέχουν πρόσβαση στο δικό τους σύνολο αντικειμένων. Τα πιο αξιοσημείωτα από αυτά είναι τα αντικείμενα Dictionary (λεξικό), FilesystemObject(σύστημα αρχείων), Drive(μονάδα δίσκου), Folder(φάκελος, κατάλογος) και File(αρχείο). Συλλογικά όλα αυτά αποκαλούνται αντικείμενα της βιβλιοθήκης Χρόνου Εκτέλεσης Script (Scripting Runtime Library).Μια ειδική κατηγορία συστατικών ASP είναι τα ActiveX Data Objects τα οποία και θα αναλύσουμε παρακάτω.

1.7 ASP και Βάσεις Δεδομένων

Ένα δυνατό χαρακτηριστικό στο ASP είναι το ότι υπάρχει συμβατότητα με ένα μεγάλο αριθμό βάσεων δεδομένων. Ανάλογα με το μέγεθος της εφαρμογής που έχουμε αναπτύξει και με τις δυνατότητες που θέλουμε να παρέχει μπορούμε να επιλέξουμε για το αν θα έχουμε μια μεγάλη και ισχυρή βάση δεδομένων ή μια μικρή για να μας παρέχει απλά έναν τρόπο αποθήκευσης.

Τέτοιες βάσεις είναι:

MS Access, μια σχετικά μικρή αλλά με πολλές δυνατότητες βάση που μπορεί να καλύψει με αξιοπιστία τις ανάγκες μιας εφαρμογής - ιστοσελίδας την οποία διαχειρίζονται ταυτόχρονα το πολύ δέκα χρήστες.

SQL Server, όπου απευθύνεται σε οποιοδήποτε τύπο και μέγεθος δικτυακού τόπου με ασφάλεια και αξιοπιστία για την αποθήκευση και διαχείριση των δεδομένων που έρχονται από τον Web Server.

Από την στιγμή που το ASP συνεργάζεται χωρίς προβλήματα συμβατότητας μαζί τους, πρέπει να επικεντρωθούμε σε θέματα ασφαλείας διαχείρισης, αποθήκευσης και μεταφοράς δεδομένων από τον Web Server στην Βάση. Για να κάνουμε την επιλογή που μας ικανοποιεί, προσπαθούμε να έχουμε όσο το δυνατόν λιγότερα ή καθόλου προβλήματα προσβασιμότητας, υπερχειλίσις δεδομένων και δύσκολης διαχείρισης.

1.8 Αντικείμενα ADO

ADO(ActiveX Data Object) είναι ο όρος που χρησιμοποιείται για να χαρακτηρίσει ένα από τα πολυτιμότερα συστατικά που μπορεί να καλεί το ASP. Τα αντικείμενα ADO επιτρέπουν την απευθείας πρόσβαση σε Βάσεις Δεδομένων και άλλες πηγές δεδομένων, και η λειτουργικότητα αυτή εξυπηρετεί πολλούς σκοπούς στα δυναμικά, καθοδηγούμενα από εφαρμογές Web Sites.

Αναλυτικά:

Τα αντικείμενα ADO είναι μια ειδική κατηγορία συστατικών ASP τα οποία είναι ειδικά σχεδιασμένα για την υποστήριξη βάσεων δεδομένων. Λόγω της σχεδίασής τους, τα αντικείμενα ADO συνεργάζονται καλά με το ASP, επειδή υπάρχουν πάρα πολλές εφαρμογές οι οποίες απαιτούν υποστήριξη βάσεων δεδομένων. Στο παρασκήνιο μπορούμε να χρησιμοποιήσουμε σχεδόν οποιαδήποτε βάση δεδομένων θέλουμε, ανεξάρτητα από το εάν αυτή είναι η Microsoft Access, το SQL Server, ή οποιαδήποτε άλλη βάση δεδομένων συμβατή με το πρότυπο ODBC. Υπάρχουν επίσης ορισμένες εξειδικευμένες βάσεις δεδομένων και τεχνολογίες προσπέλασης online βάσεων δεδομένων, εάν μας ενδιαφέρει να επενδύσουμε σε κάτι τέτοιο, ή εάν έχουμε ειδικές ανάγκες.

Οποιοδήποτε αντικείμενο αποθηκεύει δεδομένα μπορεί να θεωρηθεί σαν ένας παροχέας δεδομένων, ενώ οποιαδήποτε εφαρμογή τα χρησιμοποιεί μπορεί να θεωρηθεί σαν ένας καταναλωτής δεδομένων. Η Microsoft έχει δημιουργήσει μια αρχιτεκτονική δύο επιπέδων για την ανάκτηση δεδομένων, η οποία αποτελείται από τις τεχνολογίες ADO και OLE DB. Η OLE DB είναι ο μηχανισμός για την απευθείας επικοινωνία με την αποθήκη δεδομένων, οποιαδήποτε και εάν είναι αυτή, ενώ το ADO επικοινωνεί με την OLE DB. Αυτό σημαίνει ότι μπορούμε να ασχοληθούμε με τον προγραμματισμό σε επίπεδο ADO και η OLE DB θα φροντίσει για όλες τις τεχνικές λεπτομέρειες, στο παρασκήνιο.

- Ένα από τα βασικότερα αντικείμενα του ADO είναι το **Recordset (Σύνολο Εγγραφών)**, το οποίο μας παρέχει έλεγχο πάνω σε ένα σύνολο εγγραφών και πεδίων προερχόμενων από μια πηγή δεδομένων, έτσι ώστε να μπορέσουμε να εκτελέσουμε όλες τις κοινές λειτουργίες διαχείρισης που έχουμε στην διάθεσή μας όταν δουλεύουμε με τους πίνακες μιας βάσης δεδομένων(π.χ. εύρεση, επεξεργασία, προσθήκη και διαγραφή εγγραφών, κ.λ.π.).

Αναλυτικά:

Κάθε στήλη αντιπροσωπεύει ένα πεδίο και κάθε γραμμή περιέχει τις τιμές που σχετίζονται με μια συγκεκριμένη οντότητα - μια εγγραφή. Σε ένα αντικείμενο **Recordset** τα πεδία αντιπροσωπεύονται από την συλλογή fields. Μπορούμε να δουλέψουμε με τις τιμές ενός πεδίου μιας συγκεκριμένης εγγραφής χρησιμοποιώντας την ακόλουθη σύνταξη:

```
Variable = Recordset.Fields("όνομα_πεδίου")  
           ή  
           Recordset("όνομα_πεδίου")
```

Χρησιμοποιούμε αντικείμενα Connection και/ή Command για να υλοποιήσουμε την σύνδεση με μια πηγή δεδομένων και πιθανώς να ορίσουμε κάποιες παραμέτρους για την ανάκτηση των εγγραφών, με ένα συγκεκριμένο τύπο cursor. Χρησιμοποιούμε εντολές της SQL ή κλήσεις προς ολόκληρους πίνακες για να καθορίσουμε τις εγγραφές που θέλουμε να ανακτήσουμε. Αφού δημιουργήσουμε ένα αντικείμενο Recordset, χρησιμοποιούμε το αντικείμενο αυτό καθώς και τις μεθόδους που μας προσφέρει για να χειριστούμε τις εγγραφές που ανακτάμε με όποιο τρόπο θέλουμε.

Όπως και οποιοδήποτε άλλο αντικείμενο, το αντικείμενο Recordset υποστηρίζει μια ποικιλία μεθόδων και ιδιοτήτων με την διαφορά ότι μας παρέχουν πολλές από τις ίδιες δυνατότητες που έχουμε στην διάθεση μας όταν ανοίγουμε μια βάση δεδομένων.

Ενδεικτικά αναφέρουμε:

1.8.1 Μέθοδοι αντικειμένου Recordset

- Open (ανοίγει ένα αντικείμενο Recordset και δέχεται προαιρετικά ορίσματα)
- Close (Κλείνει ένα αντικείμενο Recordset)
- MoveFirst (Μετακινεί τον δείκτη στην αρχική εγγραφή)
- MoveNext (Μετακινεί τον δείκτη στην επόμενη εγγραφή)

Οι ιδιότητες που είναι διαθέσιμες για ένα αντικείμενο Recordset μπορούν να μας πουν αρκετά πράγματα για την θέση την οποία βρισκόμαστε μέσα στο αντικείμενο Recordset, το μέγεθος και την σύνθεσή του και διάφορα άλλα πράγματα , όπως η προέλευση και η κατάσταση ενός αντικειμένου Recordset. Συνοπτικά Αναφέρουμε:

1.8.2 Ιδιότητες αντικειμένου Recordset

- BOF (ο δείκτης βρίσκεται στην αρχή του αντικειμένου)
- EOF (ο δείκτης βρίσκεται στο τέλος του αντικειμένου)
- RecordCount (μας επιστρέφει το πλήθος των εγγραφών)

1.9 Συγγραφή κώδικα με την SQL

Η συγγραφή **SQL** κώδικα διαφέρει από την συγγραφή κώδικα με μια γλώσσα script σε ένα σημαντικό σημείο: ο SQL κώδικας επιδεικνύει συνήθως μια σαφή προδιάθεση για τα ερωτήματα. Για παράδειγμα, μια πάρα πολύ κοινή εντολή της SQL είναι η SELECT. Η εντολή SELECT μας επιτρέπει να ανακτήσουμε δεδομένα από μια βάση δεδομένων, αφού καθορίσουμε τις στήλες των οποίων τις τιμές θέλουμε να ανακτήσουμε, τον πίνακα ή τους πίνακες από τους οποίους θα ανακτηθούν τα δεδομένα, καθώς και οποιαδήποτε κριτήρια θα χρησιμοποιηθούν για τον περιορισμό των εγγραφών που θα ανακτηθούν.

Στην SQL μπορούμε να χρησιμοποιήσουμε την εντολή SELECT για την ανάκτηση δεδομένων, την INSERT για την προσθήκη εγγραφών, την DELETE για την διαγραφή εγγραφών και την UPDATE για την τροποποίηση ή επεξεργασία των δεδομένων των εγγραφών. Η εντολή της SQL που απαιτείται για την εκτέλεση οποιασδήποτε απ' τις παραπάνω ενέργειες ξεκινά με ένα από τα προαναφερθέντα ρήματα και ακολουθείται από επιπλέον στοιχεία τα οποία καθορίζουν τους πίνακες και τις εγγραφές που πρόκειται να επηρεασθούν από την εντολή.

Η SQL περιλαμβάνει επίσης μια ποικιλία εντολών οι οποίες επηρεάζουν την δομή της βάσης δεδομένων ή των πινάκων της. Σε αυτές περιλαμβάνονται οι CREATE TABLE, ALTER TABLE, και DROP TABLE, καθώς και παρόμοιες εντολές οι οποίες επηρεάζουν τις προβολές, τα ευρήματα, τα σχήματα και τα domain. Υπάρχουν επίσης ορισμένες εντολές για τον έλεγχο των δικαιωμάτων πρόσβασης και των συναλλαγών, καθώς και

για τον ορισμό άλλων λειτουργιών μέσω του προγραμματισμού όπως π.χ. η χρήση των cursor και η διεξαγωγή δυναμικών λειτουργιών.

1.10 CSS

α) Τι Είναι τα CSS

- Ο όρος CSS σημαίνει *Cascading Style Sheets*, δηλ. *Διαδοχικά Φύλλα Στυλ*.
- Τα στυλ ορίζουν τον τρόπο εμφάνισης των στοιχείων της HTML.
- Τα Στυλ (Styles) αποθηκεύονται σε Φύλλα Στυλ (Style Sheets).
- Τα Στυλ προστέθηκαν στην HTML 4.0 για να λύσουν κάποια προβλήματα.
- Τα Εξωτερικά Φύλλα Στυλ (External Style Sheets) μπορούν να μας γλυτώσουν από πολλή δουλειά.
- Πολλαπλοί ορισμοί στυλ μπορούν να συμπεριληφθούν σ' έναν.

β) Τα Στυλ Λύνουν ένα Κοινό Πρόβλημα

Τα tags της HTML σχεδιάστηκαν αρχικά για να ορίσουν το περιεχόμενο ενός εγγράφου. Η δουλειά τους ήταν να έλεγαν "Αυτή είναι μια επικεφαλίδα", "Αυτή είναι μια παράγραφος", "Αυτός είναι ένας πίνακας", χρησιμοποιώντας tags όπως <h1>, <p>, <table> κοκ. Η διάταξη (layout) του εγγράφου ήταν υπόθεση του φυλλομετρητή, χωρίς τη χρήση tags μορφοποίησης (formatting tags).

Καθώς οι δύο κύριοι φυλλομετρητές, ο Netscape και ο Internet Explorer, συνέχισαν να προσθέτουν νέα HTML tags και χαρακτηριστικά (attributes), όπως το tag και το attribute color, στις αρχικές προδιαγραφές της HTML, γινόταν ολοένα και δυσκολότερη η δημιουργία Web sites όπου το περιεχόμενο των εγγράφων HTML να μπορεί να ξεχωρίζει καθαρά από τη διάταξη παρουσίασης του εγγράφου.

Για να μπορέσει να λύσει αυτό το πρόβλημα, το *World Wide Web Consortium (W3C)* – το μη κερδοσκοπικό consortium ορισμού στάνταρτς που είναι υπεύθυνο για την στάνταρτατοποίηση της HTML – δημιούργησε τα ΣΤΥΛ (STYLES) σαν προσθήκη στην HTML 4.0.

Και ο Netscape 4.0 και ο Internet Explorer 4.0 υποστηρίζουν τα Διαδοχικά Φύλλα Στυλ (Cascading Style Sheets).

γ) Τα Φύλλα Στυλ μάς Γλυτώνουν από Πολλή Δουλειά

Τα στυλ στην HTML 4.0 ορίζουν το πώς εμφανίζονται τα HTML στοιχεία, όπως ακριβώς το tag font και το attribute color στην HTML 3.2. Τα στυλ αποθηκεύονται συνήθως σε αρχεία που είναι εξωτερικά (external) στα HTML έγγραφά μας.

Τα εξωτερικά φύλλα στυλ (external style sheets) μάς δίνουν τη δυνατότητα να αλλάξουμε την εμφάνιση και τη διάταξη όλων των σελίδων στο δικό μας Web site, με απλή επεξεργασία ενός μόνου CSS εγγράφου. Αν έχουμε ποτέ προσπαθήσει να αλλάξουμε τη γραμματοσειρά (font) ή το χρώμα (color) όλων των επικεφαλίδων (headings) σ' όλες τις ιστοσελίδες μας, θα καταλάβουμε γιατί τα CSS μπορούν να μας γλυτώσουν από πολλή δουλειά.

Τα CSS αποτελούν μια μεγάλη επιτυχία στον σχεδιασμό του Web (Web design) επειδή δίνουν τη δυνατότητα στους developers να ελέγξουν το στυλ και τη διάταξη πολλών ιστοσελίδων μονομιάς. Σαν Web developer μπορούμε να ορίσουμε ένα στυλ για κάθε HTML στοιχείο και να το εφαρμόσουμε σ' όσες ιστοσελίδες θέλουμε. Για να κάνουμε μια καθολική (global) αλλαγή, απλά αλλάζουμε το στυλ μία φορά και όλα τα στοιχεία του Web ενημερώνονται αυτόματα.

δ) Πολλαπλά Στυλ Καταλήγουν σ' ένα

Τα Φύλλα Στυλ επιτρέπουν τον ορισμό των πληροφοριών στυλ με πολλούς τρόπους. Τα στυλ μπορούν να ορισθούν μέσα σ' ένα μόνο HTML στοιχείο, μέσα στο στοιχείο <head> μιας HTML σελίδας ή σ' ένα εξωτερικό αρχείο CSS. Πολλά εξωτερικά Φύλλα Στυλ μπορούν να χρησιμοποιούνται μέσα απ' ένα μόνο HTML έγγραφο.

Ποιο στυλ θα χρησιμοποιηθεί όταν υπάρχουν περισσότερα από ένα καθορισμένα στυλ για ένα HTML στοιχείο ;

Σε γενικές γραμμές μπορούμε να πούμε ότι όλα τα στυλ θα καταλήξουν (cascade) σ' ένα νέο εικονικό (virtual) Φύλλο Στυλ σύμφωνα με τους παρακάτω κανόνες, όπου ο μεγαλύτερος αριθμός έχει και την υψηλότερη προτεραιότητα :

1. Προεπιλογή του φυλλομετρητή.
2. Εξωτερικό Φύλλο Στυλ (External Style Sheet).
3. Εσωτερικό Φύλλο Στυλ (Internal Style Sheet), μέσα στο τμήμα header του εγγράφου.
4. Inline Style, μέσα στο HTML στοιχείο.

ε) Η Σύνταξη των CSS

Η σύνταξη των CSS αποτελείται από τρία μέρη : έναν επιλογέα (selector), μια ιδιότητα (property) και μια τιμή (value) :

επιλογέας {ιδιότητα: τιμή}

selector {property: value}

Ο *επιλογέας* είναι συνήθως το στοιχείο/tag που θέλουμε να ορίσουμε, η *ιδιότητα* είναι το χαρακτηριστικό που θέλουμε να αλλάξουμε και η κάθε ιδιότητα μπορεί να πάρει μια *τιμή*. Η ιδιότητα και η τιμή ξεχωρίζουν από τον χαρακτήρα : και περικλείονται από τους χαρακτήρες { }, ως εξής :

body {color: black}

Αν η τιμή αποτελείται από πολλές λέξεις, πρέπει να τοποθετήσουμε εισαγωγικά :

p {font-family: "sans serif"}

Αν θέλουμε να ορίσουμε περισσότερες από μία ιδιότητες, πρέπει να ξεχωρίσουμε την κάθε ιδιότητα με τον χαρακτήρα ;. Το παρακάτω παράδειγμα δείχνει πώς μπορούμε να ορίσουμε μια κεντραρισμένη παράγραφο με χρώμα κειμένου κόκκινο :

p {text-align: center; color: red}

Για να κάνουμε τους ορισμούς των στυλ πιο ευανάγνωστους, μπορούμε να γράψουμε από μία ιδιότητα σε κάθε γραμμή, ως εξής :

p

{

text-align: center;

color: black;

font-family: arial

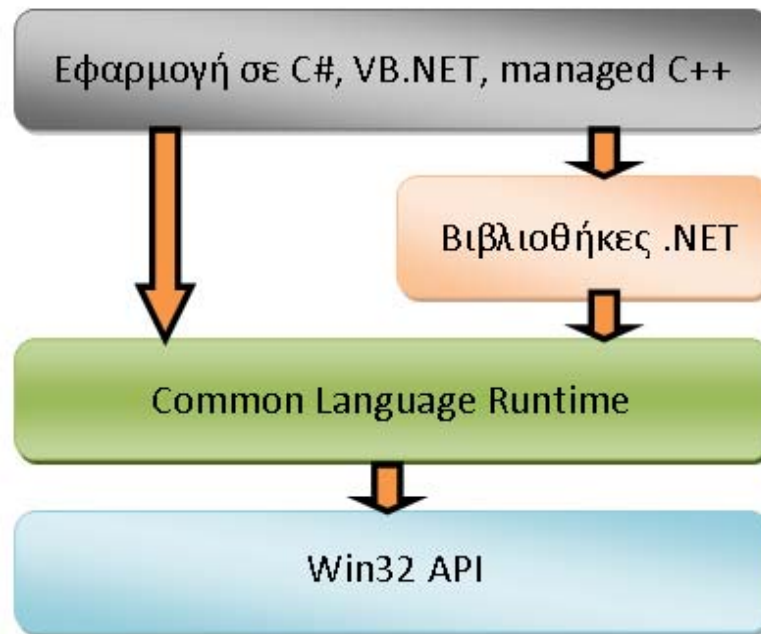
}

2. Κεφάλαιο – Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης

2.1 Πλατφόρμα .NET

Η πλατφόρμα ASP. NET που χρησιμοποιήθηκε είναι μια πλατφόρμα ανάπτυξης λογισμικού για Windows, Internet και World Wide Web. Ανακοινώθηκε από την MS τον Ιούνιο του 2001 και πιστεύεται ότι θα είναι το πρότυπο για την δημιουργία εφαρμογών για την επόμενη δεκαετία. Ειδικότερα η πλατφόρμα .NET στηρίζεται στο .NET Framework. Αυτό το πλαίσιο εργασίας εκτελεί και διαχειρίζεται εφαρμογές, περιέχει την βιβλιοθήκη κλάσεων, ενισχύει την ασφάλεια και παρέχει πολλές άλλες προγραμματιστικές δυνατότητες. Έχει πολλές δυνατότητες στις εφαρμογές δικτύου και μεγάλη ευκολία στη σύνδεση με διάφορες βάσεις δεδομένων. Εσωκλείει όλες τις τελευταίες τεχνολογίες στο δυναμικό προγραμματισμό. Οι εφαρμογές της είναι δυναμικές σελίδες που εκτελούνται στον Web Server και το αποτέλεσμα τους (σε html) εμφανίζεται στον Browser του χρήστη.

Η Καρδιά του .NET αποτελείται από το λεγόμενο Common Language Runtime (CLR). Η οντότητα αυτή είναι το managed περιβάλλον μέσα στο οποίο τρέχουν οι εφαρμογές .NET. Κατά μια έννοια κρύβει το δύσχρηστο Win32 API που χρησιμοποιείται συχνά για προγραμματισμό εφαρμογών Windows και παρουσιάζει στο χρήστη ένα απλούστερο και περισσότερο εύχρηστο (βλέπε παρακάτω σχήμα).



2.2 Χαρακτηριστικά του .NET Framework

1. Ανεξαρτησία από μία συγκεκριμένη γλώσσα

Είναι δυνατή η δημιουργία εφαρμογών σε οποιαδήποτε γλώσσα συμβατή με .NET

Κυριότερες γλώσσες είναι η Visual Basic.NET, C#, J#, Visual C++.NET

Υποστηρίζονται ακόμα 20 γλώσσες

2. Ανεξαρτησία από μία συγκεκριμένη πλατφόρμα

Η μεταγλώττιση των προγραμμάτων γίνεται σε δύο βήματα. Πρώτα γίνεται το MSIL (Microsoft Intermediate Language) και έπειτα σε κώδικα μηχανής για μία συγκεκριμένη πλατφόρμα.

