

ΑΝΩΤΑΤΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ



Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

Πτυχιακή Εργασία
«Ολοκληρωμένη Εφαρμογή Ηλεκτρονικού Καταστήματος
με βάση την τεχνολογία ASP.NET»

Ντελλή Αθηνά(ΑΜ:2040)



Επόπτες Καθηγητές
Κλεισαρχάκης Μιχάλης
Μαλάμος Αθανάσιος

Ηράκλειο 2010

Περίληψη

Σκοπός της παρούσας πτυχιακής εργασίας είναι η υλοποίηση μιας ολοκληρωμένης εφαρμογής ηλεκτρονικού καταστήματος η οποία δεν θα προσανατολίζεται μόνο στην παρουσίαση του ηλεκτρονικού καταλόγου των προϊόντων πώλησης. Πιο συγκεκριμένα ενδιαφερόμαστε για την υλοποίηση βασικών λειτουργικών δραστηριοτήτων ενός εμπορικού ιστοχώρου και η μελέτη μας είναι σε σχέση με τους ρόλους των διαφόρων χρηστών που μπορούν να επισκεφτούν ή να διαχειριστούν τον ιστοχώρο αυτό. Θα προβλέψουμε τρεις διαφορετικές ομάδες χρηστών : (α) τους απλούς χρήστες (guests) που δεν εγγράφονται στη βάση δεδομένων του ιστοχώρου, (β) τους εγγεγραμμένους χρήστες (registered users) που απολαμβάνουν περισσότερα προνόμια από τους απλούς χρήστες και (γ) τους διαχειριστές (admins) του ιστοχώρου που έχουν δικαιώματα άμεσης πρόσβασης στην βάση δεδομένων του συστήματος (π.χ. ενσωμάτωση νέων προϊόντων, αλλαγές τιμών, διαχείριση αποθήκης, διαχείριση χρηστών) κλπ.

Η τεχνολογία που θα χρησιμοποιηθεί για την παραπάνω υλοποίηση είναι η ASP.NET σε περιβάλλον Visual Studio.NET με την χρήση της γλώσσας προγραμματισμού C-Sharp (C#). Θεωρούμε ότι η τεχνολογία αυτή είναι ότι πιο ολοκληρωμένο υπάρχει στην αγορά για ολοκληρωμένα IDE's που αφορούν το Web Development. Επιπροσθέτως θα ενσωματώσουμε λύσεις που αφορούν την ταχύτερη προώθηση ιστοσελίδων στους χρήστες από τον Web Server. Μια τέτοια μελέτη αφορά στην ενσωμάτωση της τεχνολογίας AJAX που βοηθά μέγιστα στην ταχεία προώθηση και ανανέωση των ιστοσελίδων στη μεριά των χρηστών.

Περιεχόμενα

1. ΚΕΦΑΛΑΙΟ I : ΚΑΤΗΓΟΡΙΕΣ ΚΑΙ ΟΦΕΛΗ ΣΧΕΤΙΚΑ ΜΕ ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ.....	5
Εισαγωγή	5
Η τεχνολογία του καλαθιού αγοράς.....	7
Βασικές κατηγορίες ηλεκτρονικού εμπορίου	7
Οι νέες κατηγορίες του ηλεκτρονικού εμπορίου	12
Τα πλεονεκτήματα του ηλεκτρονικού εμπορίου.....	13
Τα μειονεκτήματα του ηλεκτρονικού εμπορίου	15
2. ΚΕΦΑΛΑΙΟ II: ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ ΚΑΙ ΤΑ ΜΟΝΤΕΛΑ ΤΟΥ	17
Εισαγωγή	17
Τα πλεονεκτήματα του ηλεκτρονικού μάρκετινγκ	17
Το μοντέλο του ηλεκτρονικού καταστήματος	18
Το μοντέλο της ηλεκτρονικής προμήθειας	19
Το μοντέλο της ηλεκτρονικής δημοπρασίας.....	21
Το μοντέλο του εμπορικού κέντρου ή πολυκαταστήματος	28
Το μοντέλο της ηλεκτρονικής πύλης	29
Μοντέλα δυναμικής διαμόρφωσης τιμών	31
3. ΚΕΦΑΛΑΙΟ III: Η ΧΡΗΣΗ ΤΟΥ ΕΡΓΑΛΕΙΟΥ VISUAL STUDIO.NET ΚΑΙ ΤΗΣ ASP.NET	38
Εισαγωγή και γρήγορη ματιά στο λογισμικό.....	38
Δημιουργία ενός web site	41
Επεξεργασία σελίδων	41
Διαμόρφωση ιστοχώρου με guest χρήστες και registered χρήστες.....	45
Η χρήση του website administration tool	45
Δημιουργία ρόλων για την κατηγοριοποίηση χρηστών	47
Δημιουργία κανόνων πρόσβασης	49
Διαχείριση κανόνων πρόσβασης.....	50
Δημιουργία λογαριασμού χρήστη.....	51
Δημιουργία master pages.....	52
Βασικές ιδιότητες και εργαλεία της asp.net.....	54
Asp.net login controls.....	56
Ιδιοτικοποίηση χρηστών στην asp.net	58
Εργαλεία πλοήγησης	59
Βάσεις δεδομένων και sql για την διαχείριση δεδομένων σε ιστοχώρους.....	61
Καλάθι αγορών και παραγγελίες	67
4. ΚΕΦΑΛΑΙΟ IV: ΥΛΟΠΟΙΗΣΗ ΕΝΟΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ ΜΕ ΘΕΜΑ «ΚΑΤΟΙΚΙΔΙΑ ΖΩΑ» E-PET-SHOP	69
Εργαλεία υλοποίησης	69
Περιγραφή του ηλεκτρονικού καταστήματος για την ομάδα των μη εγγεγραμμένων χρηστών	72
Περιγραφή του ηλεκτρονικού καταστήματος για την ομάδα των εγγεγραμμένων χρηστών.....	78
Περιγραφή του ηλεκτρονικού καταστήματος για την ομάδα των διαχειριστών.....	83
Περιγραφή του ηλεκτρονικού καταστήματος όσον αφορά τη βάση δεδομένων	90

5. ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ ΤΟΥ WEBSITE	93
Pet class (Pet.cs)	93
Category class (Category.cs)	95
User class (User_class.cs).....	97
Cart Item Class (Cart_Item.cs)	99
Credit Card Validation class (CreditCardValidation.cs).....	101
Order Status class (Order_Status.cs).....	102
Data Access class (Data_Access.cs)	103
6. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	117

ΚΕΦΑΛΑΙΟ Ι : ΚΑΤΗΓΟΡΙΕΣ ΚΑΙ ΟΦΕΛΗ ΣΧΕΤΙΚΑ ΜΕ ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ

ΕΙΣΑΓΩΓΗ

Υπάρχουν πολλά οφέλη για να επεκταθεί μια επιχείρηση στο Διαδίκτυο. Το ηλεκτρονικό εμπόριο μπορεί να προσφέρει υψηλής ποιότητας υπηρεσίες όσον αφορά την εξυπηρέτηση πελατών. Επίσης είναι και ένας από τους βασικούς παράγοντες που επηρεάζουν και βελτιώνουν την εφοδιαστική-αλυσίδα μιας επιχείρησης. Με αυτόν λοιπόν τον τρόπο οι επιχειρήσεις μπορούν να δείξουν και να εφαρμόσουν μια νέα στρατηγική. Αυτή η στρατηγική αφορά δύο πράγματα: πρώτον τη δυνατότητα της επιχείρησης να αυξήσει το τζίρο της με την καινούργια αυτή τεχνολογία, γιατί μπορεί να έχει εύκολο μάρκετινγκ και δεύτερον κάνει πιο εύκολο το μηχανισμό της διαχείρισης της διανομής προϊόντων. Σε αυτό το κεφάλαιο, ερευνάμε τους διαφορετικούς τύπους επιχειρήσεων που λειτουργούν στο διαδίκτυο, καθώς επίσης και τις τεχνολογίες που απαιτούνται για να χτίσουμε και να διαχειριστούμε ένα web site ηλεκτρονικού εμπορίου.

Οι διάφοροι επιχειρηματίες που αρχίζουν κάποιο ηλεκτρονικό κατάστημα αλλά και διάφοροι άλλοι άνθρωποι που ενδιαφέρονται για το ηλεκτρονικό εμπόριο πρέπει να γνωρίζουν τα διάφορα πρότυπα ηλεκτρονικού εμπορίου. Σε αυτό το εισαγωγικό κεφάλαιο της πτυχιακής εργασίας μας θα κάνουμε μια γενική επισκόπηση για το ποιες είναι οι βασικές κατηγορίες της ηλεκτρονικής επιχειρηματικότητας. Και επίσης θα μιλήσουμε επίσης για το ηλεκτρονικό καλάθι αγορών, γιατί πιστεύουμε ότι η μεθοδολογία αυτή μπόρεσε να πείσει τους διάφορους που θέλουν να ψωνίσουν από ένα κατάστημα να αγοράσουν τελικά τα προϊόντα του καταστήματος. Οι επιχειρήσεις που λειτουργούν με βάση κάποια από τις κατηγορίες για τις οποίες θα μιλήσουμε παρακάτω μπορούν με μεγάλη ενέργεια και με την βοήθεια της τεχνολογίας να διαφοροποιηθούν από τις άλλες παραδοσιακές επιχειρήσεις και να ανταπεξέλθουν στο σημερινό ανταγωνισμό της αγοράς.