Εάν το .NET Framework υπάρχει και είναι εγκατεστημένο για κάποιο λειτουργικό σύστημα τότε αυτό μπορεί να εκτελέσει οποιοδήποτε πρόγραμμα .NET.

3. Framework Class Library (FCL)

Είναι η βασική βιβλιοθήκη κλάσεων που χρησιμοποιούν τα προγράμματα .NET

Είναι κοινή για όλες τις γλώσσες και περιέχει μία τεράστια ποικιλία συστατικών επαναληπτικής χρήσης , επιταχύνοντας τον χρόνο δημιουργίας εφαρμογών.

4. Άλλα σημαντικά χαρακτηριστικά

ASP.NET

Δημιουργία εφαρμογών για PDA, Mobiles

Web Services

Διαλειτουργικότητα γλωσσών

CLR (Common Language Runtime)

Ο κώδικας που εκτελείται στο CLR έρχεται υπό μορφή assemblies, με επέκταση .dll ή .exe (δεν έχουν σχέση με τα κλασσικά .dll αρχεία των Windows). Το .NET, όπως και η Java χαρακτηρίζονται από την λεγόμενη Just in Time μεταγλώττιση. Ο κώδικας, στη γλώσσα προγραμματισμού που χρησιμοποιεί ο χρήστης, μεταγλωττίζεται αρχικά σε MSIL η οποία αποθηκεύεται σε ένα εκτελέσιμο .exe αρχείο ή σε μια βιβλιοθήκη .dll. Όταν ο χρήστης τρέξει το πρόγραμμα που ανέπτυξε, το CLR διαβάζει το MSIL κώδικα του αρχείου και Just In Time (JIT) το μεταγλωττίζει σε κώδικα Windows (native) έτοιμο προς εκτέλεση, και στην συνέχεια τον εκτελεί. Αυτό το επιπλέον βήμα πριν την εκτέλεση του κώδικα διαφοροποιεί μια managed εφαρμογή σε .NET από μια unmanaged σε C++ για παράδειγμα. Το πρόγραμμα σε C++ είναι ήδη μεταγλωττισμένο σε native Windows κώδικα και τρέχει απευθείας.

Ένα ακόμα χαρακτηριστικό του CLR είναι η αυτοματοποιημένη διαχείριση μνήμης. Σε κλασσικές γλώσσες προγραμματισμού (unmanaged) όπως η C++, όταν ο χρήστης δεσμεύσει μια ποσότητα μνήμης για να αποθηκεύσει ένα αντικείμενο πρέπει να είναι πολύ προσεκτικός στο να την αποδεσμεύσει, να την επιστρέψει στο σύστημα δηλαδή, όταν δεν την χρειάζεται. Αν το αγνοήσει αυτό συστηματικά, τότε θα δημιουργηθεί το επονομαζόμενο memory leak, δηλαδή η διαθέσιμη μνήμη του συστήματος θα ελαττώνεται διαρκώς και σε κάποιο σημείο τα Windows στερέψουν από ελεύθερη μνήμη.

Αντιθέτως το CLR προσφέρει ένα μηχανισμό Garbage Collection. Ο χρήστης μπορεί να ζητήσει όση μνήμη χρειάζεται από το σύστημα και να μην ασχοληθεί με την απελευθέρωση της. Ο Garbage Collector υλοποιεί μηχανισμούς που του επιτρέπουν να «καταλάβει» πότε μια δεσμευμένη ποσότητα μνήμης δεν χρησιμοποιείται πλέον και αυτόματα την απελευθερώνει για μετέπειτα χρήση.

2.3 Πλεονεκτήματα .NET

Το .NET έχει πολλά πλεονεκτήματα για την ανάπτυξη εφαρμογών:

- Είναι εγγενώς αντικειμενοστραφές πλατφόρμα.
- Είναι ανεξάρτητο από γλώσσα προγραμματισμού. Σε μια εφαρμογή ένας προγραμματιστής μπορεί να γράφει κώδικα σε C#, άλλος σε VB.NET και άλλος σε managed C++ και τα τμήματα που αναπτύσσει ο καθένας να συνεργάζονται μεταξύ τους χωρίς προβλήματα.
- Η χρήση βιβλιοθηκών (assemblies) κάνει πολύ εύκολη την επαναχρησιμοποίηση κώδικα.
- Παρέχει πολύ εύκολη εγκατάσταση. Αρκεί να αντιγράψουμε το κατάλογο της εφαρμογής σε ένα άλλο υπολογιστή και αυτή θα τρέξει άμεσα. Δεν υπάρχει installation, δεν πειράζει το registry.
- Παρέχει πληθώρα έτοιμων λειτουργιών που κάνουν την ανάπτυξη κώδικα πολύ εύκολη.
- Αυτοματοποιημένη διαχείριση μνήμης, ο χρήστης δεν χρειάζεται να ασχοληθεί με αποδέσμευση μνήμης.

2.4 Μειονεκτήματα .NET (κυρίως στην ανάπτυξη παιχνιδιών)

Το .NET έχει 2 μειονεκτήματα που αφορούν ειδικά την ανάπτυξη βιντεοπαιχνιδιών και όχι την γενική ανάπτυξη εφαρμογών:

- Αυτοματοποιημένη διαχείριση μνήμης, ο χρήστης δεν χρειάζεται να ασχοληθεί με αποδέσμευση μνήμης.
- Το CLR εισάγει μια (μικρή ίσως) καθυστέρηση στην εκτέλεση της εφαρμογής.

Το πρώτο, αν και είναι πολύ χρήσιμο χαρακτηριστικό για γενικό προγραμματισμό, σε ένα βιντεοπαιχνίδι μπορεί να δημιουργήσει πρόβλημα. Η απελευθέρωση μνήμης που δεν χρησιμοποιείται πλέον είναι μια ακριβή εργασία (από πλευράς χρόνου), η οποία επιπλέον είναι μη-ντετερμινιστική, μπορεί να συμβεί δηλαδή οποτεδήποτε. Αυτό μπορεί να έχει σαν αποτέλεσμα ένα παιχνίδι που τρέχει σε ένα σταθερό ρυθμό 60 καρέ το δευτερόλεπτο, να δει μια δραματική πτώση στο ρυθμό ανανέωσης για 1 δευτερόλεπτο, πράγμα ανεπίτρεπτο στην ανάπτυξη βιντεοπαιχνιδιών.

Το δεύτερο αφορά τη Just in Time μεταγλώττιση που υποστηρίζει το CLR. Από τη μία εισάγει μια καθυστέρηση στην εκκίνηση του παιχνιδιού μιας και πρέπει να μεταγλωττιστεί ο κώδικας από την άλλη ο μεταγλωττιστής ο ίδιος δεν είναι βέλτιστος με την έννοια ότι δεν παράγει το καλύτερο δυνατό native κώδικα για Windows.

Και τα δυο προβλήματα αυτά μπορούν να αντιμετωπιστούν αν τα λάβουμε υπόψη κατά την ανάπτυξη της εφαρμογής.

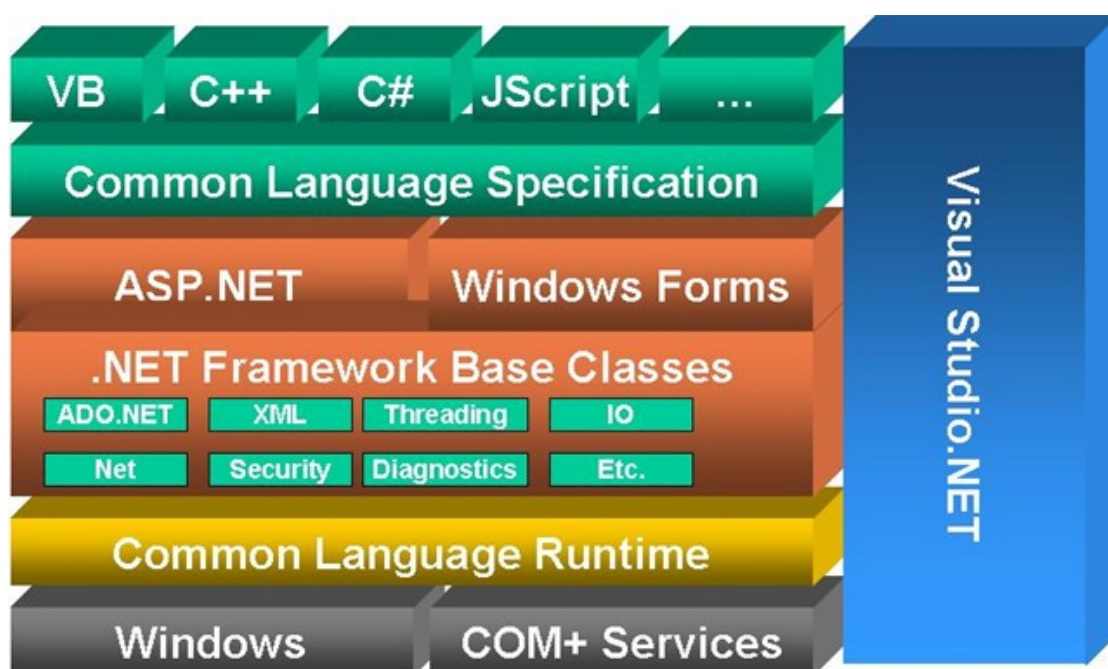
2.5 Αρχιτεκτονική .NET

Μέχρι σήμερα, η μεταγλώττιση μιας εφαρμογής από τον αντίστοιχο compiler δημιουργούσε εκτελέσιμα αρχεία (EXE ή DLL). Αυτά περιελάμβαναν απευθείας κώδικα μηχανής για την πλατφόρμα x86 και φυσικά μπορούσαν να τρέξουν μόνο σε αυτού του είδους τα συστήματα.

Στο περιβάλλον .NET η μεταγλώττιση από έναν compiler, όπως είναι οι Visual Basic, C++ και C# στο Visual Studio, δημιουργεί έναν ενδιάμεσο κώδικα, που η Microsoft ονομάζει MSIL (Microsoft Intermediate Language). Αυτός είναι ανεξάρτητος από τον επεξεργαστή και βρίσκεται σε υψηλότερο επίπεδο από τον κώδικα μηχανής. Περιλαμβάνει μάλιστα και metadata (στοιχεία με πληροφορίες για τη χρήση τους) που χρησιμοποιεί το CLR για να φορτώσει συγκεκριμένες λειτουργίες, να τοποθετήσει αντικείμενα στην μνήμη, να ελέγξει την ασφάλεια και να πραγματοποιήσει μια σειρά από πρόσθετες λειτουργίες. Δεν πρόκειται ασφαλώς για χαζό κώδικα μηχανής, καθώς το MSIL έχει σκοπό να κάνει πιο εύκολη τη μετάβαση από τα 32μπιτ Windows στη 64μπιτ πλατφόρμα των Windows 2000, όταν αυτή κυκλοφορήσει. Φυσικά, οι εντολές του MSIL δεν μπορούν να εκτελεστούν ως έχουν από τους σημερινούς επεξεργαστές, καθώς πρέπει πρώτα να μεταφραστούν σε γλώσσα μηχανής, στοιχείο που αναλαμβάνει το CLR.

Τα πλεονεκτήματα αυτού του τρόπου εκτέλεσης των εφαρμογών είναι πολλά. Πρώτα από όλα προσφέρεται ένας ενιαίος τρόπος για τη χρήση των υπηρεσιών του λειτουργικού συστήματος, σε αντίθεση με την σημερινή κατάσταση, κατά την οποία μερικές υπηρεσίες βρίσκονται σε DLL και άλλες σε αρχεία τύπου COM. Αυτό σημαίνει επίσης ότι παύει η δυσχρηστία με τα DLL, δηλαδή το φαινόμενο κατά το οποίο η εγκατάσταση μιας εφαρμογής αντικαθιστά κάποιο DLL μιας άλλης εφαρμογής με απρόβλεπτα, όπως πάντα, αποτελέσματα. Στην πλατφόρμα .NET υπάρχει πια ο διαχωρισμός των στοιχείων της κάθε εφαρμογής, έτσι ώστε όταν αυτή εκτελείται, να φορτώνονται συγχρόνως και τα στοιχεία με τα οποία αναπτύχθηκε και ελέγχθηκε. Η δυνατότητα της εύκολης μεταφοράς μεταξύ διαφορετικών επεξεργαστών και λειτουργικών συστημάτων και η ευελιξία στον προγραμματισμό με τη χρήση πολλαπλών γλωσσών συμπληρώνουν με μια πρώτη ματιά τα πιο αξιόλογα πλεονεκτήματα της πλατφόρμας .NET. Σήμερα

υπάρχουν αρκετές εκδόσεις των Windows, τα Windows 95, Windows 98, Windows 98 SE, Windows Millennium, Windows CE, Windows NT 4.0, Windows 2000 και σύντομα πρόκειται να κυκλοφορήσουν και τα 64μπιτα Windows 2000. Τα περισσότερα από αυτά τρέχουν σε επεξεργαστές x86. Η εξαίρεση αυτού του κανόνα έρχεται με τα Windows CE και τα 64μπιτα Windows. Υπάρχουν φυσικά και αρκετές νέες συσκευές στα άμεσα πλάνα των κατασκευαστών που αναμένονται να έχουν κάποιο, διαφορετικό ίσως, τρόπο επικοινωνίας με τον χρήστη. Θεωρητικά, μια εφαρμογή .NET μπορεί να εκτελείται σε λειτουργικά συστήματα διαφορετικά από τα Windows, αρκεί να υπάρχει ένα CLR για κάθε διαφορετική συσκευή, λύνοντας έτσι τα χέρια του προγραμματιστή.



Σχήμα :Αρχιτεκτονική .NET

Ένα ακόμα πλεονέκτημα που αφορά στους προγραμματιστές είναι ότι το Visual Studio .NET επιτρέπει την ενσωμάτωση περισσότερων γλωσσών προγραμματισμού. Μπορεί δηλαδή να δημιουργήσει κανείς μια κλάση σε C++ που να προέρχεται από μια άλλη κλάση σε Visual Basic ή το αντίστροφο.

Η εγκατάσταση μιας εφαρμογής αποτελεί, σε όλες ανεξαιρέτως εκδόσεις των Windows, ένα από τα πιο δύστροπα σημεία. Συνήθως, ο κατασκευαστής

συνοδεύει την εφαρμογή με αρκετά πρόσθετα αρχεία, εκτελέσιμα DLL και OCX, εγγραφές στο μητρώο και συντομεύσεις. Τα Windows 2000 έφεραν το Windows Installer που βελτίωσε κάπως την κατάσταση. Στην πλατφόρμα .NET η Microsoft αποσύνδεσε επιτέλους τα στοιχεία των εφαρμογών από το μητρώο του συστήματος. Στην πράξη, το .NET υπόσχεται να κάνει την εγκατάσταση τόσο απλή όσο μια αντιγραφή των αρχείων από τον ένα κατάλογο στον άλλο και την απεγκατάσταση τόσο εύκολη όσο τη διαγραφή αυτών των αρχείων.

2.6 Visual Basic .NET

Η Visual Basic.NET είναι μία απεικονιστική, οδηγούμενη από συμβάντα αντικειμενοστραφής γλώσσα, στην οποία όλα τα προγράμματα δημιουργούνται χρησιμοποιώντας ένα ολοκληρωμένο περιβάλλον ανάπτυξης IDE (Integrated Development Environment).

Με το IDE, ο προγραμματιστής έχει την δυνατότητα να σχεδιάσει την διεπιφάνεια, να γράψει τον κώδικα, να εκτελέσει και να διορθώσει το πρόγραμμα σε αρκετά σύντομο χρονικό διάστημα. Η διαδικασία της γρήγορης ανάπτυξης μίας εφαρμογής αναφέρεται ως Rapid Application Development - RAD.

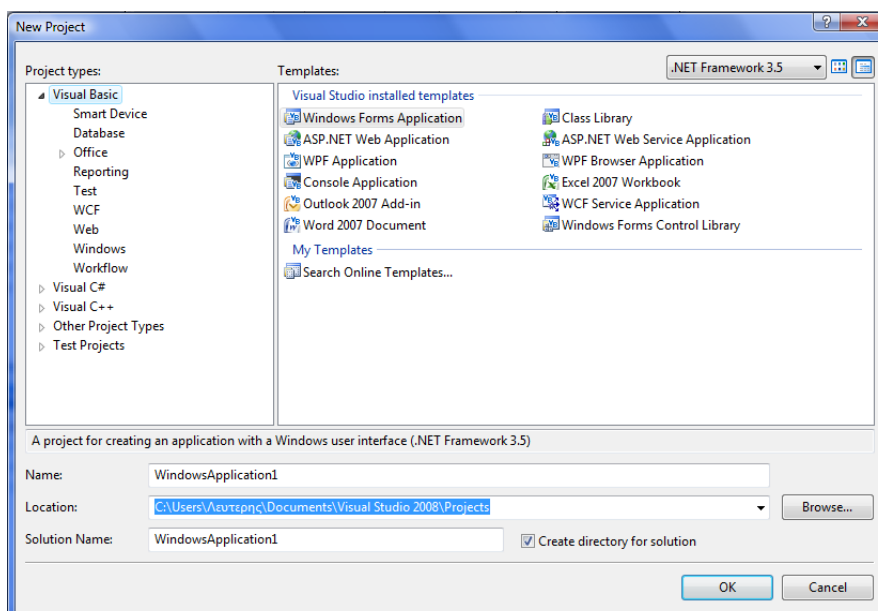
Η VB.NET είναι η πιο διαδεδομένη γλώσσα RAD που χρησιμοποιείται σήμερα.

Η Visual Basic ακολούθησε εντυπωσιακή εξέλιξη την τελευταία δεκαετία, από την έκδοση 1.0 στην σημερινή έκδοση .NET, ή 7.0, όπως είναι η εσωτερική αρίθμηση του προϊόντος. Οι νέες εντολές, οι βελτιώσεις στον τρόπο ανίχνευσης των λαθών αλλά και όσα εργαλεία προστέθηκαν μέχρι τη σημερινή έκδοση, δεν άλλαξαν τον αντικειμενικό σκοπό, που είναι η εύκολη και γρήγορη ανάπτυξη εφαρμογών.

Έτσι, η Visual Basic, μέσα από το Visual Studio, παραμένει ένα εύκολο και ιδιαίτερα γρήγορο εργαλείο για την ανάπτυξη αυτόνομων εφαρμογών αλλά και υπηρεσιών στο Web. Επιλέγοντας μια νέα λύση (Solution), μέσα από το περιβάλλον της Visual Basic, μπορείτε να δημιουργήσετε μια αυτόνομη εφαρμογή (Windows Application), μια εφαρμογή στο Internet (Web Application) ή μια υπηρεσία στο Web (Web Service). Και στις τρεις

περιπτώσεις όλα γίνονται εύκολα, με το σύστημα να κάνει για εσάς την περισσότερη δουλειά.

Ωστόσο, η νέα έκδοση Visual Basic.NET φέρνει αρκετές αλλαγές στην ίδια τη γλώσσα, στο περιβάλλον προγραμματισμού και κατ' επέκταση στη μεταφορά μιας υπάρχουσας εφαρμογής. Οι βελτιώσεις στη γλώσσα περιλαμβάνουν νέες εντολές και σχήματα που φέρνουν πιο κοντά τη VB στις υπόλοιπες αντικειμενοστρεφείς γλώσσες, όπως η C++ και η Java. Από τις πιο σημαντικές βελτιώσεις σε αυτό τον τομέα είναι η δυνατότητα της κληρονομικότητας (inheritance), της δημιουργίας κλάσεων (class) αλλά και της ενσωμάτωσης πιο αυστηρών κανόνων για την αντίδραση σε λάθη (error trapping). Ο προγραμματιστής έχει στη διάθεσή του την εντολή "inherits", που επιτρέπει τη μεταφορά όλων των ιδιοτήτων ενός αντικειμένου ή μιας κλάσης. Υπάρχει επίσης και το function overloading, που προσφέρει τη δυνατότητα να χρησιμοποιούμε τις ίδιες λειτουργίες με διαφορετικές παραμέτρους. Επιπλέον, κανείς είναι σε θέση να δημιουργήσει ένα νέο νήμα (thread) με την εντολή New Thread. Με αυτό τον τρόπο η εκτέλεση μιας εφαρμογής μπορεί να μην είναι σύγχρονη, να μην χρειάζεται δηλαδή να περιμένει μια εντολή το τέλος της προηγούμενης.



Σχήμα :Το Visual Studio απλοποιεί τις διαδικασίες χρησιμοποιώντας έτοιμα πρότυπα εφαρμογών.

Οι εντολές On Error και Resume Next έχουν αντικατασταθεί με πιο σταθερές και συνεπείς λύσεις, όπως οι εντολές Try, Catch και Final, επιτρέποντας τη συγγραφή πιο καθαρού κώδικα. Υπάρχει επίσης η δυνατότητα της αρχικοποίησης μιας μεταβλητής κατά τη δήλωσή της (Dim Y as Integer = 1). Θα βρείτε αλλαγές και σε πολλά ακόμα σημεία που αφορούν σε μεμονωμένα στοιχεία ελέγχου, όπως το μήνυμα MsgBox. Επιπλέον, μια αλλαγή που πιθανόν θα δημιουργήσει προβλήματα με τον κώδικα της προηγούμενης έκδοσης, είναι ο τρόπος αντιμετώπισης των παραμέτρων. Στην έκδοση 6 οι παράμετροι σε μια διαδικασία (function DoSomething(X As Integer)) περνούσαν ως ByVal, που σημαίνει ότι οι τιμές τους μπορούσαν να αλλαχτούν μέσα στη συγκεκριμένη διαδικασία.

Στην έκδοση .NET οι παράμετροι περνούν ως ByVal, που σημαίνει ότι πρέπει να αλλαχτεί ο κώδικας (function DoSomething(ByVal X As Integer)) για να περάσει η σωστή τιμή στον κορμό του προγράμματος. Μια αλλαγή που προβλέπεται επίσης να δημιουργήσει αρκετά προβλήματα είναι τα χαρακτηριστικά των αντικειμένων σε μια φόρμα.