Από τη δεκαετία του '60, όταν άρχισαν να διαδίδονται ευρέως οι ηλεκτρονικοί υπολογιστές (H/Y) και αργότερα, όταν συνδυάστηκαν με τις τηλεπικοινωνίες, ήταν φανερό ότι η ψηφιακή τεχνολογία θα έφερνε μια τεράστια αλλαγή σε όλους τους

κλάδους της ανθρώπινης δραστηριότητας και στη κοινωνία γενικότερα, μια αλλαγή το ίδιο σημαντική όσο και αυτή που προκάλεσαν ο ατμός και ο ηλεκτρισμός. Ο ιδρυτής της Microsoft Bill Gates πιστεύει ότι «ο κόσμος θα αλλάξει τα επόμενα δέκα χρόνια περισσότερο απ'ότι άλλαξε την τελευταία πεντηκονταετία». Αυτό είπε το 1989. Πραγματικά αυτό φάνηκε στα τέλη του περασμένου αιώνα. Δηλαδή, φάνηκε πόσο πολύ άλλαξε η επιχειρηματικότητα με την είσοδο του ηλεκτρονικού εμπορίου στις επιχειρήσεις.

Η κίνηση προς το ηλεκτρονικό εμπόριο παρουσιάζει πολλά οφέλη, καθώς επίσης και διάφορες μεγάλες προκλήσεις. Το πρότυπο ή μοντέλο storefront είναι αυτό που πολλοί άνθρωποι σκέφτονται όταν ακούν τη λέξη ή φράση «ηλεκτρονικό εμπόριο». Επί της ουσίας εδώ είναι το μοντέλο όπου κάποιες επιχειρήσεις που έχουν ήδη κάποιο χώρο και κάποια καταστήματα, μπαίνουν στο ηλεκτρονικό εμπόριο και αποκτούν κάποιο ιστοχώρο για να αυξήσουν τα έσοδα τους μέσα από ηλεκτρονικές πωλήσεις κάποιων ή όλων των προϊόντων τους. Το πρότυπο storefront συνδυάζει την επεξεργασία συναλλαγών, την ασφάλεια, την πληρωμή των προϊόντων on-line καθώς και την αποθήκευση πληροφοριών και με αυτό τον τρόπο μπορεί να βοηθήσει τους εμπόρους για να πωλήσουν τα προϊόντα τους on-line. Δηλαδή, Αυτό το πρότυπο είναι η βασική μορφή ηλεκτρονικού εμπορίου όπου εδώ ο αγοραστής και ο πωλητής αλληλεπιδρούν άμεσα μέσα από μία ιστοσελίδα.

Μέχρι πριν λίγο καιρό, το ηλεκτρονικό εμπόριο ήταν εφικτό μόνο για τις μεγάλες επιχειρήσεις. Εντούτοις, το Διαδίκτυο και ο παγκόσμιος ιστός είναι τόσο προσιτή τεχνολογία για τους καθημερινούς χρήστες του διαδικτύου και έτσι η τεχνολογία αυτή στην πόρτα μας καθιστά το ηλεκτρονικό εμπόριο πιθανό και για τις μικρές επιχειρήσεις.

Το ηλεκτρονικό εμπόριο επιτρέπει επίσης στις επιχειρήσεις να γίνεται διαχείριση της επιχείρησης 24 ώρες το εικοσιτετράωρο, όλη την ημέρα, καθημερινά και σε παγκόσμια βάση. Μερικά από τα πιο επιτυχημένα ηλεκτρονικά καταστήματα χρησιμοποιούν το πρότυπο storefront. Τα μεγαλύτερα παραδείγματα περιλαμβάνουν όλες τις μεγάλες επιχειρήσεις ηλεκτρονικών ειδών του εξωτερικού όπως για παράδειγμα την εταιρία Hewlett Packard που πουλά ηλεκτρονικούς υπολογιστές ή την εταιρία σε ακριβώς αντίθετη επιχειρησιακή κατεύθυνση Starbucks που πουλάει είδη καφέ. Πολλές από αυτές τις κορυφαίες πρότυπες επιχειρήσεις είναι επιχειρήσεις του τύπου B2C δηλαδή Business-to-Consumer ή αλλιώς στα ελληνικά «επιχείρηση

προς καταναλωτές» επιχειρήσεις. Παραδείγματος χάριν, το More.com είναι ένα παράδειγμα ηλεκτρονικού καταστήματος που αφορά την υγεία και την ομορφιά η οποία χρησιμοποιεί ένα ηλεκτρονικό καλάθι αγορών για να επιτρέψει στους πελάτες να ψωνίσουν και να αγοράσουν προϊόντα τα οποία στέλνονται στην πόρτα του σπιτιού τους από την επιχείρηση.