Όπως θα έχετε καταλάβει, οι αλλαγές είναι πολλές και η μεταφορά από την έκδοση 6 απαιτεί την πληκτρολόγηση αρκετού κώδικα. Για αυτόν το λόγο η Microsoft ενσωμάτωσε ένα βοηθητικό εργαλείο (Upgrade Tool), που αναλαμβάνει την αναβάθμιση της εφαρμογής στο νέο περιβάλλον, αλλάζοντας αυτόματα τις εντολές στο νέο φορμά. Αυτό το εργαλείο ενεργοποιείται αυτόματα αν ανοίξετε κάποια παλιότερη εφαρμογή μέσα από το Visual Studio .NET. Το αποτέλεσμα είναι να δημιουργηθεί ένα νέο έργο, αφήνοντας τα αρχικά αρχεία της έκδοσης 6.0 όπως έχουν. Δύσκολα όμως θα αποφύγετε την επέμβαση στον κώδικα, αφού τις περισσότερες φορές η αυτόματη αναβάθμιση θα δημιουργήσει αρκετές ασυμβατότητες.

Ένα ακόμα ενδιαφέρον χαρακτηριστικό στις φόρμες της Visual Basic .NET είναι η μεγαλύτερη ακρίβεια. Χαρακτηριστικά όπως το Menu Designer, Control Anchoring, Control Docking και μια σειρά από νέα στοιχεία ελέγχου επιτρέπουν να αναπτύξει κανείς πιο καλαίσθητες φόρμες. Σε αυτό έρχεται να προστεθεί και η υποστήριξη του GDI+, του διαδόχου του GDI (Graphic Device Interface) που συναντούσαμε στις προηγούμενες εκδόσεις των Windows, το οποίο προσφέρει πιο πλούσια χαρακτηριστικά. Έτσι, μπορεί κανείς εύκολα να

σχεδιάσει οθόνες που εμφανίζονται σε διαφανή και πολυεπίπεδα παράθυρα, χαρακτηριστικό που επιτρέπουν τα Windows 2000 και τα Windows XP. Άλλο ένα ενδιαφέρον στοιχείο της Visual Basic .NET είναι η πλήρης υποστήριξη του Unicode, ακόμα και στις φόρμες.

Μαζί με το Visual Studio έρχεται και μια έκδοση του Crystal Reports, της γνωστής εφαρμογής για εύκολη και γρήγορη κατασκευή εκτυπώσεων, με τις ανάλογες επεκτάσεις για την υποστήριξη εκτυπώσεων σε εφαρμογές για το Internet.

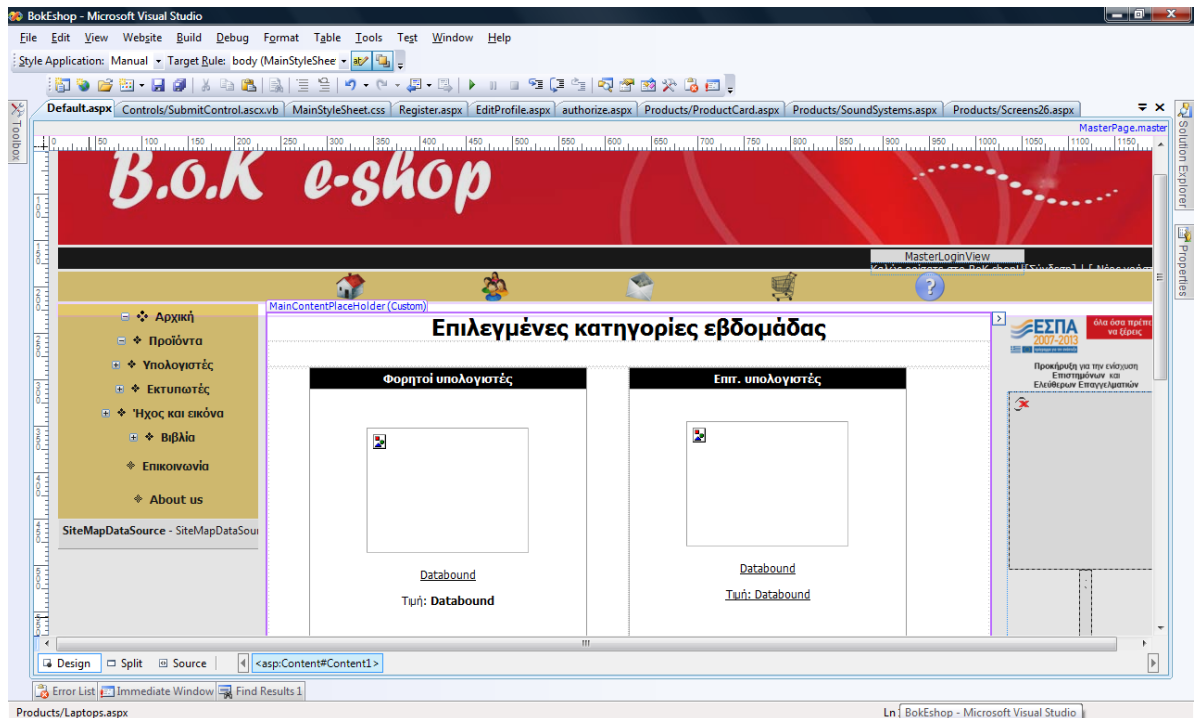
2.7 Web Forms και ASP .NET

Η Visual Basic ακολούθησε εντυπωσιακή εξέλιξη την τελευταία δεκαετία, από την έκδοση 1.0 στην σημερινή έκδοση .NET, ή 7.0, όπως είναι η εσωτερική αρίθμηση του προϊόντος. Οι νέες εντολές, οι βελτιώσεις στον τρόπο ανίχνευσης των λαθών αλλά και όσα εργαλεία προστέθηκαν μέχρι τη σημερινή έκδοση, δεν άλλαξαν τον αντικειμενικό σκοπό, που είναι η εύκολη και γρήγορη ανάπτυξη εφαρμογών.

Τα Web Form (φόρμες για το Web) αποτελούν μια καινούρια μέθοδο, μια νέα προσέγγιση στον τρόπο δημιουργίας εφαρμογών για το Internet. Πρακτικά γεφυρώνει το κενό μεταξύ της ανάπτυξης αυτόνομων εφαρμογών για τα Windows και για το Internet, καθώς χρησιμοποιεί τα ίδια εργαλεία, κενές φόρμες, μια εργαλειοθήκη και κώδικα σε Visual Basic (ή οποιαδήποτε άλλη γλώσσα). Απαραίτητη προϋπόθεση είναι η ύπαρξη του αντίστοιχου διακομιστή της πλατφόρμας .NET (Windows 2000 και το .NET FrameWork).

Με τη χρήση της Web Form ο χρήστης έχει την δυνατότητα να εισάγει δεδομένα τα οποία στέλνονται στον server για επεξεργασία.

Για παράδειγμα, οι Web Forms μπορεί να χρησιμοποιηθούν για αγορές προϊόντων, για ανάκτηση δεδομένων, μηχανές αναζήτησης.



Σχήμα : Ο σχεδιασμός μιας φόρμας (Web Form) για το Internet.

Η δημιουργία μιας εφαρμογής για το Internet γίνεται με τον ίδιο τρόπο, μέσα από το Visual Studio .NET, επιλέγοντας το ASP .NET Web Application αντί του Windows Application, το οποίο χρησιμοποιείται για τις "πατροπαράδοτες" παραθυρικές εφαρμογές.

Στη συνέχεια, πρέπει να επιλέξει κανείς τον τόπο όπου θα "φιλοξενηθεί" η εφαρμογή, ο οποίος μπορεί να είναι ένας απομακρυσμένος διακομιστής ή ένας διακομιστής στο τοπικό δίκτυο. Στην πρώτη περίπτωση το Visual Studio θα χρησιμοποιήσει τις επεκτάσεις του FrontPage και το πρωτόκολλο HTTP για να επικοινωνήσει με το διακομιστή και να δημιουργήσει τα τμήματα που αποτελούν την εφαρμογή. Οι διαφορές στο κατασκευαστικό κομμάτι της εφαρμογής είναι παρ' όλα αυτά λίγες, αφού ο χρήστης έχει στη διάθεσή του μια κενή φόρμα και την εργαλειοθήκη με την οποία έχει συνηθίσει να δουλεύει.

Η δημιουργία μιας απλής εφαρμογής, στην οποία ο τελικός χρήστης θα μπορεί να εισαγάγει το όνομα και τον κωδικό πρόσβασης, είναι υπόθεση λίγων λεπτών. Αρκεί να "τραβήξει" κανείς δύο αντικείμενα τύπου Label για τις περιγραφές, δύο άλλα αντικείμενα TextBox και ένα κουμπί εντολών. Το σκεπτικό που ακολουθούν οι φόρμες για το Web είναι το ίδιο όπως και στον κλασικό προγραμματισμό. Με διπλό κλικ, για παράδειγμα, πάνω στο κουμπί εντολών μπορείτε να γράψετε τον κώδικα που θέλετε και να αλλάξετε

οποιοδήποτε από τα χαρακτηριστικά των αντικειμένων που βρίσκονται πάνω στη φόρμα. Αν χρησιμοποιήσετε ελληνικούς χαρακτήρες στις επιλογές, τα μενού και τα πεδία κειμένου, καλό είναι να επιλέξετε, με δεξί κλικ πάνω στη φόρμα, το ελληνικό σετ χαρακτήρων.

Μια φόρμα Web Form αποτελείται από δύο μέρη: το αρχείο HTML που αντιπροσωπεύει την οπτική απεικόνιση της σελίδας, έτσι όπως θα τη δει ο τελικός χρήστης στο PC, και ένα αρχείο με κώδικα σε VB που χειρίζεται τα γεγονότα, τα αντικείμενα και τα αποτελέσματα στη συγκεκριμένη σελίδα. Ο διακομιστής ανταλλάσσει στοιχεία και δεδομένα σε μορφή XML μέσα από τις φόρμες Web Form και εμφανίζει τα αποτελέσματα σε μορφή HTML 3.2. Αυτό σημαίνει ότι τα αποτελέσματα είναι δυνατόν να εμφανιστούν μέσα από οποιοδήποτε πρόγραμμα αναζήτησης. Τα αρχεία HTML έχουν τη μορφή .aspx, η οποία είναι η νέα έκδοση των σελίδων .asp, που επιτρέπει την πολύ καλύτερη διαχείριση της VB .NET μέσα στην ίδια την ιστοσελίδα. Θα δείτε ότι μια φόρμα με το όνομα WebForm, για παράδειγμα, αποτελείται από δύο αρχεία, το WebForm.aspx και το WebForm.vb με κώδικα HTML και VB αντίστοιχα.

Φυσικά, μπορείτε να γράψετε απλό κώδικα HTML και VB μέσα στην ίδια ιστοσελίδα από το FrontPage ή έναν άλλο editor, αρκεί να την αποθηκεύσετε σε ένα διακομιστή που διαθέτει την υποστήριξη του .NET Framework και το συγκεκριμένο αρχείο να έχει την κατάληξη .aspx. Δουλεύοντας με τη νέα δομή των ιστοσελίδων ASP .NET, θα δείτε ότι υπάρχουν πολύ μεγάλες αλλαγές και για αυτό δεν ενδείκνυται καθόλου να προσπαθήσετε να μεταφέρετε τον παλιό σας κώδικα στη νέα μορφή. Αλλωστε, ένα διακομιστής Windows 2000 με το .NET Framework υποστηρίζει και τις δύο εκδόσεις.

3. Κεφάλαιο – Περιγραφή της Βάσης Δεδομένων

3.1 Βάσεις Δεδομένων και SQL Server 2005

Η ανάγκη για την χρήση μιας Βάσης Δεδομένων ήταν εξαρχής φανερή, λόγω της φύσεως του προβλήματος. Έπρεπε όλα τα δεδομένα να είναι αποθηκευμένα σε κάποιο κεντρικό σύστημα ώστε να είναι απ' ευθείας προσβάσιμα από την Web εφαρμογή και συνεπώς διαθέσιμα στο χρήστη.

Αρχικά έπρεπε να επιλεγεί ποια Βάση Δεδομένων θα χρησιμοποιηθεί. Οι επιλογές ήταν πολλές όπως Oracle, mySQL, Access, SQL Server 2000 κ.α. Τα κριτήρια επιλογής της βάση ήταν να υποστηρίζει πολλούς χρήστες ταυτόχρονα, να υπάρχει η κατάλληλη τεχνογνωσία και να είναι τύπου server. Τελικά επιλέχθηκε ο SQL Server γιατί ήταν η Βάση Δεδομένων που πληρούσε τα περισσότερα κριτήρια. Η επίσης διαθέσιμη Access απορρίφτηκε διότι υποστηρίζει μέχρι δέκα χρήστες ταυτόχρονα.

Το Microsoft SQL Server είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (relational database management system, RDBMS), σχεδιασμένο για εφαρμογές επεξεργασίας συναλλαγών εντός επικοινωνίας (online transaction processing, OLTP), αποθηκών δεδομένων (data warehousing), και ηλεκτρονικού εμπορίου (e-commerce) με υψηλές απαιτήσεις.

Το Microsoft SQL Server 2005 είναι σχεδιασμένο για να υποστηρίζει βάσεις δεδομένων μεγάλου όγκου και κρίσιμης σημασίας σε πολλά διαφορετικά πεδία εφαρμογών, στα οποία περιλαμβάνονται η επεξεργασία συναλλαγών εντός επικοινωνίας (online transaction processing, OLTP), οι αποθήκες δεδομένων (data warehousing) και το ηλεκτρονικό εμπόριο (e-commerce). Για την υποστήριξη αυτών των λειτουργιών, το SQL Server διαθέτει ορισμένα εργαλεία, στα οποία περιλαμβάνονται κάποια βοηθητικά προγράμματα γραμμής διαταγών όπως τα SQL Server Configuration Manager, SQL Server Error and Usage Reporting, SQL Server Surface Area Configuration και το SQL Server Management Studio Express , ένα προηγμένο παραστατικό εργαλείο για τη διαχείριση πολλών βάσεων δεδομένων και του ίδιου του SQL Server.

3.2 Τι είναι το SQL Server Management Studio Express:

Είναι το βασικό εργαλείο που χρησιμοποιείται για τη διαχείριση του SQL Server 2000

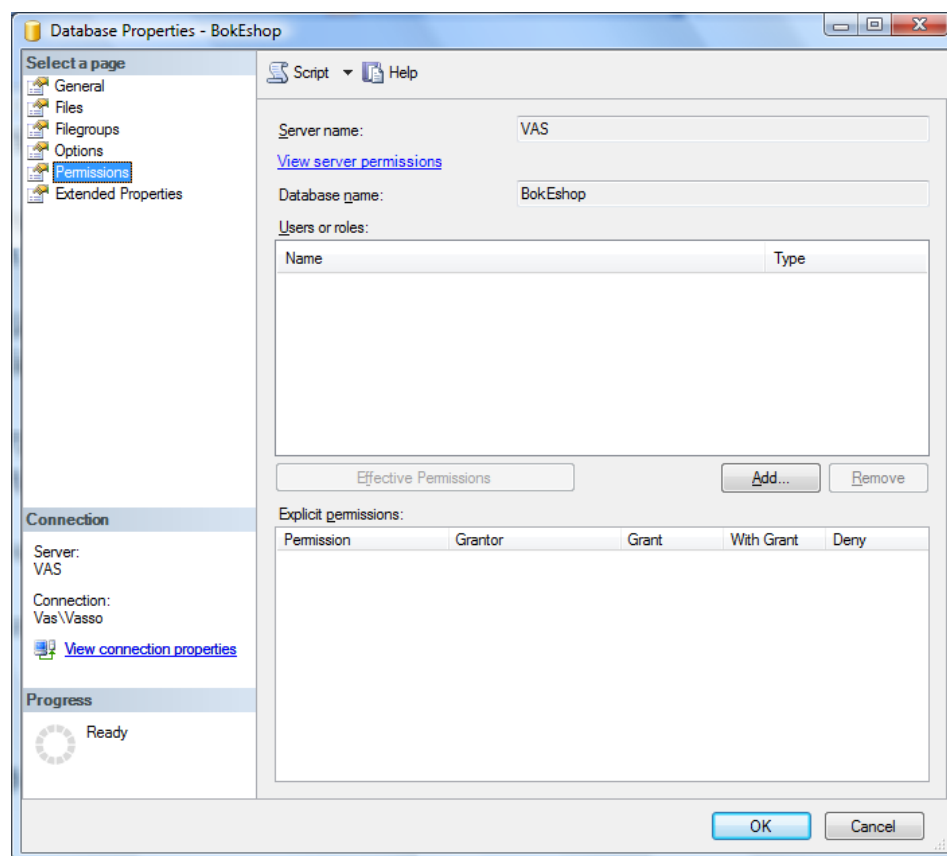
Χρησιμοποιώντας τη γραφική αναπαράσταση του SQL Server Management Studio Express, μπορείτε να κάνετε τα εξής:

-Να ορίζετε ομάδες παρουσιών (instances) του SQL Server και να δηλώνεται μεμονωμένους διακομιστές μέσα σε μία ομάδα.

-Να διευθετείτε όλες τις επιλογές του SQL Server για τον κάθε δηλωμένο διακομιστή.

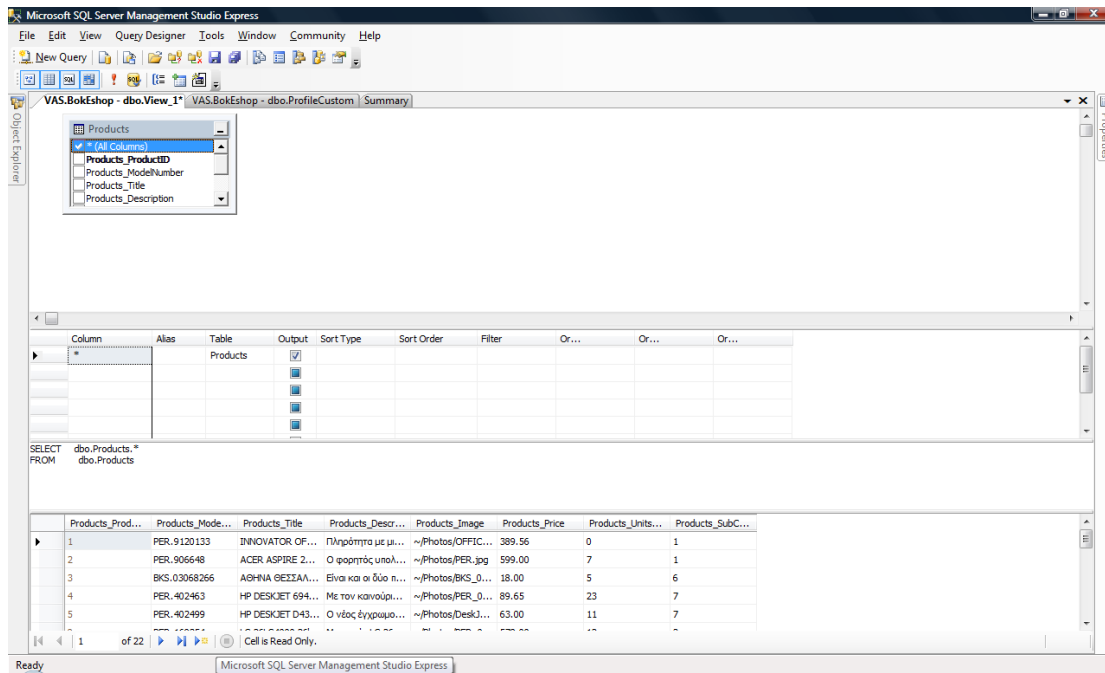
-Να δημιουργείτε και να διαχειρίζεστε όλες τις βάσεις δεδομένων, τα αντικείμενα, τους

λογαριασμούς σύνδεσης χρηστών (logins), τους χρήστες, και τις άδειες του SQL Server στον κάθε δηλωμένο διακομιστή.



-Να ορίζετε και να εκτελείτε όλες τις διαχειριστικές εργασίες του SQL Server στον κάθε δηλωμένο διακομιστή.

-Να βλέπετε το περιεχόμενο πινάκων και προβολών χρησιμοποιώντας το παράθυρο σχεδίασης ερωτήματος (Query Designer).

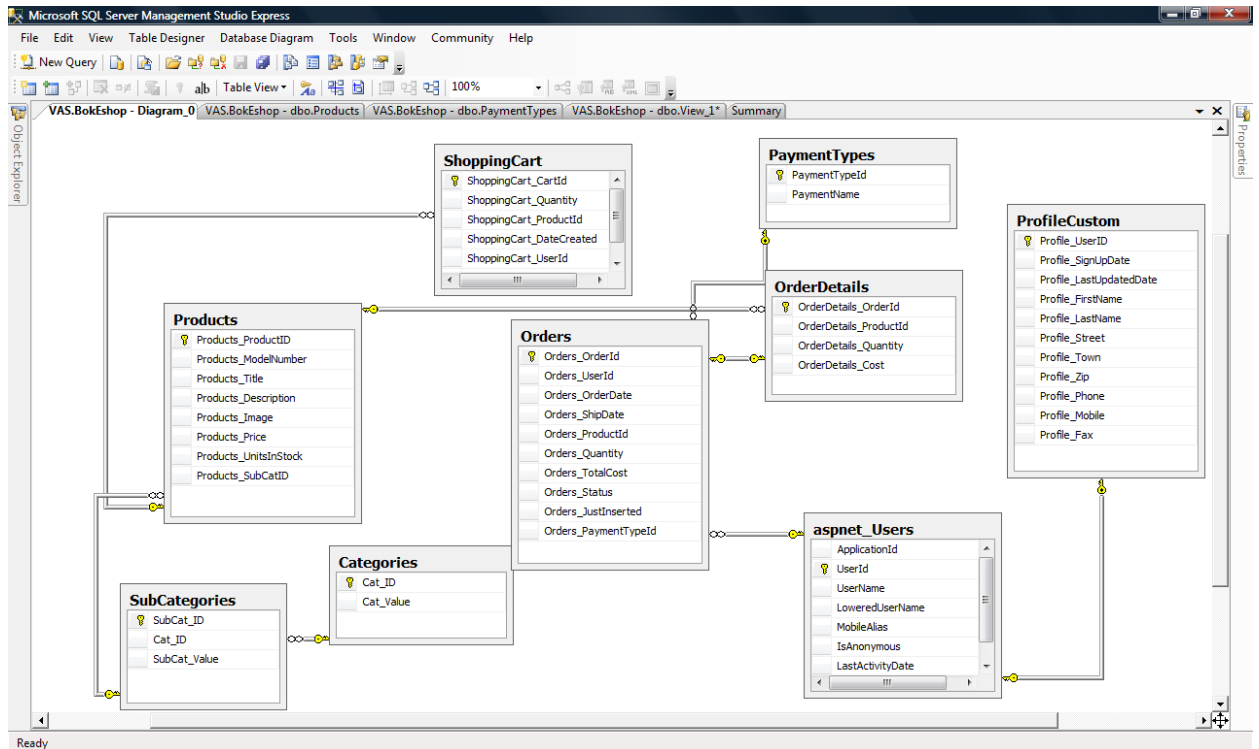


-Να σχεδιάζετε και να δοκιμάζετε εντολές,δέσμες εντολών,και σενάρια SQL αλληλεπιδραστικά, καλώντας τον Query Analyzer (Αναλυτής Ερωτημάτων) του SQL Server

-Να καλείτε τους διάφορους οδηγούς (wizards)του SQL Server.

3.3 Πίνακες της Βάσης Δεδομένων

Η βάση δεδομένων αποτελείται από τους παρακάτω πίνακες:



- Πίνακας dbo.asp_users:

Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
UserId	uniqueidentifier	<input type="checkbox"/>
UserName	nvarchar(256)	<input type="checkbox"/>
LoweredUserName	nvarchar(256)	<input type="checkbox"/>
MobileAlias	nvarchar(16)	<input checked="" type="checkbox"/>
IsAnonymous	bit	<input type="checkbox"/>
LastActivityDate	datetime	<input type="checkbox"/>

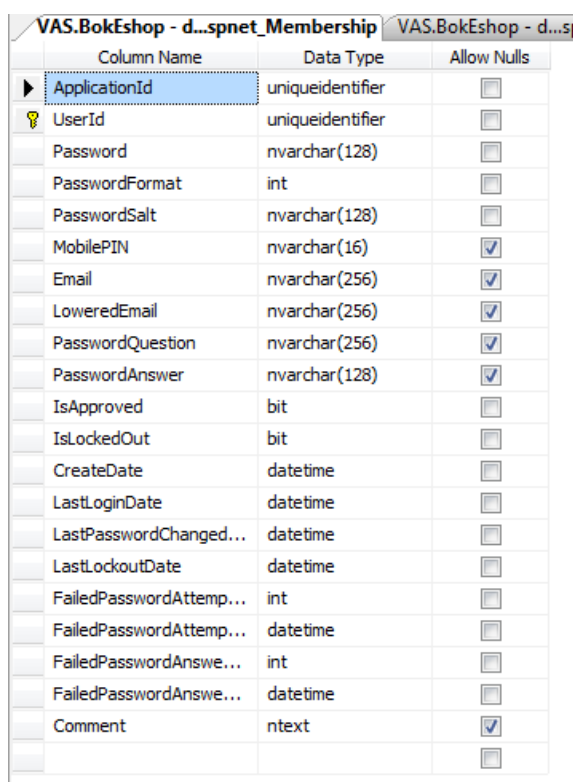
Σχήμα Πίνακας dbo.asp_users

Στο παραπάνω πίνακα γίνεται η εισαγωγή των στοιχείων του πελάτη για την είσοδο του στην σελίδα. Χρησιμοποιήσαμε το LoginControl της ASP, το οποίο δημιουργεί κάποιους πίνακες στην βάση δεδομένων οι οποίοι κρατάνε τα απαραίτητα στοιχεία.

Μία εγγραφή για κάθε πελάτη. Ο πίνακας περιλαμβάνει το UserName του χρήστη και το UserId τα οποία χρησιμοποιούμε για να προσπελάσουμε διαφορά στοιχεία του χρήστη.

Επίσης δημιουργείται ένας πίνακας που λέγεται asp_Membership, ο οποίος περιλαμβάνει διάφορα στοιχεία όπως το UserId, την ερώτηση ασφαλείας, την απάντηση ασφαλείας σε μορφή hash όπως και τον κωδικό σε μορφή hash, ώστε να είναι τα στοιχεία του χρήστη ασφαλή και μη προσβάσιμα.

- Πίνακας dbo.Membership:



Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
UserId	uniqueidentifier	<input type="checkbox"/>
Password	nvarchar(128)	<input type="checkbox"/>
PasswordFormat	int	<input type="checkbox"/>
PasswordSalt	nvarchar(128)	<input type="checkbox"/>
MobilePIN	nvarchar(16)	<input checked="" type="checkbox"/>
Email	nvarchar(256)	<input checked="" type="checkbox"/>
LoweredEmail	nvarchar(256)	<input checked="" type="checkbox"/>
PasswordQuestion	nvarchar(256)	<input checked="" type="checkbox"/>
PasswordAnswer	nvarchar(128)	<input checked="" type="checkbox"/>
IsApproved	bit	<input type="checkbox"/>
IsLockedOut	bit	<input type="checkbox"/>
CreateDate	datetime	<input type="checkbox"/>
LastLoginDate	datetime	<input type="checkbox"/>
LastPasswordChanged...	datetime	<input type="checkbox"/>
LastLockoutDate	datetime	<input type="checkbox"/>
FailedPasswordAttemp...	int	<input type="checkbox"/>
FailedPasswordAttemp...	datetime	<input type="checkbox"/>
FailedPasswordAnsw...	int	<input type="checkbox"/>
FailedPasswordAnsw...	datetime	<input type="checkbox"/>
Comment	ntext	<input checked="" type="checkbox"/>

Σχήμα Πίνακας asp_Membership

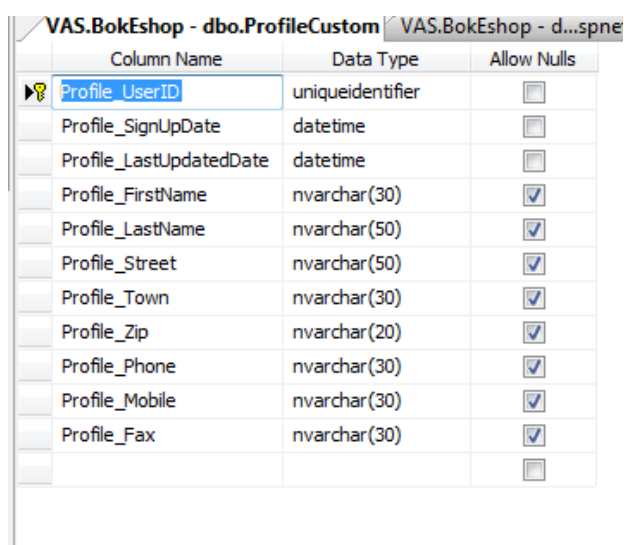
Οι πίνακες αυτοί δημιουργούνται με την εντολή Aspnet_regsql.exe <options>, όπου options τα στοιχεία της βάσης, όπως το όνομα του Server, το όνομα της βάσης και ο κωδικός της.

Ο χρήστης μπορεί να ανακτήσει τον κωδικό του στην περίπτωση που τον ξεχάσει μέσω της σελίδας pwdRecover.aspx όπου χρησιμοποιείται το control PasswordRecovery και ο χρήστης

πρέπει να απαντήσει στην ερώτηση ασφαλείας. Αν η απάντηση είναι αυτή που είχε πληκτρολογήσει κατά την εγγραφή του, τότε του αποστέλλεται ένα email με έναν νέο κωδικό, τον οποίο μπορεί να αλλάξει.

Για να πάρουμε το προφίλ του χρήστη, δηλαδή ονοματεπώνυμο, διεύθυνση κτλ, δημιουργήσαμε έναν πίνακα, τον ProfileCustom.

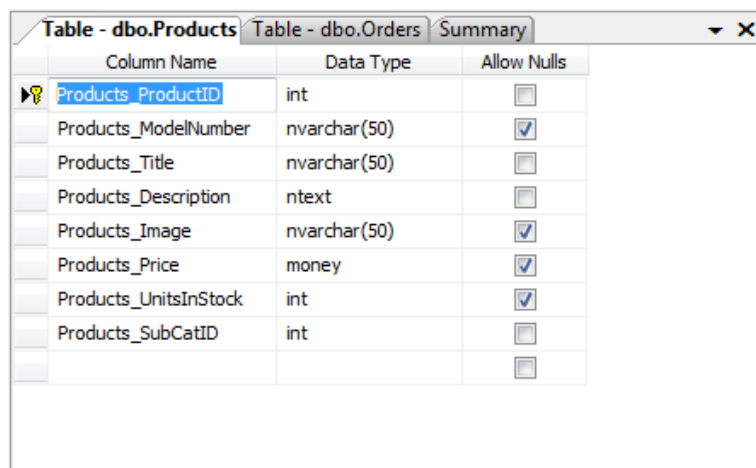
- Πίνακας dbo.ProfileCustom:



Column Name	Data Type	Allow Nulls
Profile_UserID	uniqueidentifier	<input type="checkbox"/>
Profile_SignUpDate	datetime	<input type="checkbox"/>
Profile_LastUpdatedDate	datetime	<input type="checkbox"/>
Profile_FirstName	nvarchar(30)	<input checked="" type="checkbox"/>
Profile_LastName	nvarchar(50)	<input checked="" type="checkbox"/>
Profile_Street	nvarchar(50)	<input checked="" type="checkbox"/>
Profile_Town	nvarchar(30)	<input checked="" type="checkbox"/>
Profile_Zip	nvarchar(20)	<input checked="" type="checkbox"/>
Profile_Phone	nvarchar(30)	<input checked="" type="checkbox"/>
Profile_Mobile	nvarchar(30)	<input checked="" type="checkbox"/>
Profile_Fax	nvarchar(30)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Ο πίνακας περιλαμβάνει το όνομα, το επίθετο, την διεύθυνση του πελάτη, την πόλη, τον ταχυδρομικό κώδικα, την χώρα διαμονής, το e-mail, του, τον αριθμό τηλεφώνου, το κινητό του τηλέφωνο, το φαξ και την ημερομηνία εγγραφής του στην σελίδα μας. Παρατηρούμε πως πρωτεύων κλειδί είναι το πεδίο Profile_UserID το οποίο συνδέεται με τον πίνακα asp_Users και παίρνει αυτόματα τον αριθμό από εκεί.

- Πίνακας dbo.Product



Column Name	Data Type	Allow Nulls
Products_ProductID	int	<input type="checkbox"/>
Products_ModelNumber	nvarchar(50)	<input checked="" type="checkbox"/>
Products_Title	nvarchar(50)	<input type="checkbox"/>
Products_Description	ntext	<input type="checkbox"/>
Products_Image	nvarchar(50)	<input checked="" type="checkbox"/>
Products_Price	money	<input checked="" type="checkbox"/>
Products_UnitsInStock	int	<input checked="" type="checkbox"/>
Products_SubCatID	int	<input type="checkbox"/>
		<input type="checkbox"/>

Σχήμα Πίνακας dbo.Product

Ο πίνακας dbo.Product περιέχει πληροφορίες σχετικές με τα προϊόντα του καταστήματος. Ο πίνακας Product περιλαμβάνει τον κωδικό του προϊόντος, το μοντέλο, τον τίτλο, την περιγραφή, την τιμή και την εικόνα του προϊόντος. Περιέχει πληροφορίες όπως το πόσα προϊόντα υπάρχουν στην αποθήκη (το πεδίο UnitsInStock). Τέλος παρατηρούμε πως πρωτεύων κλειδί είναι το πεδίο ProductID το οποίο παίρνει αυτόματα ένα αύξοντα αριθμό από την βάση.

- Πίνακας dbo.Order

Column Name	Data Type	Allow Nulls
Orders_OrderId	int	<input type="checkbox"/>
Orders_UserId	uniqueidentifier	<input type="checkbox"/>
Orders_OrderDate	datetime	<input type="checkbox"/>
Orders_ShipDate	datetime	<input checked="" type="checkbox"/>
Orders_ProductId	int	<input type="checkbox"/>
Orders_Quantity	int	<input type="checkbox"/>
Orders_TotalCost	money	<input checked="" type="checkbox"/>
Orders_Status	bit	<input checked="" type="checkbox"/>
Orders_JustInserted	bit	<input checked="" type="checkbox"/>
Orders_PaymentTypeId	int	<input type="checkbox"/>

Σχήμα Πίνακας Order

Ο παραπάνω πίνακας περιέχει πεδία τα οποία είναι απαραίτητα, για την ορθή και ομαλή λειτουργία, όλων των σταδίων που οδηγούν στην ολοκλήρωση μιας παραγγελίας, όπως το ποιος έκανε την παραγγελία, το πότε, ποιο προϊόν, την ποσότητα, την κατάσταση παραγγελίας και τον τρόπο πληρωμής. Πρωτεύων κλειδί του πίνακα Order είναι το πεδίο OrderID το οποίο παίρνει αυτόματα ένα αύξοντα αριθμό από την βάση.

- Πίνακας dbo.OrderDetails

Column Name	Data Type	Allow Nulls
OrderDetails_OrderId	int	<input type="checkbox"/>
OrderDetails_ProductId	int	<input type="checkbox"/>
OrderDetails_Quantity	int	<input type="checkbox"/>
OrderDetails_Cost	money	<input type="checkbox"/>

Σχήμα Πίνακας OrderDetails

Ο παραπάνω πίνακας περιέχει τέσσερα πεδία, το ID της παραγγελίας, το ID του προϊόντος και τον αριθμό των τεμαχίων για κάθε προϊόν και την τιμή. Ο πίνακας αυτός συνδέεται με τον Orders και παίρνει το OrderId από εκεί.

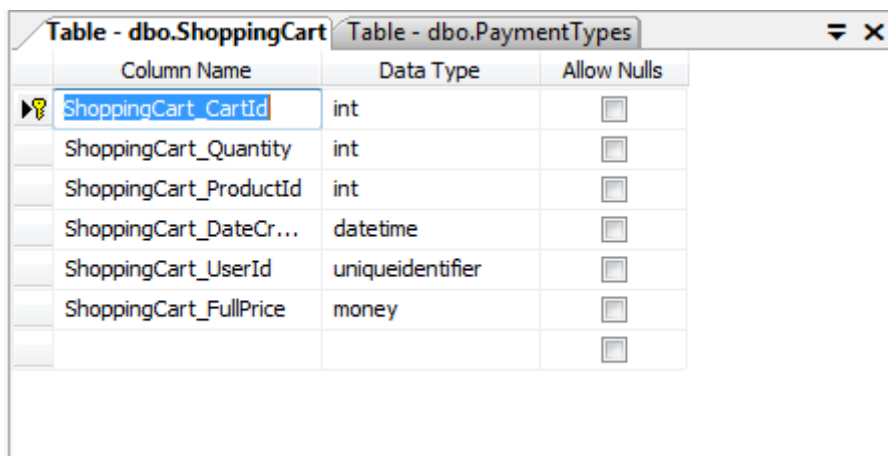
- Πίνακας dbo.Payment

Column Name	Data Type	Allow Nulls
PaymentTypeId	int	<input type="checkbox"/>
PaymentName	nvarchar(50)	<input type="checkbox"/>

Σχήμα Πίνακας Payment

Ο παραπάνω πίνακας περιέχει τον τρόπο πληρωμής και τον κωδικό του τρόπου πληρωμής ο οποίος είναι και το πρωτεύον κλειδί του πίνακα. Ο πίνακας αυτός συνδέεται με τον Orders ώστε να ενημερώνονται οι παραγγελίες για τον τρόπο πληρωμής.

- Πίνακας dbo.ShoppingCart



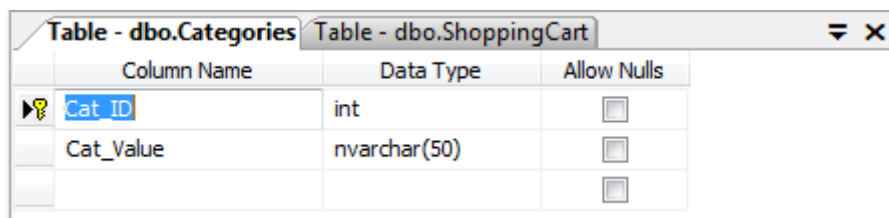
Column Name	Data Type	Allow Nulls
ShoppingCart_CartId	int	<input type="checkbox"/>
ShoppingCart_Quantity	int	<input type="checkbox"/>
ShoppingCart_ProductId	int	<input type="checkbox"/>
ShoppingCart_DateCr...	datetime	<input type="checkbox"/>
ShoppingCart_UserId	uniqueidentifier	<input type="checkbox"/>
ShoppingCart_FullPrice	money	<input type="checkbox"/>

Σχήμα Πίνακας ShoppingCart

Ο παραπάνω πίνακας περιέχει τον κωδικό του «καλαθιού», τον κωδικό του προϊόντος, τον αριθμό των προϊόντων (τεμαχίων) που έχουμε προσθέσει στο καλάθι αγορών και τέλος την ημερομηνία κατά την οποία το προϊόν τοποθετήθηκε στο καλάθι .

Ο πίνακας αυτός είναι προσωρινός καθώς όταν ολοκληρώνεται μία παραγγελία το καλάθι αδειάζει για να μην πιάνει χώρο στην μνήμη.

- Πίνακας dbo.Categories



Column Name	Data Type	Allow Nulls
Cat_ID	int	<input type="checkbox"/>
Cat_Value	nvarchar(50)	<input type="checkbox"/>

Σχήμα Πίνακας Categories

Ο παραπάνω πίνακας περιέχει τον κωδικό και το όνομα της κατηγορίας. Ο πίνακας αυτός χρησιμοποιείται για την εμφάνιση των διαθέσιμων κατηγοριών των προϊόντων. Τέλος παρατηρούμε πως πρωτεύον κλειδί είναι το πεδίο CategoryID το οποίο παίρνει αυτόματα ένα αύξοντα αριθμό από την βάση.

- Πίνακας dbo.SubCategories

Column Name	Data Type	Allow Nulls
SubCat_ID	int	<input type="checkbox"/>
Cat_ID	int	<input type="checkbox"/>
SubCat_Value	nvarchar(50)	<input type="checkbox"/>
		<input type="checkbox"/>

Σχήμα Πίνακας SubCategories

Ο παραπάνω πίνακας περιέχει τον κωδικό της υποκατηγορίας, τον κωδικό της κατηγορίας στην οποία ανήκει και το όνομα της υποκατηγορίας. Ο πίνακας αυτός χρησιμοποιείται για την εμφάνιση των διαθέσιμων υποκατηγοριών των προϊόντων. Τέλος παρατηρούμε πως πρωτεύων κλειδί είναι το πεδίο SubCat_ID το οποίο παίρνει αυτόματα ένα αύξοντα αριθμό από την βάση.

3.4 Αποθηκευμένες Διεργασίες - Stored Procedure

Οι αποθηκευμένες διαδικασίες ή αλλιώς stored procedure που χρησιμοποιήθηκαν στην εφαρμογή, είναι διαδικασίες οι οποίες αποθηκεύονται μέσα στον SQL SERVER και οποιαδήποτε στιγμή χρειαστεί να εκτελεστεί κάποια απ' αυτές, καλείται από την εφαρμογή και εκτελείται.

Ενδεικτικά μερικά από τα stored procedure που χρησιμοποιήθηκαν είναι:

- Select storeprocedure

```
DECLARE @ApplicationId UNIQUEIDENTIFIER
SET @ApplicationId = NULL
```

```
SELECT @Profile_SignUpDate = Profile_SignUpDate, @Profile_FirstName =
Profile_FirstName, @Profile_LastName = Profile_LastName,
@Profile_Street = Profile_Street, @Profile_Zip = Profile_Zip, @Profile_Town =
Profile_Town, @Profile_Phone = Profile_Phone,
@Profile_Mobile = Profile_Mobile, @Profile_Fax = Profile_Fax
FROM ProfileCustom, aspnet_Users
WHERE aspnet_Users.ApplicationId = @ApplicationID AND aspnet_Users.UserName =
@UserName AND aspnet_Users.UserId = ProfileCustom.Profile_UserID
```


- Update – insert storeprocedure

```

DECLARE @UserId UNIQUEIDENTIFIER
SELECT @userId = UserId
FROM aspnet_Users
WHERE ApplicationID = @ApplicationID AND LoweredUserName = LOWER(@UserName)

IF (@UserId IS NULL)
EXEC aspnet_Users_CreateUser @ApplicationID, @UserName, @IsUserAnonymous,
@CurrentUtcDate, @UserId OUTPUT

IF EXISTS(SELECT 1 FROM ProfileCustom WHERE Profile_UserId = @UserId)
BEGIN
UPDATE ProfileCustom SET Profile_LastUpdatedDate = @CurrentUtcDate,
Profile_FirstName = @Profile_FirstName,
Profile_LastName = @Profile_LastName, Profile_Street = @Profile_Street,
Profile_Zip = @Profile_Zip, Profile_Town = @Profile_Town, Profile_Phone =
@Profile_Phone, Profile_Mobile = @Profile_Mobile,
Profile_Fax = @Profile_Fax
WHERE Profile_UserId = @UserId
END

ELSE
BEGIN
INSERT INTO ProfileCustom(Profile_UserId, Profile_FirstName, Profile_LastName,
Profile_Street, Profile_Zip, Profile_Town,
Profile_Phone, Profile_Mobile, Profile_Fax)
VALUES(@UserId, @Profile_FirstName, @Profile_LastName, @Profile_Street,
@Profile_Zip, @Profile_Town, @Profile_Phone, @Profile_Mobile,
@Profile_Fax)
END

```

Τα stored procedure προσφέρουν :

- ΕΥΚΟΛΙΑ : Όταν ένα stored procedure τροποποιηθεί στον SQL, αν το όνομά του και οι μεταβλητές, που χρειάζεται, παραμείνουν ίδια δεν είναι αναγκαίο να επέμβει κάποιος στον κώδικα ASP.NET.
- ΤΑΧΥΤΗΤΑ ΣΤΗΝ ΕΦΑΡΜΟΓΗ : Τα stored procedure εκτελούνται γρήγορα διότι δεν χρειάζεται να περάσουν οι παράμετροι στις υπορουτίνες κάθε φορά που καλούνται. Αυτό γίνεται μόνο την πρώτη φορά που καλούνται τα stored procedure.
- ΚΑΛΥΤΕΡΗ ΧΡΗΣΗ ΤΩΝ ΠΟΡΩΝ : Τα stored procedure εκτελούνται στον SQL, δεν επιβαρύνουν το σύστημα του χρήστη και

εκμεταλλεύονται καλύτερα το δίκτυο, αφού ο πελάτης δεν στέλνει μέσω δικτύου μεγάλα ερωτήματα, αλλά μόνο το όνομα του stored procedure με τις απαραίτητες παραμέτρους. Ο server λαμβάνει το μήνυμα του πελάτη, εκτελεί το stored procedure και το αποτέλεσμα αποστέλλεται στον πελάτη. Τέλος τα stored procedure εκμεταλλεύονται την ταχύτητα του SQL server και γενικά το υλικό (Hardware) του.