Η ΤΕΧΝΟΛΟΓΙΑ ΤΟΥ ΚΑΛΑΘΙΟΥ ΑΓΟΡΑΣ

Μια από τις πιο συνηθέστερες τεχνολογίες ηλεκτρονικού εμπορίου είναι το καλάθι αγορών. Αυτή η τεχνολογία είναι τεχνολογία επεξεργασίας αγορών και συναλλαγών σε κάποιο ηλεκτρονικό κατάστημα και επιτρέπει στους πελάτες του καταστήματος να συσσωρεύσουν τα προϊόντα που επιθυμούν να αγοράσουν δεδομένου μπορούν να συνεχίζουν να ψωνίζουν και η πληρωμή των προϊόντων θα γίνει συνολικά για όλες τις αγορές λίγο πριν την έξοδο τους από το ηλεκτρονικό κατάστημα. Η υποστήριξη του καλαθιού αγορών γίνεται με βάση κάποιο κατάλογο προϊόντων, ο οποίος φιλοξενείται στη βάση δεδομένων του εξυπηρετητή και στον κεντρικό υπολογιστή που επίσης φιλοξενεί και το συγκεκριμένο ιστοχώρο, δηλαδή το ηλεκτρονικό κατάστημα. Ο κεντρικός υπολογιστής του site είναι το βασικό εργαλείο διαχείρισης και αποθήκευσης στοιχείων από τον επιχειρηματία. Αυτοί οι υπολογιστές είναι συχνά υπολογιστές servers που διευθύνουν όλες τις λειτουργίες απαραίτητες για το τρέξιμο μιας ιστοσελίδας ηλεκτρονικού καταστήματος.

Η βάση δεδομένων είναι ένα μέρος του κεντρικού υπολογιστή με σκοπό να αποθηκεύσει και να κάνει έκθεση σχετικά με τις πληροφορίες προϊόντων. Παραδείγματος χάριν, μια βάση δεδομένων για ένα ηλεκτρονικό κατάστημα που κάνει λιανεμπόριο ρούχων θα περιελάμβανε τα χαρακτηριστικά και τις προδιαγραφές τέτοιων προϊόντων όπως για παράδειγμα, μια αναλυτική περιγραφή του προϊόντος, μια φωτογραφία, το μέγεθος, τη διαθεσιμότητα, τις πληροφορίες για της αποστολή και διανομή του, το επίπεδο αποθεμάτων και διάφορες άλλες πληροφορίες.

ΒΑΣΙΚΕΣ ΚΑΤΗΓΟΡΙΕΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ

Το ηλεκτρονικό εμπόριο διαιρείται συνήθως σε **τρεις κύριες κατηγορίες** ηλεκτρονικών συναλλαγών με βάση τον διαχωρισμό που κάνουν οι Korper και Ellis στο βιβλίο τους «The E-Commerce book – Building the Empire»:

1. Ηλεκτρονικό εμπόριο μεταξύ επιχειρήσεων – Διεπιχειρηματικό εμπόριο (Business to Business ή B2B). Αυτή η βασική κατηγορία για το ηλεκτρονικό εμπόριο επιτρέπει στις επιχειρήσεις να βελτιώσουν τη μεταξύ τους συνεργασία, απλοποιώντας τις διαδικασίες των προμηθειών, μειώνοντας το κόστος των προμηθειών και επίσης κάνοντας ταχύτερη την αποστολή των προϊόντων τους με τον αποτελεσματικότερο έλεγχο του επιπέδου των διαθέσιμων αποθεμάτων. Επιπλέον, τα περισσότερα από τα έγγραφα που ανταλλάσσονται όταν γίνεται μια συναλλαγή ανάμεσα στις δύο επιχειρήσεις, όπως παραγγελίες, τιμολόγια, επιταγές κλπ. αυτά μπορούν να παράγονται σε ηλεκτρονική μορφή και να εκτυπώνονται μόνο στο σημείο παραλαβής με αποτέλεσμα το ηλεκτρονικό εμπόριο αυτής της μορφής να αποτελεί μια ελκυστική εναλλακτική λύση απέναντι στη σημερινή διαδικασία της δακτυλογράφησης, ταχυδρομικής αποστολής κ.λπ., που κοστίζει σε χρήμα, χρόνο και ανθρώπινα λάθη. Αυτή η κατηγορία εμπορίου τυπικά περιλαμβάνει τους προμηθευτές, τους αντιπροσώπους-διανομής, τους κατασκευαστές, τους χώρους αποθήκευσης κλπ.