- ΑΞΙΟΠΙΣΤΙΑ - ΣΤΑΘΕΡΟΤΗΤΑ - ΑΝΕΞΑΡΤΗΣΙΑ : Όλοι οι χρήστες της Βάσης Δεδομένων μπορούν να χρησιμοποιούν τα ίδια stored procedure χωρίς να παίρνουν μεγάλο ρίσκο λάθους και είναι ευκολότερη η διόρθωση των διεργασιών αυτών.

4. Κεφάλαιο – Ανάλυση Διαδικασιών σε ASP.NET

4.1 Επικύρωση για τις Web φόρμες - ASP.NET Validation Controls

Τίποτα δεν μπορεί να καταστρέψει πιο γρήγορα ένα πρόγραμμα από να επιτραπεί στα "κακά" δεδομένα να εισέλθουν σε διαδικασίες εισαγωγής δεδομένων.

Τα Validation Server Controls είναι ένα γκρουπ από ASP.NET controls τα οποία επιτρέπουν να γίνεται επικύρωση των στοιχείων εισόδου του χρήστη και να εμφανίσουν ένα μήνυμα όταν αποτύχει η επικύρωση των στοιχείων.

Στο ASP.NET, μπορείτε να προσθέσετε στοιχεία ελέγχου επικύρωσης σε όλα σχεδόν τα στοιχεία ελέγχου δεδομένων εισόδου της σελίδας, που είναι είτε HTML στοιχεία ελέγχου διακομιστή, είτε στοιχεία ελέγχου διακομιστή μιας Web φόρμας. Μπορούν να ορισθούν περισσότερα από ένα είδη επικύρωσης σε ένα συγκεκριμένο στοιχείο ελέγχου εισόδου, κάνοντας δυνατή την επικύρωση με διάφορα κριτήρια. Όταν εκτελείται η ASP.NET σελίδα με τα στοιχεία ελέγχου επικύρωσης, οι τιμές των στοιχείων ελέγχου εισόδου που επηρεάζονται από τα στοιχεία ελέγχου επικύρωσης υπόκεινται σε επεξεργασία σε σχέση με την αντίστοιχη λογική που τα διέπει, όπως αναφέρεται στα στοιχεία ελέγχου επικύρωσης.. Μετά τους υπολογισμούς, οι ιδιότητες των στοιχείων ελέγχου επικύρωσης παίρνουν την τιμή True ή False, ανάλογα με το αποτέλεσμα. Μόλις γίνει η επεξεργασία όλων των στοιχείων ελέγχου επικύρωσης, η ίδια η σελίδα ορίζει μια ιδιότητα με βάση τις τιμές των ιδιοτήτων επικύρωσης των στοιχείων ελέγχου. Αν κάποιος από τα στοιχεία ελέγχου δεν περάσει την επικύρωση, η ιδιότητα της σελίδας δίνει αυτόματα την τιμή False στην ιδιότητα επικύρωσης. Όταν δημιουργείτε τον κώδικα επικύρωσης, η ιδιότητα της σελίδας θα είναι μια σημαντική ιδιότητα που πρέπει να ελέγχετε για να είστε σίγουροι ότι όλα είναι σωστά. Αφού ελεγχθεί η ιδιότητα της σελίδας και δείτε ότι η επικύρωση απέτυχε, η σελίδα συνήθως επιστέφει στο χρήστη και θα εμφανιστεί κάποιο μήνυμα σφάλματος, αν έχει ορισθεί ένα τέτοιο μήνυμα στα στοιχεία ελέγχου που απέτυχαν να περάσουν την επικύρωση.

Το ASP.NET έχει έξι είδη στοιχείων ελέγχου επικύρωσης: το RequiredFieldValidator, RegularExpressionValidator, CompareValidator, RangeValidator, CustomValidator και το ValidationSummary. Όλα αυτά τα

στοιχεία ελέγχου έχουν ένα κοινό σύνολο από ιδιότητες και μεθόδους, οι περισσότερες από τις οποίες, κληρονομούνται από την κλάση `BaseValidator` και από την κλάση `WebControl` που βρίσκονται στο χώρο ονομάτων `System.UI.WebControls` και την κλάση `Control` που μπορεί να βρεθεί στο χώρο ονομάτων `System.Web.UI`. Μια εξαίρεση σε αυτόν τον κανόνα είναι η ιδιότητα `Text`, η οποία κληρονομείται από την κλάση `Label`.

4.1.1 RequiredFieldValidator

Το στοιχείο ελέγχου `RequiredFieldValidator` ελέγχει αν ο χρήστης έδωσε δεδομένα ή αν επέλεξε μια τιμή στο στοιχείο ελέγχου εισόδου. Μόλις γίνει η αντιστοίχιση με ένα στοιχείο ελέγχου εισόδου, αυτό το στοιχείο ελέγχου επικύρωσης διασφαλίζει ότι εισάγεται μια τιμή στο πεδίο του στοιχείου ελέγχου. Μαζί με τις κοινές ιδιότητες επικύρωσης των στοιχείων ελέγχου, το `RequiredFieldValidator` έχει μια μοναδική ιδιότητα: την ιδιότητα `InitialValue`. Αυτή καθορίζει μια αρχική τιμή για το στοιχείο ελέγχου επικύρωσης. Αν η τιμή στο στοιχείο ελέγχου είναι ίση με την τιμή της ιδιότητας `InitialValue`, η επικύρωση αποτυγχάνει.

Η σύνταξη για αυτό το στοιχείο ελέγχου είναι η παρακάτω:

```
<asp:RequiredFieldValidator runat="server" Display="dynamic"
ID="valRequireName" SetFocusOnError="true"
ControlToValidate="txtName" ErrorMessage="Το όνομα
σας είναι απαραίτητο">*</asp:RequiredFieldValidator>
```

Αυτό το παράδειγμα δημιουργεί ένα στοιχείο ελέγχου `RequiredFieldValidator` με όνομα `valRequireName` που επικυρώνει το στοιχείο ελέγχου `txtName`. Αν δεν υπάρχει τιμή στο στοιχείο ελέγχου θα εμφανιστεί το μήνυμα σφάλματος «`Το όνομα σας είναι απαραίτητο.`». Παρατηρούμε ότι λείπει η ιδιότητα `InitialValue` που από default έχει τιμή το κενό. Επίσης υπάρχει η ιδιότητα `EnableClientScript="False"` η οποία ορίζει ότι δεν είναι ενεργοποιημένη η επικύρωση από την πλευρά του πελάτη.

4.1.2 RegularExpressionValidator

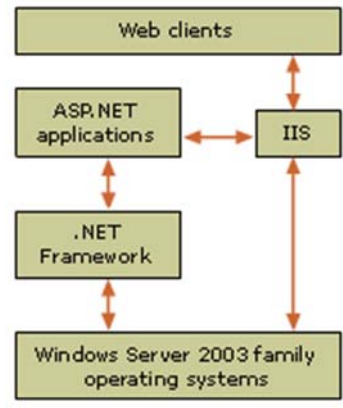
Το στοιχείο ελέγχου `RegularExpressionValidator` ελέγχει αν συμπίπτει ένα στοιχείο με ένα μοτίβο που καθορίζεται από μια κανονική παράσταση (regular expression). Μια *κανονική παράσταση* είναι ένα μοτίβο κειμένου, το οποίο αποτελείται από συνηθισμένους και ειδικούς χαρακτήρες, που είναι γνωστοί (*μετα-χαρακτήρες* ή *metacharacter*). Το μοτίβο χρησιμεύει ως πρότυπο, καθώς περιγράφει μία ή περισσότερες συμβολοσειρές τις οποίες πρέπει να βρει όταν κάνει αναζήτηση σε ένα κείμενο.

Αυτό το είδος της επικύρωσης σας επιτρέπει να ψάχνετε για προβλεπόμενες ακολουθίες χαρακτήρων, όπως εκείνες που χρησιμοποιούνται για τις διευθύνσεις ηλεκτρονικού ταχυδρομείου, για αριθμούς τηλεφώνου κτλ. Υπάρχει μια μοναδική ιδιότητα που σχετίζεται με τα στοιχεία ελέγχου `RegularExpressionValidator`: η ιδιότητα `ValidationExpression`. Προσδιορίζει την κανονική παράσταση που ορίζει τα κριτήρια επικύρωσης.

```
<asp:RegularExpressionValidator runat="server" Display="dynamic"
ID="valEmailPattern" SetFocusOnError="true"
ControlToValidate="txtEmail"
ValidationExpression="\w+([-+.']\w+)*@\w+([-.\w+)*\.\w+([-.\w+)*"
ErrorMessage="The e-mail address you specified is not well-
formed">*</asp:RegularExpressionValidator>
```

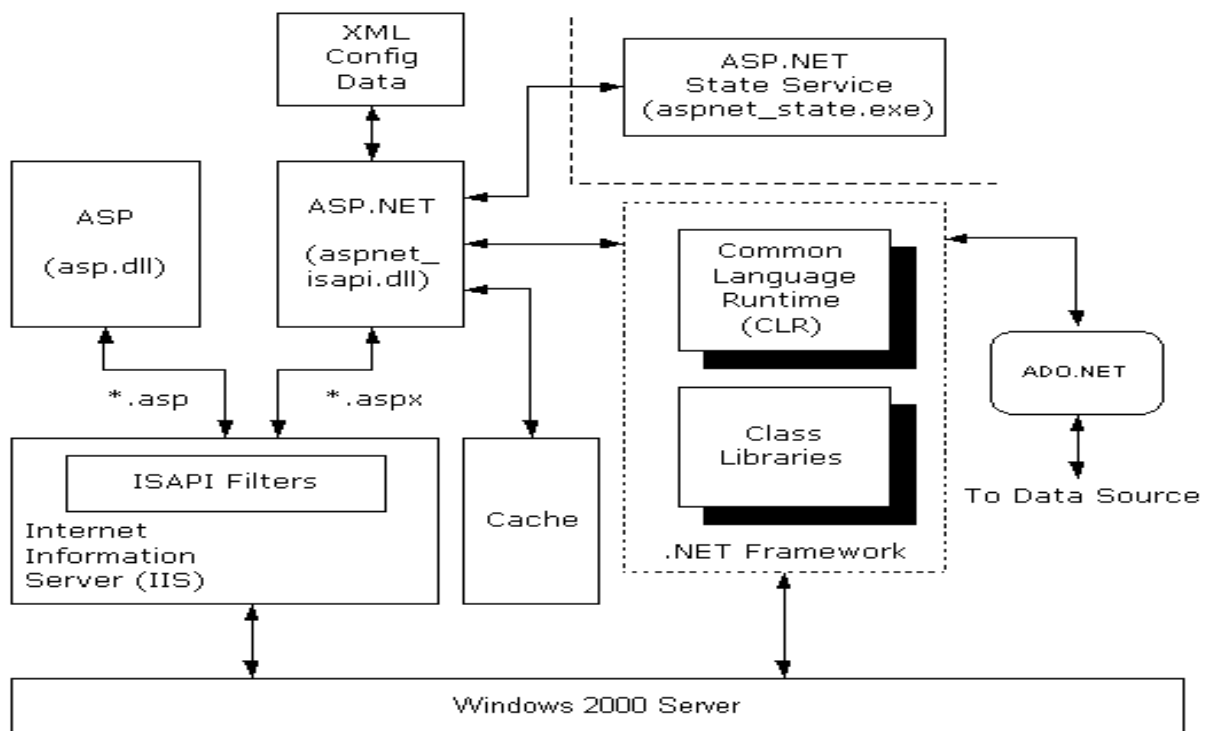
Αυτό το παράδειγμα δημιουργεί ένα στοιχείο ελέγχου `RegularExpressionValidator` με όνομα `valEmailPattern` που επικυρώνει το στοιχείο ελέγχου `txtEmail`. Όταν εισάγεται μια τιμή συγκρίνεται με την ιδιότητα `ValidationExpression` που στην συγκεκριμένη περίπτωση επιβεβαιώνει την σωστή εισαγωγή ενός mail. Επίσης υπάρχει η ιδιότητα `SetFocusOnError="true"` η οποία ορίζει ότι είναι ενεργοποιημένη η επικύρωση.

4.1.3 ASP.NET APXITEKTONIKH



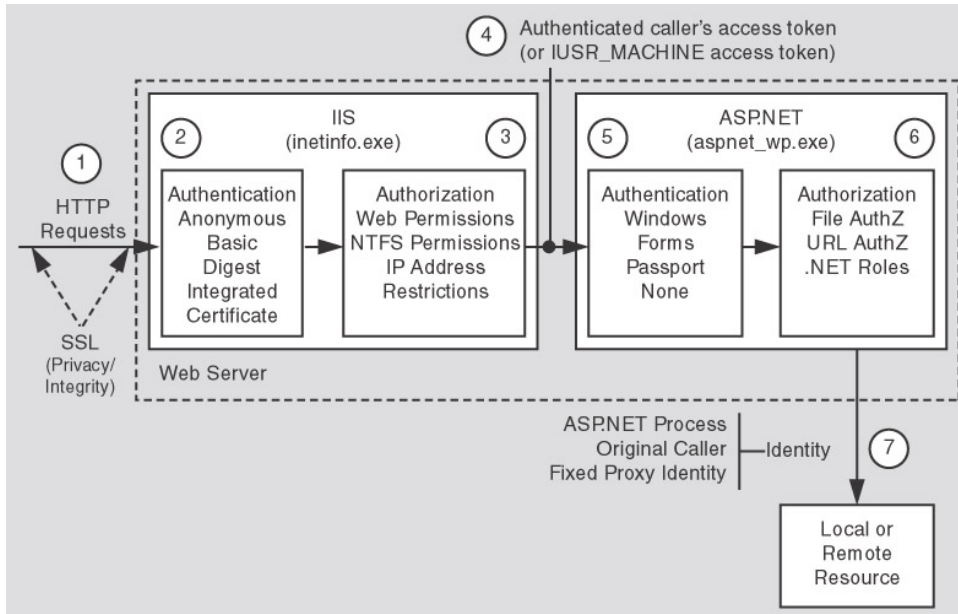
Όπως δείχνει το παράδειγμα, όλοι οι πελάτες επικοινωνούν με Ιστού ASP.NET εφαρμογές μέσω του Microsoft Internet Information Services (IIS). IIS πιστοποιεί την αίτηση, εφόσον απαιτείται, και στη συνέχεια τοποθετεί την πηγή που ζητήθηκε (όπως μια εφαρμογή ASP.NET). Εάν ο πελάτης έχει το δικαίωμα, ο πόρος που είναι διαθέσιμες.

ASP.NET Architecture



4.2 Ασφάλεια

Η αρχιτεκτονική ασφαλείας του ASP.NET αποτελείται από δύο μέρη. Το πρώτο στοιχείο είναι η ενσωμάτωσή της στο IIS. Το δεύτερο στοιχείο βρίσκεται στο βασικό κατάλογο της εφαρμογής ASP.NET, σε ένα αρχείο που ονομάζεται Web.config. Αυτό το αρχείο περιέχει διάφορες ρυθμίσεις διαμόρφωσης, όπως ρυθμίσεις ασφαλείας, για την εφαρμογή μας.



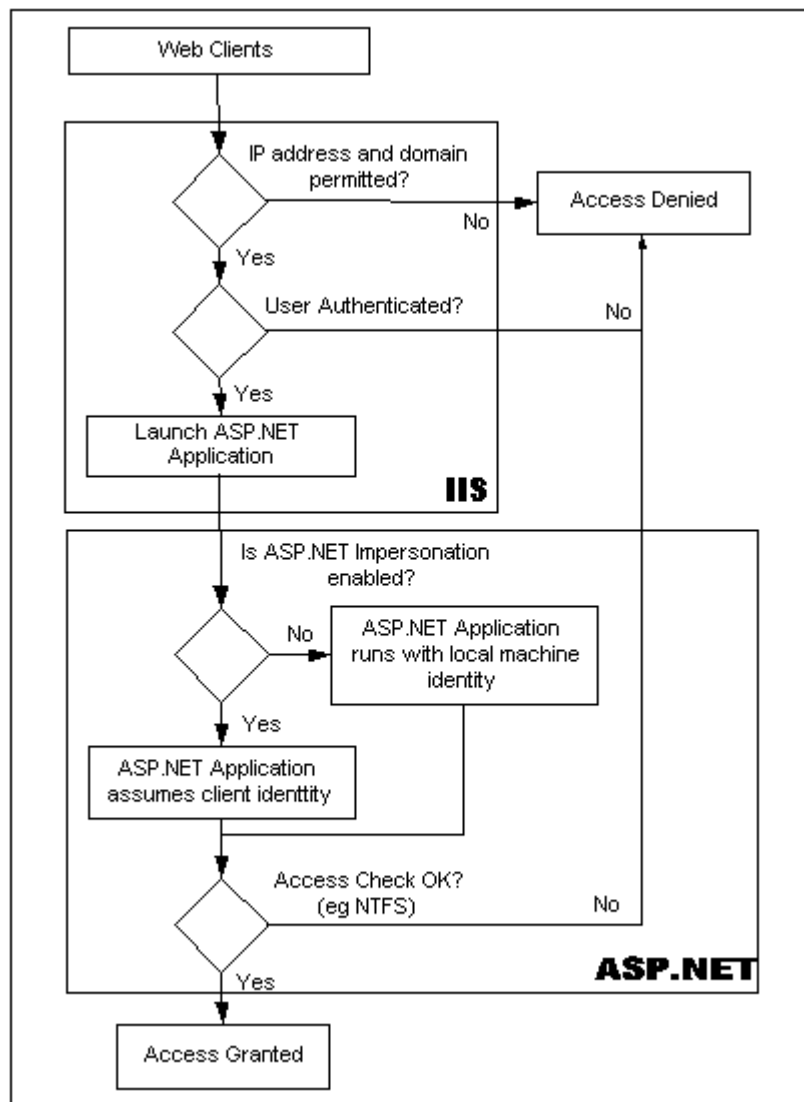
Σχήμα: Η αρχιτεκτονική ασφαλείας του ASP.NET

ASP.NET Configuration File Security Settings

Θα πρέπει να κατανοήσουμε τη σχέση μεταξύ των υπηρεσιών Internet Information Services (IIS) πιστοποίησης της ταυτότητας και του Microsoft® ASP.NET αρχιτεκτονική ασφαλείας κατά το σχεδιασμό της αίτησής σας. Αυτό θα επιτρέψει σε σας για τον έλεγχο ταυτότητας των χρηστών σας κατάλληλα και να λάβει το κατάλληλο πλαίσιο ασφαλείας εντός της αίτησής σας. Θα πρέπει να σημειωθεί ότι η εφαρμογή ASP.NET διαμόρφωση της ασφαλείας και IIS διαμόρφωση ασφαλείας είναι εντελώς ανεξάρτητη και μπορεί να χρησιμοποιηθεί ανεξάρτητα ή σε συνδυασμό μεταξύ τους.

Η IIS διατηρεί σχέση με την ασφάλεια των ρυθμίσεων διαμόρφωσης στη μετα-βάση IIS. Ωστόσο, ASP.NET υποστηρίζει την ασφάλεια (και άλλες) ρυθμίσεις σε XML αρχεία ρυθμίσεων. Αν και αυτό απλοποιεί γενικά την ανάπτυξη της εφαρμογής σας από την άποψη της ασφαλείας, το μοντέλο ασφαλείας που εγκρίθηκε από την αίτησή σας θα απαιτήσει τη σωστή

διαμόρφωση των τόσο τη μεταβάση IIS και ASP.NET εφαρμογής σας μέσω του αρχείου ρυθμίσεων του (Web.config).



Σχήμα: Η σχέση μεταξύ της ασφάλειας των υπηρεσιών IIS και του ASP.NET

Η ASP.NET πιστοποίηση και IIS Security

Η ASP.NET εκτελεί έλεγχο ταυτότητας με παρόχους ταυτότητας, τα οποία είναι ενότητες κώδικα ότι στον έλεγχο των εντολών και να εφαρμόσει άλλες λειτουργίες ασφάλειας, όπως η παραγωγή cookies. Η ASP.NET υποστηρίζει τις ακόλουθες τρεις παρόχους ταυτότητας:

- **Forms Authentication . Έντυπα Authentication.** Η χρήση αυτής της παροχής προκαλεί χωρίς έλεγχο αιτήματα να κατευθυνθούν προς μια συγκεκριμένη μορφή HTML χρησιμοποιώντας ανακατεύθυνση πλευρά του πελάτη. Ο χρήστης μπορεί να παρέχει πιστοποιήσεις σύνδεσης, στη συνέχεια, και μετά τη μορφή πίσω στον server. Εάν η αίτηση πιστοποιεί την αίτηση (με συγκεκριμένες εφαρμογές λογική), τα θέματα ASP.NET ένα cookie που περιέχει τα διαπιστευτήρια ή ένα κλειδί για τον επαναπροσδιορισμό της ταυτότητας του πελάτη. Οι επόμενες αιτήσεις που εκδίδονται με το cookie στο αίτημα, το οποίο σημαίνει ότι οι επόμενες πιστοποιήσεις είναι περιττές.
- **Passport Authentication . Authentication διαβατηρίου.** Αυτή είναι μια κεντρική υπηρεσία ελέγχου ταυτότητας που προβλέπεται από τη Microsoft που προσφέρει μια ενιαία δυνατότητα σύνδεσης και την ένταξη των υπηρεσιών για τις συμμετέχουσες τοποθεσίες. Η ASP.NET, σε συνδυασμό με το διαβατήριο του Microsoft ® Software Development Kit (SDK), παρέχει παρόμοιες λειτουργίες όπως Forms Authentication στους χρήστες διαβατηρίου.
- **Windows Authentication . Windows Authentication.** Ο φορέας αυτός, χρησιμοποιεί τις δυνατότητες ελέγχου ταυτότητας των IIS. Μετά IIS ολοκληρώνει ταυτότητας του, ASP.NET χρησιμοποιεί την επαλήθευση της ταυτότητας του για να επιτρέψει την πρόσβαση.

Για να καταστεί δυνατή συγκεκριμένο πάροχο ταυτότητας για μια εφαρμογή ASP.NET, πρέπει να δημιουργήσετε μια καταχώρηση στο αρχείο ρυθμίσεων της εφαρμογής ως εξής:

```
// web.config file // Αρχείο web.config
<authentication mode = "[Windows/Forms/Passport/None]"> mode
<authentication = "[Windows/Forms/Passport/None]">
</authentication> </ authentication>
```

Το ακόλουθο παράδειγμα δείχνει τη σύνταξη του αρχείου διαμόρφωσης:

```
<authentication mode="Forms">
  <forms cookieless="UseCookies" loginUrl="~/Login.aspx"
name="BOKFORMAUTH"/>
</authentication>

<membership defaultProvider="AspNetMembershipProvider">
  <providers>
    <add connectionString="BoKConnectionString"
enablePasswordRetrieval="false" enablePasswordReset="true"
requiresQuestionAndAnswer="true" applicationName="/"
requiresUniqueEmail="true" passwordFormat="Hashed"
maxInvalidPasswordAttempts="5" minRequiredPasswordLength="4"
minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="20"
passwordStrengthRegularExpression="" name="AspNetMembershipProvider"
type="System.Web.Security.SqlMembershipProvider, System.Web,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
  </providers>
```

</membership>

Η ρύθμιση ASP.NET παρέχει λειτουργικότητα για τη ρύθμιση ενός ολόκληρου διακομιστή, μια εφαρμογή ASP.NET, ή μεμονωμένες σελίδες στο υποκαταλόγους εφαρμογή.

Μπορείτε να ρυθμίσετε τα χαρακτηριστικά, όπως τα μέσα ελέγχου ταυτότητας, προσωρινή αποθήκευση σελίδα, επιλογές μεταγλώττισης, σφάλματα, debug και πολλά άλλα.

Λαμβάνοντας υπόψη ότι μετά την κωδικοποίηση και τη διαμόρφωση των βέλτιστων πρακτικών μπορούν να συμβάλουν στη βελτίωση της ασφάλειας της αίτησής σας, είναι σημαντικό να κρατάτε συνεχώς διακομιστή αίτηση σας ενημερωμένο με τις τελευταίες ενημερώσεις ασφαλείας για τα Microsoft Windows και Microsoft Internet Information Services (IIS), καθώς και κάθε ενημερωμένες εκδόσεις για τον Microsoft SQL Server ή άλλων πηγών δεδομένων μέλους.

4.2.1 Ασφάλεια του IIS (Internet Information Server)

Το πρώτο βήμα για την ασφάλεια της εφαρμογής μας ξεκινά με το Web διακομιστή οποίος φιλοξενεί την τοποθεσία μας. Στις περισσότερες περιπτώσεις, αυτός ο Web διακομιστής είναι ο Internet Information Server 5.0+, ο οποίος εγκαθίσταται αυτόματα με τα Windows 2000+ Server και μπορεί να προστεθεί με μη αυτόματο τρόπο στα Windows 2000+ Professional από τα Windows Components. Το IIS μας δίνει αρκετές μεθόδους για να ασφαλίσουμε τη Web τοποθεσία μας. Αυτές οι επιλογές βρίσκονται στην καρτέλα Directory Security (ασφάλεια καταλόγου) στο παράθυρο Properties (ιδιότητες) της τοποθεσίας μας.

Δημιουργία ενός αρχείου Web.config

Μπορείτε να δημιουργήσετε ένα αρχείο Web.config, χρησιμοποιώντας ένα πρόγραμμα επεξεργασίας κειμένου όπως το Notepad. Πρέπει να

δημιουργήσετε ένα αρχείο κειμένου με το όνομα **Web.config** στο ριζικό κατάλογο της εφαρμογής ASP.NET. Το αρχείο Web.config πρέπει να είναι ένα καλά μορφοποιημένο έγγραφο XML και πρέπει να έχει μορφή παρόμοια με τον% SystemRoot% \ Microsoft.NET \ Framework \ % VersionNumber% \ CONFIG \ αρχείο Machine.config.

Το αρχείο Web.config πρέπει να περιέχει μόνο καταχωρήσεις για στοιχεία ρύθμισης που παρακάμπτουν τις ρυθμίσεις του αρχείου Machine.config. Τουλάχιστον, το αρχείο Web.config πρέπει να έχουν το στοιχείο **<configuration>** και την **<system.web>** στοιχείο. Τα στοιχεία αυτά θα περιέχουν μεμονωμένα στοιχεία ρύθμισης.

Το ακόλουθο παράδειγμα δείχνει ένα ελάχιστο αρχείο Web.config:

```
<? xml version = "1.0" encoding = "utf-8"?>
<configuration>
  <system.web>

  </ system.web>
</ configuration>
```

Η πρώτη γραμμή του αρχείου Web.config περιγράφει το έγγραφο σε μορφή XML και καθορίζει τον τύπο κωδικοποίησης χαρακτήρων. Αυτή η πρώτη γραμμή πρέπει να είναι η ίδια για όλους. Config αρχεία.

Οι γραμμές που ακολουθούν σηματοδοτούν την αρχή και το τέλος της **<configuration>** στοιχείο και το στοιχείο της **<system.web>** αρχείο Web.config. Με τον εαυτό τους, οι γραμμές αυτές δεν κάνουν τίποτα. Ωστόσο, οι γραμμές παρέχουν μια δομή που σας επιτρέπει να προσθέσετε μελλοντικές ρυθμίσεις. Μπορείτε να προσθέσετε την πλειοψηφία του ASP.NET ρυθμίσεις μεταξύ των <system.web> και </ system.web> γραμμές. Οι γραμμές αυτές σηματοδοτούν την αρχή και το τέλος του ASP.NET ρυθμίσεις.

4.2.2 Το αρχείο Web.config

Το web.config που είναι αποθηκευμένο σε ένα xml αρχείο στο κατάλογο root περιγράφει διάφορες ρυθμίσεις διαμόρφωσης. Υπάρχει ένας κόμβος σε

αυτό το αρχείο που ονομάζεται <security> και περιέχει πληροφορίες διαμόρφωσης ασφάλειας για την ASP.NET εφαρμογή μας. Ακόμα σε αυτό το αρχείο υπάρχουν και διάφορες άλλες ιδιότητες που ανάλογα το πως θα ορισθούν επιτρέπουν ή απαγορεύουν σε διάφορους χρήστες την πρόσβαση στον πόρο που ζητούν.

Στο αρχείο Web.config εισάγαμε το connection string (απαραίτητο για την σύνδεση με τη βάση) έτσι ώστε να χρησιμοποιείται από όλες τις φόρμες με ευκολία και να μην φαίνεται στον HTML κώδικα. Ο κώδικας που προσθέτουμε στο Web.config για να ορίσουμε το connection string είναι ο παρακάτω:

```
<appSettings/>
  <connectionStrings>

    <add name="BoKConnectionString" connectionString="Data Source=VAS;Initial
Catalog=BokEshop;Integrated Security=True" providerName="System.Data.SqlClient"/>

  </connectionStrings>
```

Για να προστατέψουμε κάποια τμήματα της σελίδας απο χρήστες οι οποίοι δεν είναι εξουσιοδοτημένοι να τα δουν, προσθέτουμε στο web.config.

```
<location path="Admin">
  <system.web>
    <authorization>
      <allow roles="Administrators" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

4.3 Cookies

Ένας δημοφιλής τρόπος προσαρμογής των αλληλεπιδράσεων με ιστοσελίδες είναι μέσω των cookies. Ένα cookie είναι ένα αρχείο κειμένου, αποθηκευμένο από ένα δικτυακό τόπο στον υπολογιστή κάποιου, που επιτρέπει στον τόπο να παρακολουθεί τις ενέργειες του επισκέπτη. Την πρώτη φορά που ο χρήστης επισκέπτεται τον δικτυακό τόπο, ο υπολογιστής του χρήστη μπορεί να δεχθεί ένα cookie. Αυτό το cookie ενεργοποιείται στην συνέχεια κάθε φορά που ο χρήστης επισκέπτεται ξανά αυτόν τον τόπο. Οι συλλεγόμενες πληροφορίες προορίζονται να είναι μια ανώνυμη εγγραφή που περιέχει δεδομένα τα οποία χρησιμοποιούνται για την προσωποποίηση των μελλοντικών επισκέψεων του χρήστη στην τοποθεσία. Για παράδειγμα, τα cookies σε μια εφαρμογή αγορών (όπως είναι και το E-Shop) μπορεί να

αποθηκεύουν μοναδικά προσδιοριστικά για τους χρήστες. Όταν ένας χρήστης προσθέτει τεμάχια σε μια online αγορά ή εκτελεί μια άλλη εργασία που προκύπτει από μια αίτηση στον Web server, ο server λαμβάνει ένα cookie που περιέχει τα μοναδικά προσδιοριστικά του χρήστη. Ο server χρησιμοποιεί κατόπιν τα μοναδικά προσδιοριστικά για να εντοπίσει το «καλάθι» αγορών και εκτελεί οποιαδήποτε αναγκαία επεξεργασία.

Εκτός από τον προσδιορισμό των χρηστών, τα cookies μπορούν επίσης να δηλώνουν τις προτιμήσεις αγορών των clients. Όταν μια Web Form λαμβάνει μια αίτηση από έναν client, η Web Form θα μπορούσε να εξετάζει τα cookie(s) που έστειλε στον client σε προηγούμενες επικοινωνίες, να προσδιορίζει τις προτιμήσεις του client και αμέσως να προβάλλει προϊόντα που ενδιαφέρουν τον client.

Κάθε αλληλεπίδραση βασισμένη στο HTTP ανάμεσα σε έναν client και έναν server, περιλαμβάνει μια επικεφαλίδα που περιέχει πληροφορίες είτε για την αίτηση (όταν η επικοινωνία ξεκινά από τον client προς τον server), είτε για την απόκριση (όταν η επικοινωνία ξεκινά από τον server προς τον client.) Όταν μια Web Form λαμβάνει μια αίτηση, η επικεφαλίδα περιλαμβάνει πληροφορίες όπως ο τύπος αίτησης (π.χ Get) και οποιαδήποτε cookies που έχουν προηγουμένως αποσταλεί από τον server για να αποθηκευθούν στο σύστημα του client. Όταν ο server διατυπώσει την απάντηση του, οι πληροφορίες επικεφαλίδας περιλαμβάνουν οποιαδήποτε cookies θέλει ο server να αποθηκεύσει στον υπολογιστή του client και άλλες πληροφορίες, όπως ο τύπος MIME της απάντησης.

Εάν ο προγραμματιστής ενός cookie δεν θέσει μια ημερομηνία εκπνοής, το πρόγραμμα περιήγησης Web διατηρεί το cookie για την διάρκεια της συνεδρίας περιήγησης. Διαφορετικά, το πρόγραμμα περιήγησης Web διατηρεί το cookie μέχρι την ημερομηνία εκπνοής του. Όταν το πρόγραμμα περιήγησης ζητήσει έναν πόρο από έναν Web server, τα cookies που είχαν προηγουμένως αποσταλεί στον client από αυτόν τον Web server επιστρέφουν στον Web server ως μέρος της αίτησης που διαμορφώθηκε από το πρόγραμμα περιήγησης. Τα cookies διαγράφονται όταν εκπνεύσουν. Η ημερομηνία εκπνοής ενός cookie μπορεί να τεθεί στην ιδιότητα **Expires** αυτού του cookie.

4.4 Σύνδεση με Βάση Δεδομένων

Μέσα στον κώδικα έχουμε πρόσβαση στην βάση δημιουργώντας μία σύνδεση:

```
Dim con As New
SqlConnection(ConfigurationManager.ConnectionStrings("BoKConnectionString").Connection
String)
Dim da2 As New SqlDataAdapter(" SELECT dbo.Products.*,
dbo.SubCategories.SubCat_Value FROM dbo.SubCategories RIGHT OUTER JOIN " _
& " dbo.Products ON dbo.SubCategories.SubCat_ID =
dbo.Products.Products_SubCatID ", con)
Dim dt As New DataTable()
da2.Fill(dt)

gvwProducts.DataSource = dt
```

Στον παραπάνω κώδικα δηλώνουμε ένα SqlConnection του οποίου το connectionstring το διαβάζουμε από το αρχείο web.config.

Παράδειγμα1 : Διαδραφή προϊόντος από τον πίνακα

```
Dim con As New
SqlConnection(ConfigurationManager.ConnectionStrings("BoKConnectionString").Connection
String)
Dim cmd As New SqlCommand()
cmd.CommandText = "DELETE FROM Products WHERE Products_ProductID = " &
Convert.ToInt16(gvwProducts.Rows(e.RowIndex).Cells(0).Text)
cmd.Connection = con
con.Open()
cmd.ExecuteNonQuery()
con.Close()
```

Παράδειγμα2 : Εισαγωγή παραγγελίας στον πίνακα

```
Try
conn.ConnectionString =
ConfigurationManager.ConnectionStrings("BoKConnectionString").ConnectionString
conn.Open()
Cmd.CommandText = "INSERT INTO Orders(Orders_UserId, Orders_OrderDate,
Orders_ProductId, Orders_Quantity, Orders_TotalCost, Orders_Status,
Orders_JustInserted, Orders_PaymentTypeId) " _
& "Values (@Orders_UserId, @Orders_OrderDate, @Orders_ProductId,
@Orders_Quantity, @Orders_TotalCost, @Orders_Status, @Orders_JustInserted,
@Orders_PaymentTypeId) "

Dim User As Guid = New Guid(CStr(myObject.ProviderUserKey.ToString))
Cmd.Parameters.Add("@Orders_UserId", SqlDbType.UniqueIdentifier).Value = User
Cmd.Parameters.Add("@Orders_OrderDate", SqlDbType.DateTime).Value = Now.Date
Cmd.Parameters.Add("@Orders_ProductId", SqlDbType.Int).Value = productId
Cmd.Parameters.Add("@Orders_Quantity", SqlDbType.Int).Value = quantity
Cmd.Parameters.Add("@Orders_TotalCost", SqlDbType.Money).Value =
ShoppingCart_FullPrice
Cmd.Parameters.Add("@Orders_Status", SqlDbType.Bit).Value = False

Cmd.Parameters.Add("@Orders_JustInserted", SqlDbType.Bit).Value = False

Cmd.Parameters.Add("@Orders_PaymentTypeId", SqlDbType.Int).Value =
ddl_PayType.SelectedValue

Session.Clear()
Cmd.Connection = conn
```

```
Cmd.ExecuteNonQuery()

Catch ee As Exception
    Label1.Visible = True

    Label1.Text = ee.Message.ToString()