Ένα παράδειγμα θα μας βοηθήσει να αντιληφθούμε πλήρως την διαδικασία B2B. Υποθέτουμε ότι μια αεροπορική εταιρεία επιθυμεί να κατασκευάσει ένα αεροπλάνο. Όπως όλοι μας καταλαβαίνουμε η κατασκευή ενός αεροπλάνου απαιτεί πολλά διαφορετικά μηχανικά και μη μέρη από μεγάλους και μικρούς προμηθευτές. Ο σκοπός του ηλεκτρονικού εμπορίου εδώ είναι η αυτοματοποίηση ολόκληρης της αλυσίδας προμηθειών. Ουσιαστικά, στο παράδειγμα αυτό, η αυτοματοποίηση είναι υπό την «ομπρέλα» της διοίκησης και οργάνωσης της αλυσίδας προμηθειών, της διαδικασίας δηλαδή συνένωσης πολλών διαφορετικών προμηθευτών αγαθών για να δημιουργηθούν τα τελικά προϊόντα.

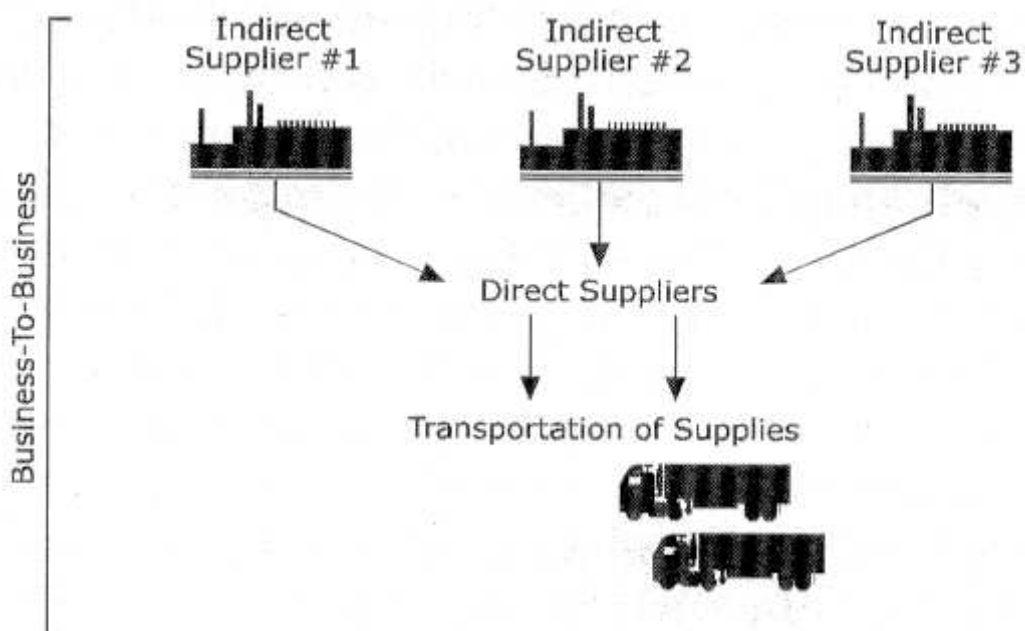
Το Διαδίκτυο προσφέρεται σαν το ιδανικό μέσο για μεταφορά αιτήσεων, ευαίσθητων πληροφοριών καθώς και μιας ποικιλίας από ογκώδη αρχεία. Επομένως επιτρέπει στον επιχειρηματία να επεκτείνει την επιχείρησή του, επικοινωνώντας με προμηθευτές και αγοραστές και να αναπτύξει, όσο το δυνατό, περισσότερες on line λύσεις.

Στα ωφέλη από τη χρήση μεθοδολογιών και τεχνικών ηλεκτρονικού διεπιχειρησιακού εμπορίου περιλαμβάνονται σύμφωνα με τους Rayport και Jaworski που έγραψαν ένα άλλο εισαγωγικό βιβλίο που αφορά το ηλεκτρονικό εμπόριο περιλαμβάνονται τα εξής:

- Η καλύτερη διοίκηση αποθηκών και εφοδιαστικής αλυσίδας τόσο για τον προμηθευτή όσο και για τον αγοραστή.
- Η ταχύτερη προσαρμογή στις αλλαγές των απαιτήσεων των πελατών.
- Η ταχύτερη εισαγωγή στην αγορά νέων προϊόντων.
- Η δραστική μείωση του γραφειοκρατικού κόστους.
- Η επίτευξη χαμηλότερων τιμών για κάποια προϊόντα σε σχέση με τις φυσικές αγορές.
- Η μείωση του πλήθους των επισφαλών αγορών.