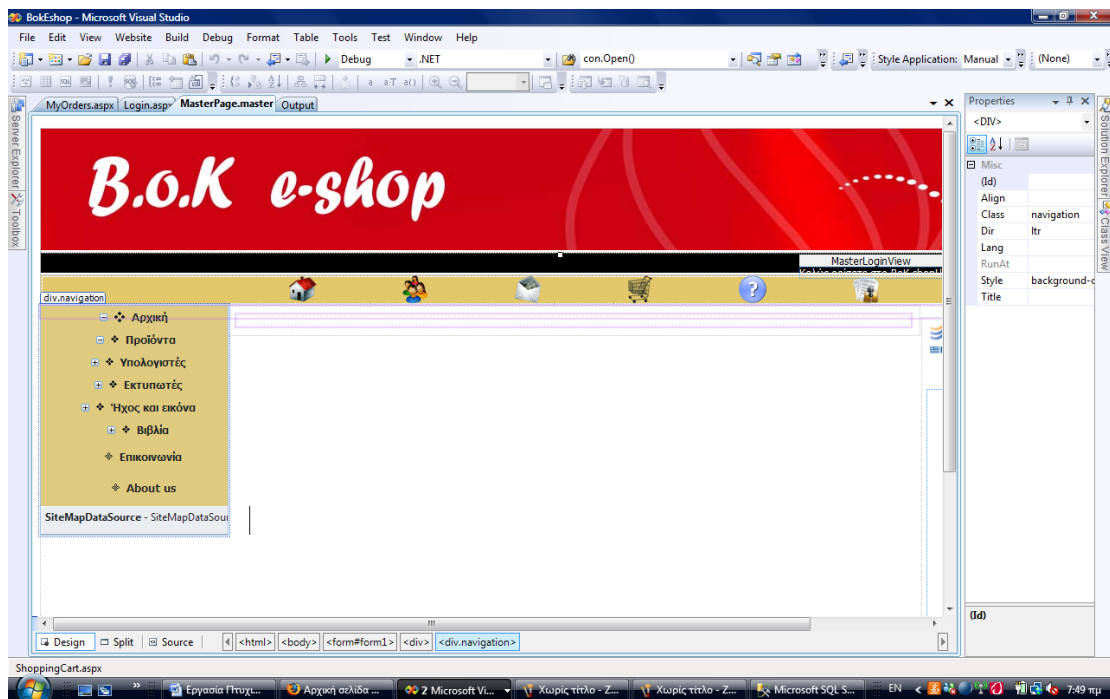
Finally

    ds1.Dispose()
    dal.Dispose()
    conn.Close()
    conn.Dispose()
```

5. Κεφάλαιο - Ανάλυση της εφαρμογής μας

5.1 Οι σελίδες αναλυτικά

- Η φόρμα MasterPage.aspx



Σχήμα MasterPage.aspx

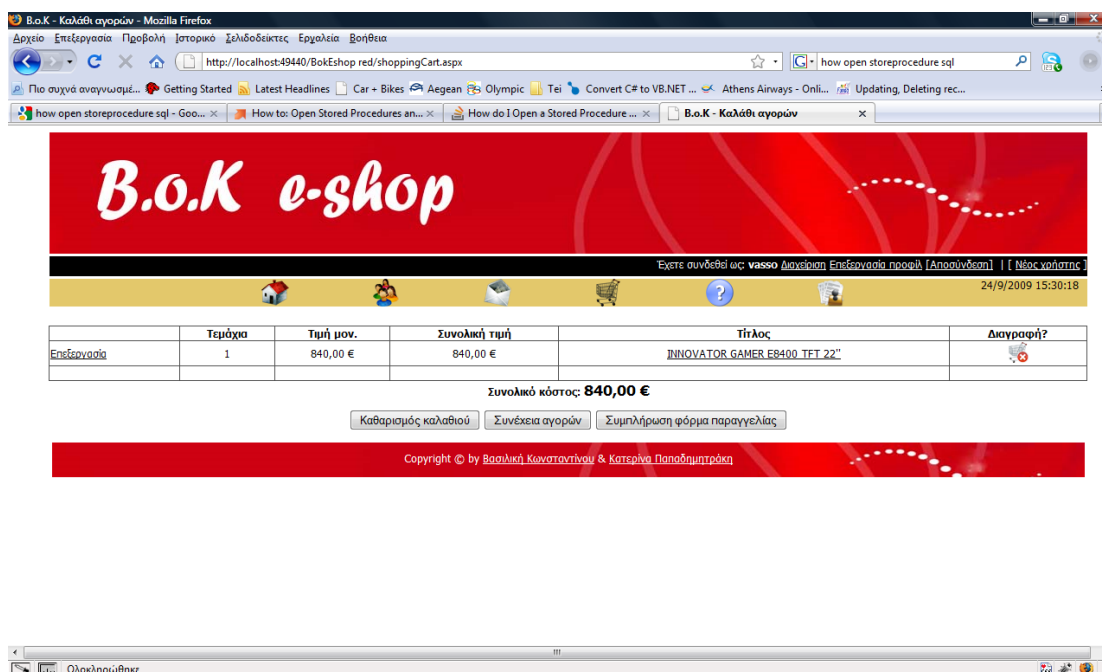
Η παραπάνω φόρμα (Σχήμα 5.1) είναι η πρώτη φόρμα της web εφαρμογής μας, όπου ο πελάτης έχει την δυνατότητα να πλοηγηθεί στις άλλες φόρμες του ηλεκτρονικού καταστήματος BoK e-shop. Σε αυτή τη φόρμα απεικονίζονται κάποια από τα προϊόντα που με τη χρήση του menu που είναι σε μορφή tree view, έχει την δυνατότητα να επιλέξει μια από τις διαθέσιμες υπηρεσίες που προσφέρει το ηλεκτρονικό κατάστημα. Το menu που είναι σε μορφή tree view, είναι τοποθετημένο στα αριστερά της οθόνης και εκεί παρουσιάζονται όλα τα προϊόντα που διαθέτει το κατάστημα.

Πιο συγκεκριμένα ο πελάτης μπορεί να δει ποια και πόσα προϊόντα έχει μέσα στο καλάθι αγορών του, κάνοντας κλικ στο αντίστοιχο κουμπί στο οριζόντιο menu, στο εικονίδιο με το καλάθι. Άλλη μια υπηρεσία που παρέχεται στον πελάτη, είναι η δημιουργία ενός λογαριασμού (μέσω της σελίδας Register.aspx), με τον οποίο διασφαλίζεται κάθε συναλλαγή του πελάτη προς το κατάστημα και αντιστρόφως. Ο πελάτης έχοντας στην διάθεση του ένα μοναδικό λογαριασμό (ο οποίος αποτελείται από ένα username και ένα password) μπορεί να προβεί στην αγορά οποιουδήποτε προϊόντος που επιθυμεί εκείνος. Ο πελάτης μπορεί να έρθει σε

επικοινωνία με το κατάστημα στέλνοντας ένα e-mail μέσω της φόρμας ContactUs.aspx, η οποία εμφανίζεται κάνοντας κλικ στο κουμπί Επικοινωνία.

Η δημιουργία αυτής της φόρμας έγινε με το CSS και κωδικα HTML. Στην σχεδίαση της χρησιμοποιήσαμε το Photoshop. Οι υπόλοιπες φόρμες της Web εφαρμογής μας κληρονομούν την MasterPage.Master , το μόνο που διαφοροποιείται είναι το περιεχόμενο , δηλαδή το Content Place Holder της κάθε φόρμας.

- Η φόρμα **ShoppingCart.aspx**



Σχήμα ShoppingCart.aspx

Η φόρμα **ShoppingCart** είναι η βασικότερη φόρμα της εφαρμογής, διότι πάνω σε αυτήν στηρίζεται η σωστή λειτουργία του καταστήματος και η ασφάλεια των συναλλαγών. Ο πελάτης σε αυτήν την φόρμα μέσω ενός GridView, το οποίο περιέχει πληροφορίες όπως κωδικός προϊόντος, μοντέλο, τιμή μονάδας, ποσότητα και γενικό σύνολο, βλέπει ποια και πόσα προϊόντα έχει τοποθετήσει μέσα στο «καλάθι» αγορών του. Σε περίπτωση όπου ο πελάτης για οποιοδήποτε λόγο δεν επιθυμεί να συμπεριλάβει ένα συγκεκριμένο προϊόν στο «καλάθι» του, τότε μπορεί πολύ απλά να το διαγράψει κάνοντας κλικ στο κουμπί Διαγραφή. Κάνοντας κλικ στο κουμπί «καθαρισμός καλαθιού» ο πελάτης αδειάζει το «καλάθι» του, αφαιρώντας από αυτό, όλα τα προϊόντα τα οποία περιελάμβανε, κάνοντάς κλικ στο «Συνέχεια αγορών» στέλνεται στην αρχική σελίδα για συνέχιση των αγορών και με το κουμπί

«Συμπλήρωση φόρμα παραγγελίας» στέλνεται στην σελίδα submitOrder.aspx όπου μπορεί να εισάγει τα στοιχεία στα οποία θα γίνει η αποστολή και να επιλέξει τον τρόπο πληρωμής.

- **Η φόρμα SubmitOrder.aspx**

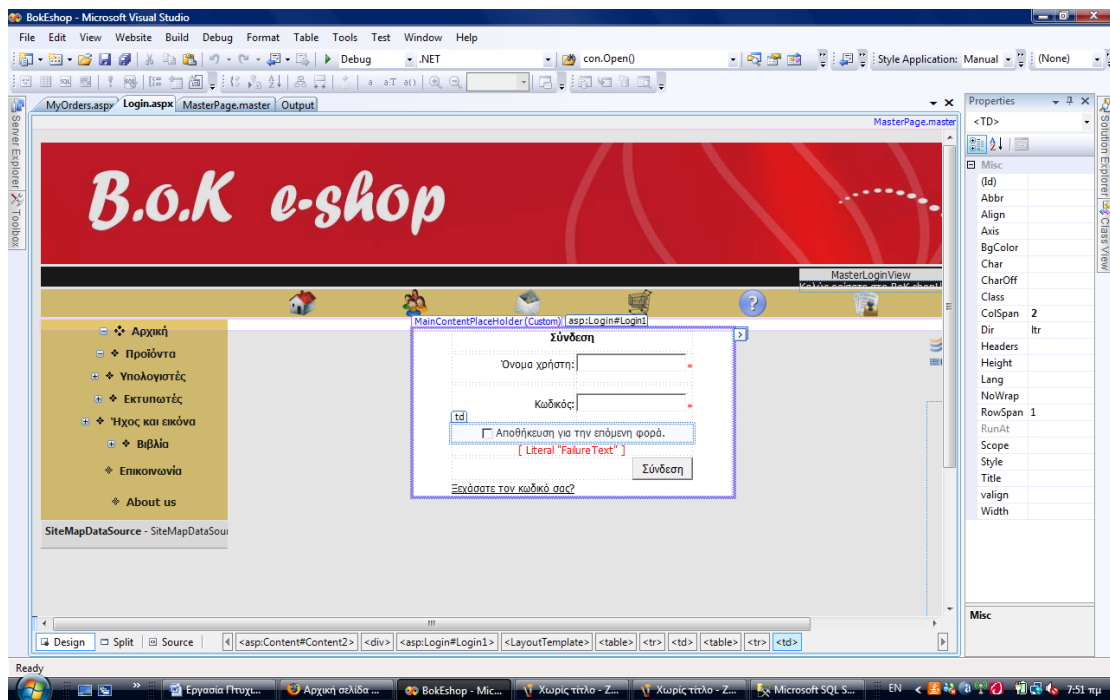
The screenshot shows the B.O.K e-shop website interface. At the top, there is a red banner with the logo "B.O.K e-shop". Below the banner, a navigation menu is visible on the left side, listing categories like "Αρχική", "Προϊόντα", "Υπολογιστές", "Εκτυπωτές", "Ήχος και εικόνα", "Βιβλία", "Επικοινωνία", and "About us". The main content area displays a form for submitting an order. The form includes the following fields and elements:

- Header: "Είσαγεται τα στοιχεία σας, ώστε να γίνει η αποστολή των προϊόντων που επιθυμείτε"
- Fields: "Όνομα:" (vasso), "Επώνυμο:" (konstantinou)
- Text: "Συμπληρώστε τα στοιχεία σας για να γίνει η αποστολή των προϊόντων."
- Section: "Στοιχεία αποστολής"
- Fields: "Οδός / Αριθμός:" (askidh 12), "Τ.Κ.:" (71305), "Πόλη:" (Προσωπικά στοιχεία), "Τηλ.:" (12313212312), "Κινητό τηλ.:" (asdasd), "Fax:" (asdasdasd)
- Section: "Επιλέξτε τον τρόπο πληρωμής"
- Dropdown: "Με αντικαταβολή"
- Buttons: "Ολοκλήρωση παραγγελίας", "Ακύρωση παραγγελίας"
- Footer: "Copyright © by Βασιλική Κωνσταντίνου & Κατερίνα Παπαδημητράκη"

Σχήμα SubmitOrder.aspx

Η φόρμα SubmitOrder.aspx μαζί με την ShoppingCart.aspx αποτελούν τις βασικότερες φόρμες της εφαρμογής και ο λόγος είναι ότι αποτελούν τον πυρήνα για την ορθή λειτουργία του καταστήματος και την ασφάλεια των συναλλαγών και από τις δυο πλευρές, τόσο από την πλευρά του πελάτη όσο και από την πλευρά του καταστήματος. Σε αυτήν την φόρμα ο πελάτης επιλέγει με την βοήθεια ενός dropdown list τον τρόπο πληρωμής με τον οποίο επιθυμεί να πραγματοποιηθεί η συναλλαγή, στην συνέχεια καταχωρεί τα στοιχεία αποστολής και στο τέλος εαν επιθυμεί να ολοκληρωθεί η παραγγελία τότε κάνει κλικ στο κουμπί «Ολοκλήρωση Παραγγελίας». Σε περίπτωση όμως που για οποιοδήποτε λόγο, δεν θέλει να προχωρήσει στην ολοκλήρωση της παραγγελίας, τότε κάνοντας κλικ στο κουμπί «Ακύρωση Παραγγελίας», την ακυρώνει.

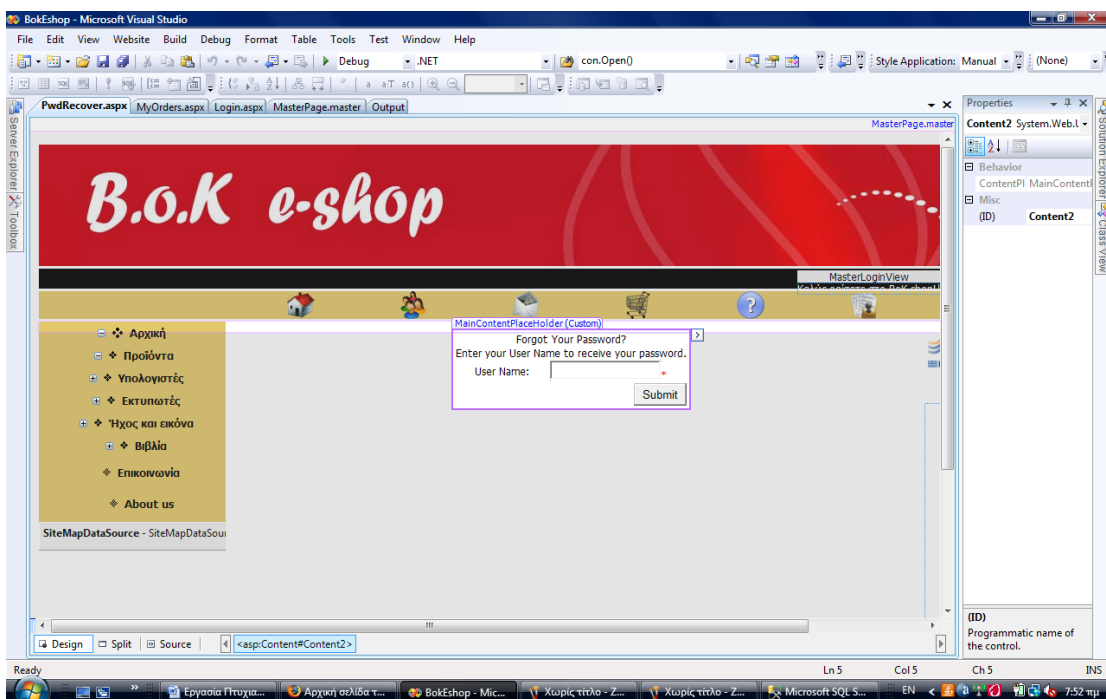
- Η φόρμα Login.aspx



Σχήμα Login.aspx

Η παραπάνω φόρμα Login.aspx είναι η φόρμα στην οποία ο πελάτης, κάνει login, εισάγοντας το username και το password του. Μόλις ολοκληρωθεί η διαδικασία με επιτυχία, τότε ο πελάτης μπορεί να προχωρήσει στην αγορά των προϊόντων που επιθυμεί. Σε περίπτωση που ο πελάτης επισκέπτεται για πρώτη φορά το κατάστημα και επιθυμεί να πραγματοποιήσει μια αγορά, τότε μπορεί να δημιουργήσει έναν νέο λογαριασμό κάνοντας κλικ στο hyperlink “Νέος Χρήστης”, ο οποίος υπερσύνδεσμος θα τον οδηγήσει στην φόρμα Register.aspx. Τέλος σε κάθε ένα από τα πεδία υπάρχει και ένα RequiredFieldValidator (είναι υποχρεωτικό να συμπληρωθεί για να δημιουργηθεί νέος λογαριασμός) με το οποίο υποδεικνύουμε στον πελάτη, να συμπληρώσει το κάθε πεδίο σωστά και παράλληλα γίνονται και οι αντίστοιχοί έλεγχοι. Η φόρμα login.aspx δίνει την δυνατότητα στον πελάτη να θυμηθεί τον κωδικό του, σε περίπτωση που τον έχει ξεχάσει, μέσω του υπερσυνδέσμου “Ξεχάσατε τον κωδικό σας?”. Σε αυτή την περίπτωση ο χρήστης μεταβαίνει στην PwdRecover.aspx όπου του στέλνεται ένα e-mail με τον κωδικό πρόσβασης του, εφόσον έχει απαντήσει στην ερώτηση ασφαλείας .

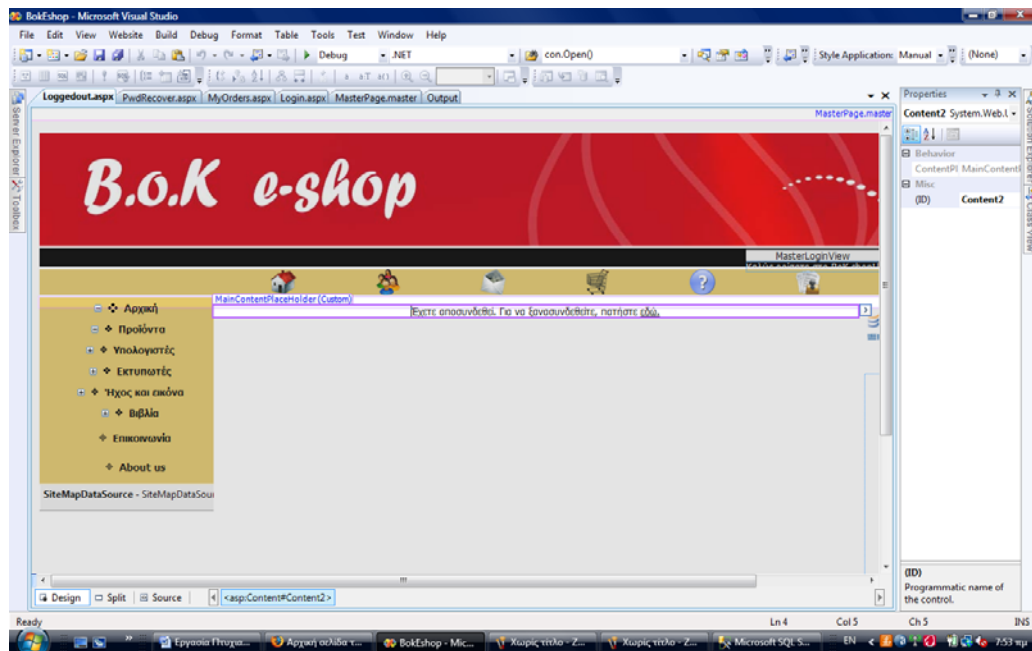
- Η φόρμα PwdRecover.aspx



Σχήμα PwdRecover.aspx

Με τη φόρμα PwdRecover.aspx της web εφαρμογής ο χρήστης έχει την δυνατότητα να “ανακτήσει” τον κωδικό πρόσβασης του, συμπληρώνοντας στην φόρμα PwdRecover.aspx το username του και απαντώντας στην ερώτηση ασφαλείας . Τότε του στέλνεται στο e-mail που έχει δηλώσει ένας νέος κωδικός, τον οποίο μπορεί να αλλάξει.

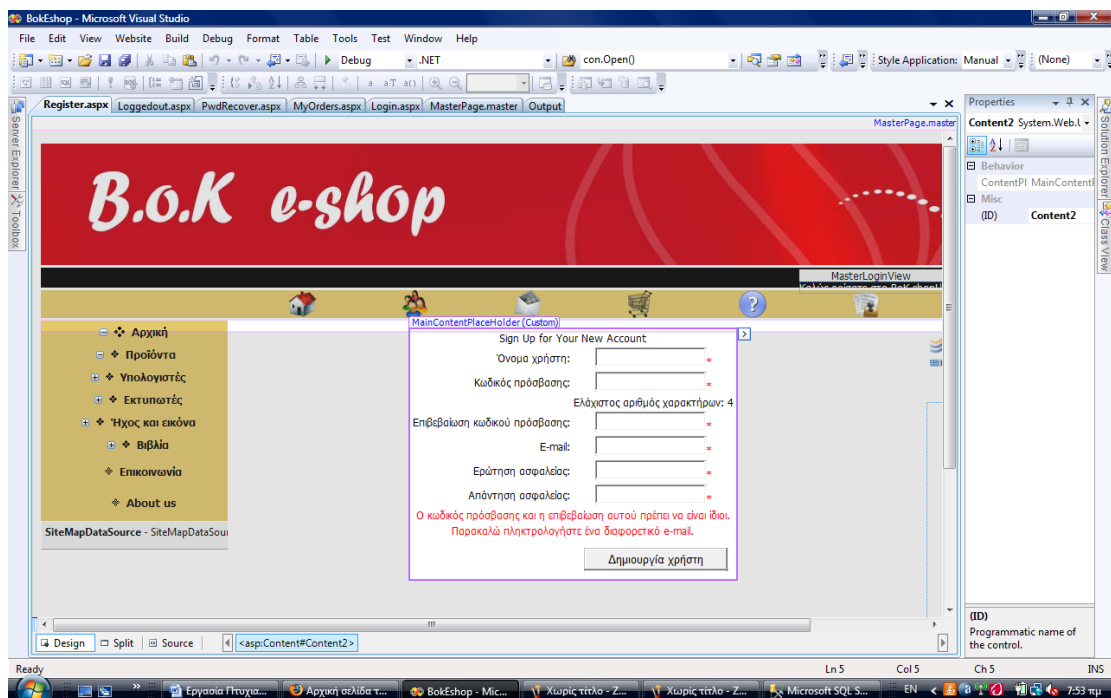
- Η φόρμα LogOut.aspx



Σχήμα LogOut.aspx

Η φόρμα LogOut.aspx είναι η φόρμα στην οποία, ο πελάτης έχει την δυνατότητα, πατώντας στον υπερσύνδεσμο «αποσύνδεση» να αποσυνδεθεί .Εμφανίζεται το μήνυμα “έχετε αποσυνδεθεί. Για να ξανασυνδεθείτε πατήστε εδώ. ”, όπου οδηγεί στην φόρμα login.aspx.

- Η φόρμα Register.aspx



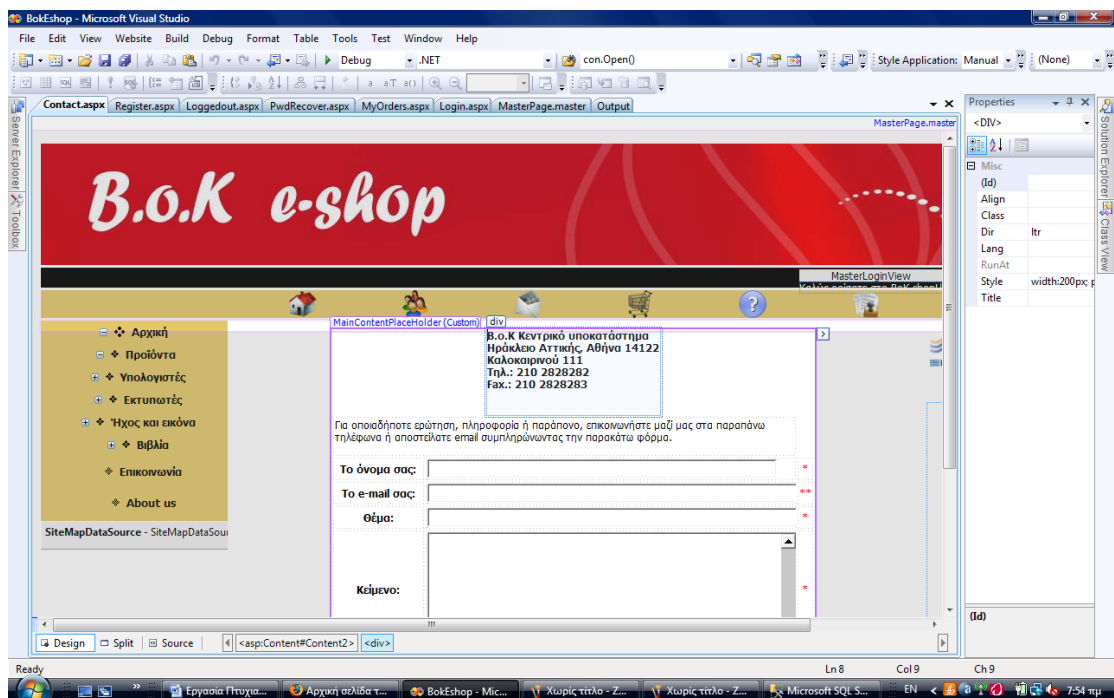
Σχήμα Register.aspx

Η φόρμα Register.aspx είναι η φόρμα στην οποία, όταν ένας πελάτης επισκέπτεται για πρώτη φορά το κατάστημα, μπορεί να δημιουργήσει έναν νέο λογαριασμό, με τον οποίο μπορεί να πραγματοποιήσει με ασφάλεια τις αγορές του. Πιο συγκεκριμένα στην φόρμα Register.aspx, ο πελάτης καλείται να συμπληρώσει μια φόρμα, αποτελούμενη από 6 πεδία. Μόλις συμπληρωθούν όλα τα πεδία, ο πελάτης αποκτά ένα νέο λογαριασμό στο ηλεκτρονικό κατάστημα, εφόσον κάνει κλικ πρώτα στο κουμπί «Δημιουργία Χρήστη». Όταν όλα τα βήματα γίνουν με την σωστή σειρά, τότε εμφανίζεται ένα μήνυμα το οποίο πληροφορεί τον πελάτη, ότι η καταχώρηση των στοιχείων του, έγινε με επιτυχία. Τέλος εάν ο πελάτης επιθυμεί να φύγει από την φόρμα αυτήν, μπορεί να το πετύχει χρησιμοποιώντας το κεντρικό μενού του καταστήματος, το οποίο υπάρχει σε κάθε φόρμα που το αποτελεί και από εκεί να πλοηγηθεί σε οποιαδήποτε σελίδα (φόρμα) του καταστήματος. Πρέπει επίσης να αναφέρουμε ότι σε κάθε ένα από τα 5 πεδία υπάρχει και ένα RequiredFieldValidator με το οποίο επισημένουμε στον πελάτη, να συμπληρώσει το κάθε πεδίο και παράλληλα γίνονται και οι αντίστοιχοι έλεγχοι.

Για να επικυρωθεί ο λογαριασμός, στέλνεται κατά την εγγραφή ένα mail στον χρήστη όπου του εμφανίζεται ένας υπέρσυνδεσμος όπως <http://localhost:49627/BeOk/authorize.aspx?ID=5971a108-a78d-4ab4-a363-f4be1b039032>

Κάνοντας κλικ πάνω σε αυτό, οδηγείται στην σελίδα authorize.aspx όπου ελέγχεται αν όντως έχει κάνει εγγραφή στην βάση. Αν ναι, τότε ο χρήστης εγκρίνεται και θεωρείται «έγκυρος χρήστης».

- **Η φόρμα ContactUs.aspx**



Σχήμα ContactUs.aspx

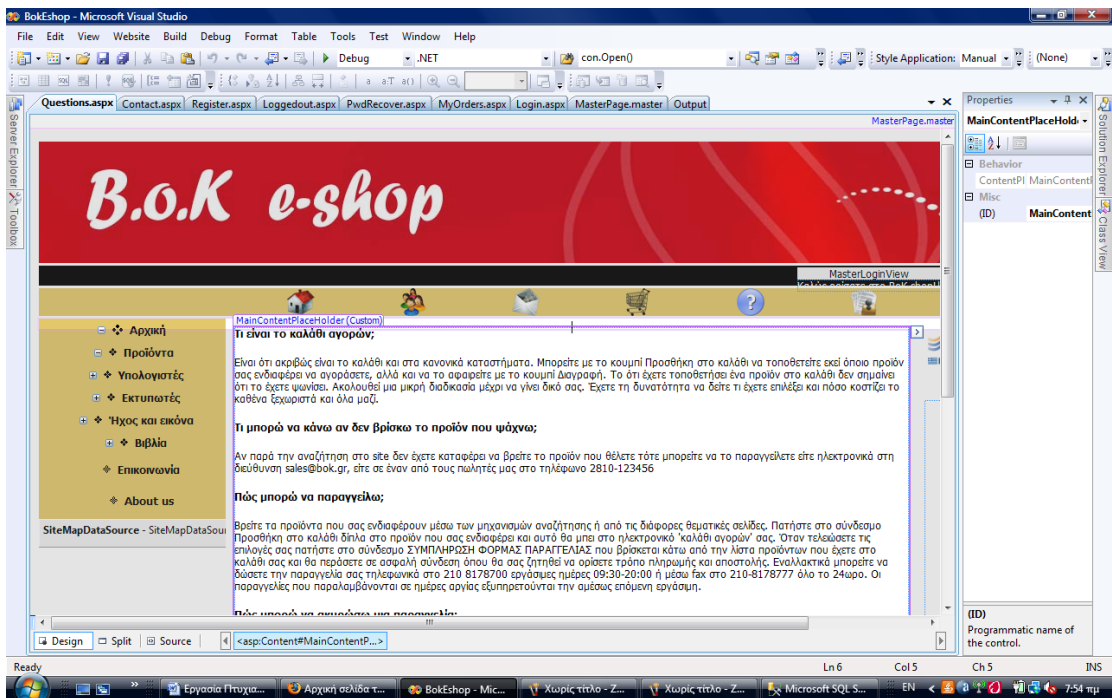
Η φόρμα ContactUs.aspx είναι η φόρμα στην οποία, ο πελάτης μπορεί να έρθει σε επικοινωνία με το κατάστημα για οποιοδήποτε λόγο. Για παράδειγμα την εξέλιξη της παραγγελίας, την υποβολή των παραπόνων του κτλ. Τα περιεχόμενα του textbox στέλνονται στο e-mail του καταστήματος, όπου έχει πρόσβαση ο administrator.

- **Η φόρμα AboutUs.aspx**

Σχήμα AboutUs.aspx

Στην φόρμα AboutUs.aspx της web εφαρμογής αναγράφονται κάποια στοιχεία του ηλεκτρονικού καταστήματος.

- Η φόρμα Questions.aspx



Στην φόρμα Questions.aspx της web εφαρμογής αναγράφονται οι πιο πιθανές ερωτήσεις με τις αντίστοιχες απαντήσεις τους, οι οποίες βοηθούν στην καλύτερη κατανόηση σε θέματα αγοράς προϊόντων.

5.2 Οι σελίδες των ADMINISTRATOR


Για τη σωστή λειτουργία και διαχείριση του ηλεκτρονικού καταστήματος, έχουμε δημιουργήσει κάποιες φόρμες τις οποίες τις διαχειρίζεται αποκλειστικά ο Administrator της εφαρμογής. Η βασική αρμοδιότητα του ο administrator είναι η διαχείριση της βάσης δεδομένων του ηλεκτρονικού καταστήματος. Αυτές οι φόρμες είναι οι εξής: ManageProd.aspx, ManageCategories.aspx, NewProduct.aspx, EditProduct.aspx, UserEdit.aspx, NewUserRegister.aspx, UsersManage.aspx.

- Η φόρμα **ManageProd.aspx**

Εχετε συνδεθεί ως: vassou | Διαχειριστείτε το προφίλ σας | Διακοπή σύνδεσης

24/9/2009 16:23:40

Νέο προϊόν

Κωδικός	Αριθμός κωδ.	Τίτλος	Περιγραφή	Εικόνα	Τιμή μον.	Stock	Κατηγορία	Διαγραφή	
1	PER.9120133	INNOVATOR OFFICE VALUE TFT 19"	Πληρότητα με μικρό κόστος! Με ΝΕΑ σύνθεση με τον εξοπλισμό INTEL DUAL CORE CELERON® E1500 και συσκευή		389,12 €	0	Desktops	Διαγραφή	Εισαγωγή
2	PER.906648	ACER ASPIRE 2930Z-322G 16MM T3200 20-48MB 160GB	Ο φορητός υπολογιστής της ACER, ASPIRE 2930Z-322 είναι ένα την σχεδίαση Gemstone και είναι ιδανικός		599,00 €	6	Laptops	Διαγραφή	Εισαγωγή
3	BKS.03068266	ΑΘΗΝΑ ΘΕΣΣΑΛΟΝΙΚΗ	Είναι και οι δύο παντρεμένοι. Ένα λάθος τηλεφώνημα τους φέρνει κοντά. Μεσ στη μοναξιά της προσωπικής τους		18,00 €	5	Literary	Διαγραφή	Εισαγωγή
4	PER.402463	HP DESKJET 6940 (C8970B)	Με τον καινούριο εκτυπωτή 6940 της HP μπορείτε να συνδέσετε εύκολα με πολλούς υπολογιστές και να		89,65 €	23	InkJet	Διαγραφή	Εισαγωγή
5	PER.402499	HP DESKJET D4360 CB700B	Ο νέος έγχρωμος inkjet εκτυπωτής, Hp Deskjet D4360, είναι ιδανικός για οικιακή χρήση, εκτυπώνοντας με		63,00 €	11	InkJet	Διαγραφή	Εισαγωγή

1 2 3 4 5

Βρίσκεστε στην σελίδα 1 από 5

Copyright © by Βασιλική Κωνσταντίνου & Κατερίνα Παπαδημητράκη

Στην φόρμα **ManageProd.aspx**, ο administrator έχει την δυνατότητα να διαχειρίζεται τα προϊόντα του καταστήματος που είναι περασμένα στη βάση δεδομένων, να διαγράψει κάποιο προϊόν ή να εισάγει καινούργιο μέσω του κουμπιού «Νέο προϊόν».

- Η φόρμα **NewProduct.aspx**



Σε αυτή την φόρμα οδηγείται αν θέλει να εισάγει νέο προϊόν μέσω της `manageprod.aspx`.

- Η φόρμα `EditProduct.aspx`



Σε αυτήν την φόρμα οδηγείται, αν πατήσει στο κουμπί «Επεξεργασία» της φόρμας `ManageProd.aspx` όπου μπορεί να επεξεργαστεί κάποιο προϊόν.

Εδώ χρησιμοποιείται ένα control της Asp, το FileUpload το οποίο είναι ευχρηστο στην εύρεση φωτογραφίας και εισαγωγή του path στην βάση μας. Ο κώδικας είναι ο παρακάτω για να γίνει η αποθήκευση στην βάση.

Try