2. Ηλεκτρονικό εμπόριο μεταξύ επιχειρήσεων και καταναλωτών. Αυτό το μοντέλο αφορά περισσότερο τη διεπιχειρηματικότητα είναι οποία είναι περισσότερο προσανατολισμένη προς τον καταναλωτή (Business-to-Consumer, B2C). Επίσης μια άλλη κατηγορία που όμως ανήκει σε αυτή τη βασική κατηγορία την οποία εξηγούμε σε αυτή την ενότητα είναι επίσης το κομμάτι που αφορά και στην Διεπιχειρηματικότητα μεταξύ μιας Επιχείρησης και τους Κράτους που ονομάζεται Business-to Government, B2G. Αυτή λοιπόν η κατηγορία ηλεκτρονικού εμπορίου αποτελεί την πιο διαδεδομένη μορφή ηλεκτρονικού εμπορίου όπου περιλαμβάνει τις λιανικές πωλήσεις μέσω του Internet προς το ευρύ καταναλωτικό κοινό και μπορεί να παραδοθεί μέσω δικτύου, όπως συμβαίνει με ένα λογισμικό, ή να πακεταριστεί, όπως γίνεται με ένα βιβλίο. Σε μια μεγάλη έρευνα που διενήργησε το περιοδικό «Economist» για το e-commerce αναφέρει τα παρακάτω για το B2C [πηγή : Peet, 2000, βλέπε βιβλιογραφία): «*Η διεπιχειρηματικότητα προσανατολισμένη προς τον καταναλωτή περιλαμβάνει όλες τις κανονικές δραστηριότητες του λιανικού εμπορίου στο Διαδίκτυο, όπως την πώληση βιβλίων από την Amazon.com ή τις on-line αγοροπωλησίες μετοχών από τον Charles Schwab*». Σύμφωνα με την άποψη του περιοδικού, απορρέει ότι το ηλεκτρονικό εμπόριο B2C περιλαμβάνει ενέργειες και συναλλαγές μεταξύ μιας επιχείρησης και των πελατών της οι οποίες είναι αμφίδρομες, δηλαδή και ως προς την κατεύθυνση του καταναλωτή αλλά και ως προς την κατεύθυνση της επιχείρησης. Οι περισσότεροι άνθρωποι είναι σχετικοί με το μοντέλο ηλεκτρονικού εμπορίου B2C. Οι περισσότερες εταιρείες της Αμερικής όπως η Dell ή η Amazon έγιναν γνωστές σε πολλά νοικοκυριά. Οι εταιρείες αυτές όπως και πολλές άλλες συγκέντρωσαν την προσοχή στις πωλήσεις αγαθών μέσω του διαδικτύου.

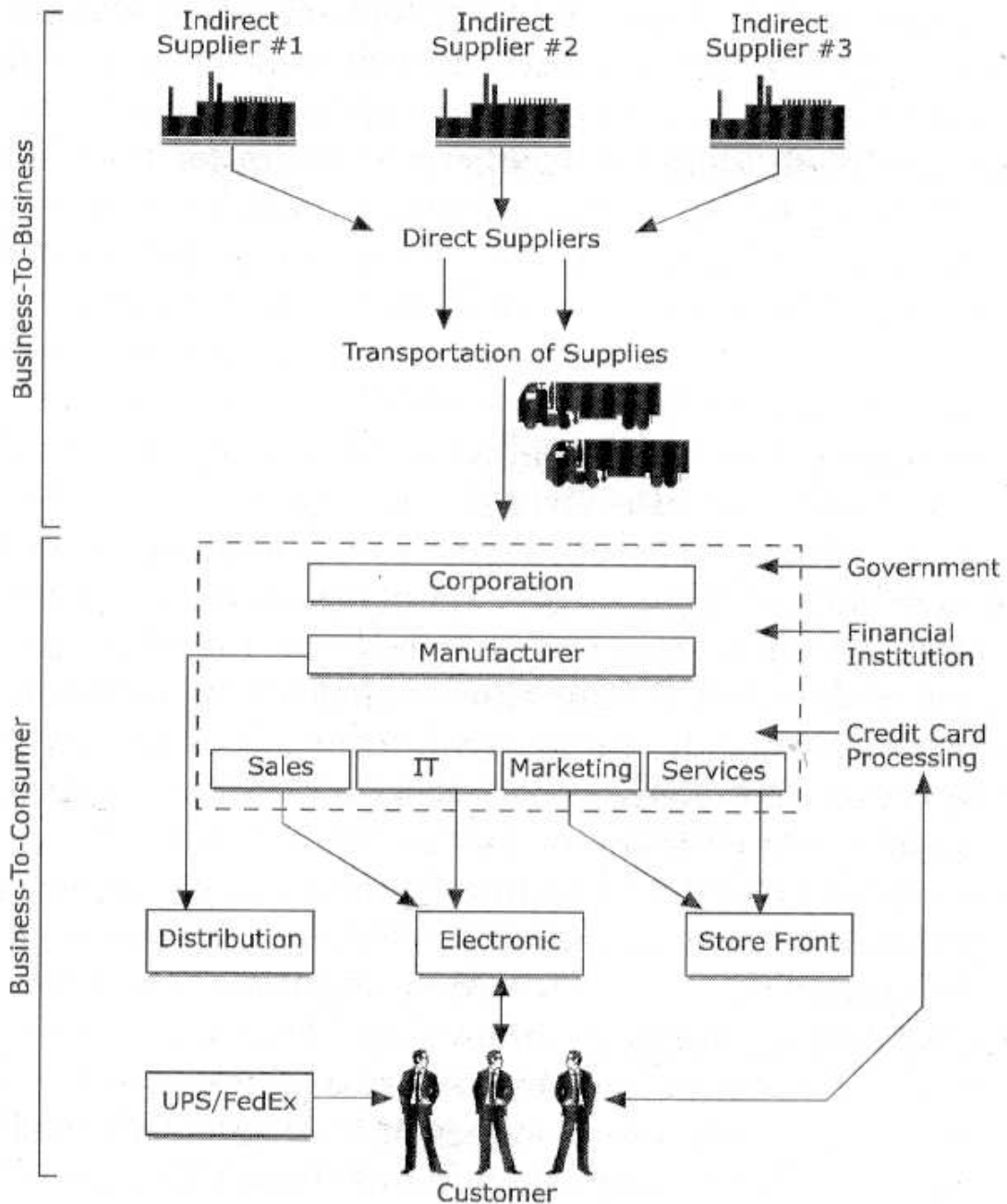
Τελικά όπως το σκεφτόμαστε, κάθε είδος προϊόντος μπορεί να συμμετέχει στις ηλεκτρονικές αγορές. Στην πράξη όμως, λίγες κατηγορίες προϊόντων και υπηρεσιών μπορούν να διακινηθούν εντελώς με ηλεκτρονικό τρόπο και μόνο μέσω του ηλεκτρονικού εμπορίου. Για τα υπόλοιπα είδη προϊόντων και υπηρεσιών στις ηλεκτρονικές αγορές μπορούν να πραγματοποιηθούν οι διαδικασίες γνωστοποίησης, ενημέρωσης, προσέλκυσης και συναλλαγής, ολοκληρώνοντας την παράδοση του προϊόντος στον αγοραστή μέσα από κάποιο φυσικό δίκτυο διανομής όπως αναφέρουν και οι ερευνητές Σιώμοκος και Τσιάμης το 2004. Στα παρακάτω διαγράμματα παρουσιάζεται, προσφέροντας μας ένα τυπικό μοντέλο ηλεκτρονικού εμπορίου προσανατολισμένο προς τον καταναλωτή:



Διάγραμμα 1 : Μοντέλο ηλεκτρονικού εμπορίου B2B (Korper S. & Ellis J. (2000))

Ήδη από αυτό το σημείο διαγράφεται καθαρά η σημαντική συνεισφορά του ηλεκτρονικού εμπορίου για την πώληση των αγαθών και υπηρεσιών. Μέσα από την ψηφιακή τεχνολογία, το αγοραστικό κοινό όλου του πλανήτη έχει την ευκαιρία για αγορά μεγαλύτερης ποικιλίας προϊόντων σε χαμηλότερες τιμές απ' ότι έχει στις φυσικές-παραδοσιακές αγορές. Φυσικά, η επικοινωνία και οι σχέσεις που δημιουργούνται μέσω του e-commerce είναι διαδραστικές και από τους καταναλωτές αλλά και από τους επιχειρηματίες, και έτσι και οι επιχειρηματίες μπορούν να

προσφέρουν γρηγορότερες και καλύτερες υπηρεσίες, επιτρέποντας στους καταναλωτές την αγορά αγαθών από την καρέκλα του γραφείου τους μέχρι την αγορά του αυτοκινήτου τους, κερδίζοντας και οι δύο πλευρές τα δύο πιο πολύτιμα και δυσεύρετα πράγματα της εποχής μας, **χρήμα** και **χρόνο**.



Διάγραμμα 2 : Πιο λεπτομερές διάγραμμα για το Μοντέλο ηλεκτρονικού εμπορίου B2B (Korper S. & Ellis J. (2000))

Απ' την άλλη πλευρά για τις μικρομεσαίες επιχειρήσεις αναδύονται νέες επιχειρηματικές ευκαιρίες που στο παρελθόν φάνταζαν ουτοπικές, διότι τώρα έχουν την ευκαιρία διάθεσης των αγαθών τους 24 ώρες το 24ωρο, 7 μέρες την εβδομάδα, 365 μέρες το χρόνο (24X7X365), παράλληλα μειώνουν το κόστος που σχετίζεται με αποθήκευση, προσωπικό, προμηθευτές, και, φυσικά αυξάνουν το μερίδιο τους στην αγορά.