```
Dim Id As Label = CType(Me.DetailsView1.Rows(0).Cells(0).FindControl("lbl_ProductId"), Label)
Dim lbl_id As Integer = CInt(Id.Text)

Dim ddl As DropDownList = CType(Me.DetailsView1.FindControl("ddl_Cat"), DropDownList)
Dim ddl_Value As Integer = ddl.SelectedValue()
Dim txt_Model As TextBox = CType(Me.DetailsView1.FindControl("txt_ModelNumber"), TextBox)
Dim txt_ModelNumber As String = txt_Model.Text

Dim txt_Title As TextBox = CType(Me.DetailsView1.FindControl("txt_Title"), TextBox)
Dim lbl_Title As String = txt_Title.Text

Dim txt_Description As TextBox = CType(Me.DetailsView1.FindControl("txt_Desc"), TextBox)
Dim Desc As String = txt_Description.Text

Dim lblOfImage As Image = CType(Me.DetailsView1.FindControl("Image2"), Image)
imagestring = CStr(lblOfImage.ImageUrl)

Dim FileUpload1 As FileUpload = TryCast(Me.DetailsView1.FindControl("FileUpload1"), FileUpload)
If FileUpload1.HasFile Then
    FileUpload1.SaveAs(Server.MapPath("~/Photos/") & FileUpload1.FileName)
    imagestring = "~/Photos/" & FileUpload1.FileName
End If

cmd.CommandText = "UPDATE Products SET Products_Modelnumber=@Products_Modelnumber,
Products_Title=@Products_Title, Products_Description = @Products_Description, Products_Image =
@Products_Image, " _
& " Products_Price = @Products_Price, Products_UnitsInStock = @Products_UnitsInStock, Products_SubCatID
= @Products_SubCatID WHERE Products_ProductID = @Products_ProductID "

cmd.Parameters.Add("@Products_ProductID", SqlDbType.Int).Value = lbl_id
cmd.Parameters.Add("@Products_Modelnumber", SqlDbType.NVarChar).Value = txt_ModelNumber
cmd.Parameters.Add("@Products_Title", SqlDbType.NVarChar).Value = lbl_Title
cmd.Parameters.Add("@Products_Description", SqlDbType.NText).Value = Desc
cmd.Parameters.Add("@Products_Image", SqlDbType.NVarChar).Value = imagestring
cmd.Parameters.Add("@Products_Price", SqlDbType.Money).Value = FormatPrice(txt_price.Text)

cmd.Parameters.Add("@Products_UnitsInStock", SqlDbType.Int).Value = s
cmd.Parameters.Add("@Products_SubCatID", SqlDbType.Int).Value = ddl_Value

cmd.Connection = con
con.Open()
cmd.ExecuteNonQuery()
```

- **Η φόρμα ManageCategories.aspx**

Εχετε συνδέσει ως vasso Διαχείριση Εισαγωγή προφίλ [Αποσύνδεση] 24/9/2009 16:35:28

ΕΣΠΑ 2007-2013
Προσφέρει για την πρόληψη Επιπτώσεων και Ελαφίσρων Επενδύσεων

πλ. παραγγελιών
800 800 1000
επί κινητό 210 2828200
9.00-17.00

Quest
xpsd
Αγοράστε τώρα ένα Quest Xpsd και αναβαθμίστε

Για να προσθέσετε κατηγορία προϊόντος.

Κωδικός κατηγορίας	Όνομα κατηγορίας
1	Computers
2	Televisions
3	Sound systems-Image
4	Books
5	Printers

Για να προσθέσετε υποκατηγορία προϊόντος.

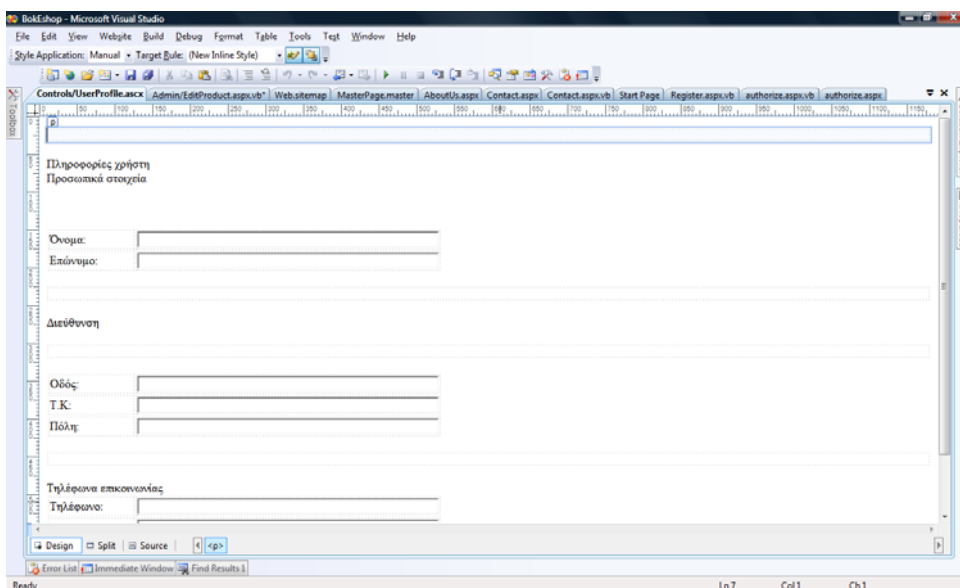
Όνομα κατηγορίας	Κωδικός κατηγορίας	Όνομα υποκατηγορίας	Κωδικός υποκατηγορίας
Computers	1	Desktops	1
Computers	1	Laptops	2
Televisions	2	26"	3
Televisions	2	22"	4
Books	4	Scientific	5
Books	4	Literary	6
Printers	5	Inkjet	7
Printers	5	Laser	8
Sound systems-Image	3	Home Cinema	11
Sound systems-Image	3	Photo machines	15

Στην φόρμα ManageCategories.aspx, ο administrator έχει την δυνατότητα να διαχειρίζεται τις κατηγορίες και τις υποκατηγορίες των προϊόντων του καταστήματος.

- Η φόρμα UserEdit.aspx



Στην φόρμα UserEdit.aspx καταγράφονται τα στοιχεία των χρηστών που εγγράφονται στο ηλεκτρονικό κατάστημα. Τα πεδία που τους χαρακτηρίζουν στη Βάση Δεδομένων είναι : Username, Email, Registered, Last Login, Last Activity, Online Now, Approved, Locked Out. Παρακάτω βλέπουμε το control UserProfile, στο οποίο ο χρήστης εισάγει τα στοιχεία του όσον αφορά το προφίλ. Η αποθήκευση των στοιχείων αυτών γίνεται στον πίνακα ProfileCustom και αυτά τα στοιχεία χρησιμοποιούνται για την αποστολή των προϊόντων, τα οποία μπορεί φυσικά να αλλάζει κάθε φορά.



- **Η φόρμα NewUserRegister.aspx**

Στην φόρμα NewUserRegister.aspx καταγράφονται τα στοιχεία των χρηστών που δημιουργούν λογαριασμό στο ηλεκτρονικό κατάστημα. Τα πεδία που τους χαρακτηρίζουν στη Βάση Δεδομένων είναι : Username, Password, Confirm Password, Email, Security Question, Security answer.

- **Η φόρμα UsersManage.aspx**

Εχετε συνδεθεί ως: vasso Διαχείριση Επεξεργασία προφίλ [Αποσύνδεση] | [Νέος χρήστης]

24/9/2009 16:42:37

Δημιουργία νέου χρήστη

Όνομα χρήστη	Έγκυρο	Email	Αποκαταμένος	Ημερ.δημιουργίας		
vassort	<input checked="" type="checkbox"/>	epp1283@epp.telher.gr	<input type="checkbox"/>	18/9/2009 08:40:20		
vasso2	<input checked="" type="checkbox"/>	epp1625@epp.telher.gr	<input type="checkbox"/>	29/3/2009 12:04:55		
vasso	<input checked="" type="checkbox"/>	vkonstantinou@cubeware.gr	<input type="checkbox"/>	18/2/2009 16:43:50		

Copyright © by Βασιλική Κωνσταντίνου & Κατερίνα Παπαδημητράκη

Στην φόρμα UsersManage.aspx, ο administrator έχει την δυνατότητα να παρακολουθεί την ενεργητικότητα των χρηστών του καταστήματος μέσω της βάσης δεδομένων. Μπορεί δηλαδή να δει μέσα από τη βάση το username, το e-mail του, τότε δημιουργήθηκε ο λογαριασμός του, εάν έχει κάνει authenticate ή όχι. Επίσης μπορεί να επεξεργαστεί και τα στοιχεία του όπως και να αλλάξει τον ρόλο στον οποίο υπάγεται, ας πούμε να τον κάνει administrator μέσω της φόρμας UserEdit.aspx.

- Η φόρμα ManageOrders.aspx

Εχετε συνδεθεί ως **vasso** Διαχείριση Επεξεργασία προφίλ [Αποσύνδεση] | [Νέος χρήστης]

24/9/2009 16:46:19

Πατήστε εδώ ώστε να δείτε τι παραγγελίες έχουν γίνει σήμερα!

Σημερινές παραγγελίες

Όλες οι παραγγελίες

	Κωδ.παραγγελίας	Ημερ.παραγγ.	Ημερ. παράδοσης	Κωδ.προϊόντος	Ποσότητα	Σύνολο	Κατάσταση	Τρόπος πληρωμής	
Επεξεργασία	115	22/9/2009 00:00:00		19	1	22,1400	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή
Επεξεργασία	116	22/9/2009 00:00:00		19	1	22,1400	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή
Επεξεργασία	117	22/9/2009 00:00:00		19	1	22,1400	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή
Επεξεργασία	118	22/9/2009 00:00:00		19	2	44,2800	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή
Επεξεργασία	119	22/9/2009 00:00:00		29	1	94,0500	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή

1 2 3 4 5 6

Βρίσκεστε στην σελίδα 1 από 6

Copyright © by Βασιλική Κωνσταντίνου & Κατερίνα Παπαδημητράκη

Στην φόρμα αυτή ο διαχειριστής μπορεί να δει τι παραγγελίες έχουν γίνει στο κατάστημα του, διάφορα στοιχεία, καθώς μπορεί επίσης να επεξεργαστεί την κατάσταση της παραγγελίας πατώντας το «Επεξεργασία».

Κάνοντας κλικ στο «Επιλογή» του εμφανίζονται πιο κάτω αναλυτικά τα στοιχεία που αφορούν την παραγγελία που επέλεξε.

Εχετε συνδεθεί ως **vasso** Διαχείριση Επεξεργασία προφίλ [Αποσύνδεση] | [Νέος χρήστης]

24/9/2009 16:59:42

Πατήστε εδώ ώστε να δείτε τι παραγγελίες έχουν γίνει σήμερα!

Σημερινές παραγγελίες

Όλες οι παραγγελίες

	Κωδ.παραγγελίας	Ημερ.παραγγ.	Ημερ. παράδοσης	Κωδ.προϊόντος	Ποσότητα	Σύνολο	Κατάσταση	Τρόπος πληρωμής	
Επεξεργασία	143	23/9/2009 00:00:00		17	1	1119,0000	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή
Επεξεργασία	144	23/9/2009 00:00:00		15	1	829,0000	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή
Επεξεργασία	145	23/9/2009 00:00:00		25	1	129,0000	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή
Επεξεργασία	147	23/9/2009 00:00:00		10	1	395,0000	<input type="checkbox"/>	Με αντικαταβολή	Επιλογή
Επεξεργασία	148	23/9/2009 00:00:00		25	1	129,0000	<input type="checkbox"/>	Με Paypal	Επιλογή

1 2 3 4 5 6

Βρίσκεστε στην σελίδα 5 από 6

Στοιχεία παραλήπτη

Κωδ.παραγγελίας	144
Κωδ.χρήστη	109e0b7b-fb0c-4fa3-8fb9-8f783a1c0678
Όνομα χρήστη	vassort
Ημερ.παραγγελίας	23/9/2009 00:00:00
Ημερ.παράδοσης	
Κωδ.προϊόντος	15
Ποσότητα	1
Συνολικό κόστος	829,0000
Κατάσταση παραγγελίας	<input type="checkbox"/>
Όνομα	sdffsdff
Επώνυμο	sdffsdffsdff
Διεύθυνση	324234
Πόλη	sdffsdffsdff
Ταχυδρομικός κωδ.	3423423
Κινητό τηλ.	4534534534

Copyright © by Βασιλική Κωνσταντίνου & Κατερίνα Παπαδημητράκη

Ο διαχειριστής έχει την δυνατότητα να δει τις σημερινές παραγγελίες σε φύλλο Excel, ώστε να μπορεί να το αποθηκεύσει τοπικά στον υπολογιστή του.

6. Επίλογος

Ως αποτέλεσμα της εργασίας μας, συμπεραίνουμε ότι στις διαδικτυακές εφαρμογές τα πιο κρίσιμα και ευάλωτα σημεία είναι τα θέματα πιστοποίησης χρήστη, διασφάλιση προσωπικών δεδομένων και οι ασφαλή συναλλαγές.

Πρέπει ο χρήστης να νιώθει σίγουρος ότι δεν μπορεί τρίτο άτομο να αποκτήσει πρόσβαση στα στοιχεία του, αλλά και από την πλευρά του καταστηματαρχη, πρέπει να είναι σίγουρος ότι θα πληρωθεί και όλα θα διεξαχθούν ομαλά.

7. Κεφάλαιο - Βιβλιογραφία – Πηγές

ASP.NET , Ryan, Danny , John Wiley and Sons Ltd 2001

Programming ASP.NET , Liberty, Jesse, Hurwitz, Dan , O'Reilly 2003

Asp.Net 2.0 Revealed , Lorenz, Patrick , Springer-Verlag Berlin and Heidelberg GmbH & Co. KG 2003

Mastering ASP.NET with VB.Net , Jones, A.Russell , John Wiley and Sons Ltd 2002

Building Secure Microsoft ASP.NET Applications , Microsoft Press , Microsoft Press,U.S. 2003

Distributed Data Applications with ASP.NET , Homer, Alex, Sussman, Dave , Springer-Verlag Berlin and Heidelberg GmbH & Co. KG 2004

ASP.NET Data Web Controls , Mitchell, Scott , Macmillan Computer Publishing (a Pearson Education company) 2003

Professional ASP.NET 2.0 , Evjen, Bill, Hanselman, Scott, Muhammad, Farhan , John Wiley and Sons Ltd 2005

Beginning ASP.Net 2.0 Databases , Kauffman, John, Thangarathinam, Thiru , John Wiley and Sons Ltd , 2005

Programming Microsoft ASP.NET 2.0 Applications , Esposito, D. , Microsoft Press,U.S. 2006

Beginning ASP.NET 1.1 E-Commerce , Darie and Watson , Apress , 2004

ASP.NET E-Commerce Programming : Problem – Design – Solution , Hoffman , Wrox , 2003

Beginning ASP.NET 2.0 , Ullman, Chris, Kauffman, John, Hart, Chris , John Wiley and Sons Ltd , 2005

Beginning ASP.Net 1.1 Databases , Maharry, Dan, Foggon, Damien , APress,US 2004

Visual Basic .Net “Εισαγωγή στο .NET και τα Web Services”, δεύτερη έκδοση , Deitel & Nieto , Μ. Γκιούρδας , 2003

Πλήρες Εγχειρίδιο της Visual Basic .NET , Πετρούτσος , Μ. Γκιούρδας , 2005
SQL Server 2000 Web Application Developer's Guide , Craig Utley, Berkley 2001.

Rebecca M. Riordan, *Προγραμματισμός στο SQL Server 2000 Βήμα Βήμο* (μτφρ. Τάκης Άλβας), Αθήνα 2001.

Ιστοσελίδες:

www.asp.net

www.msdn.microsoft.com