3. Ηλεκτρονικό εμπόριο εντός της επιχείρησης (Intra-organisational ecommerce). Αυτή είναι μία άλλη μορφή ηλεκτρονικού εμπορίου που στην ουσία αφορά το ενδοδίκτυο (που είναι το εσωτερικό Διαδίκτυο ή όπως αλλιώς λέγεται Intranet). Αποσκοπεί στη χρήση εφαρμογών που έχουν σχέση με ηλεκτρονικό εμπόριο εσωτερικά σε μια επιχείρηση για την αποτελεσματικότερη λειτουργία των δραστηριοτήτων της, με σκοπό την παροχή καλύτερων προϊόντων και υπηρεσιών, από πλευράς επιχείρησης, στους πελάτες. Οι εφαρμογές που συνήθως εντάσσονται στη λειτουργία ενός ενδοδικτύου είναι οι εξής:

- **Επικοινωνία μεταξύ ομάδων εργασίας** (π.χ. e-mail, κλειστές ομάδες συζήτησης, βίντεο-συνδιασκέψεις).
- **Ηλεκτρονική δημοσίευση** (π.χ. τυπικές λειτουργικές διαδικασίες, προδιαγραφές προϊόντων, χρονοδιαγράμματα εργασιών και συναντήσεων).
- **Παραγωγικότητα πωλήσεων** (π.χ. καλύτερη ροή πληροφοριών ανάμεσα στην επιχείρηση και το σύνολο των πωλητών της καθώς επίσης ανάμεσα στην επιχείρηση και τους πελάτες με σκοπό την καλύτερη ενημέρωση και γνώση των στελεχών)

ΟΙ ΝΕΕΣ ΚΑΤΗΓΟΡΙΕΣ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ

Τα τελευταία χρόνια δημιουργήθηκε η ανάγκη για μεγαλύτερο επίπεδο εκλέπτυνσης των μορφών του ηλεκτρονικού εμπορίου, με αποτέλεσμα να εμφανιστούν νέες κατηγορίες, με πιο αντιπροσωπευτικές τις παρακάτω:

1. Consumer-to-consumer ή C2C: Αυτή η περίπτωση αφορά ιδιώτες που αγοράζουν και προϊόντα μεταξύ τους (π.χ. όπως η ιστοσελίδα του e-bay). Το e-bay είναι μια ιστοσελίδα που ανήκει στην κατηγορία των ιστοσελίδων δημοπρασιών (που λέγονται και auction sites). Το χαρακτηριστικό στοιχείο των auction sites είναι η δυναμική

τιμολόγηση των προϊόντων/υπηρεσιών (dynamic pricing), όπου οι τιμές δεν είναι προκαθορισμένες, αλλά ρυθμίζονται με βάση την προσφορά και τη ζήτηση ανά χρονική στιγμή.

2. **Consumer-to-business ή C2B:** Αυτού του είδους οι εφαρμογές συχνά αναφέρονται και σαν **reserve auction**, καθώς επιτρέπουν στον αγοραστή να ονοματίσει τη δική του τιμή για ένα συγκεκριμένο προϊόν ή υπηρεσία και να ζητήσει προσφορές από ένα σύνολο πωλητών. Η τιμή αυτή ονομάζεται **demand bid (απαιτούμενη προσφορά)** και είναι συνήθως δεσμευτική για τον υποψήφιο αγοραστή. Εν συνεχεία το ηλεκτρονικό κατάστημα στέλνει το demand bid στους συμμετέχοντες πωλητές, ώστε να κάνουν τις προσφορές τους (π.χ. η ιστοσελίδα priceline.com)

3. **Business-to-government ή B2G:** Αυτή η κατηγορία αφορά όλες τις επιχειρηματικές συναλλαγές με διάφορους κυβερνητικούς φορείς και οργανισμούς και ίσως και δημόσιες υπηρεσίες, όπως για παράδειγμα η καταβολή φόρων, η συμπλήρωση εγγράφων, η υποβολή αιτήσεων (Φ.Π.Α., ΚΒΣ) κ.λπ.

4. **Government-to-consumers ή G2C:** Αυτή η κατηγορία ηλεκτρονικού εμπορίου αναφέρεται σε δοσοληψίες πολιτών με κυβερνητικούς φορείς (π.χ. ιστοσελίδα Κέντρων Εξυπηρέτησης Πολιτών).

ΤΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ

Τα πλεονεκτήματα από την υιοθέτηση και χρήση του ηλεκτρονικού εμπορίου και του ηλεκτρονικού επιχειρείν γενικότερα παρουσιάζουν ιδιαίτερη ποικιλομορφία και διαφορετικότητα μεταξύ τους διότι διέπουν όλες τις επιχειρηματικές διεργασίες και λειτουργίες μιας επιχείρησης. Παρακάτω ομαδοποιήσαμε και ταξινομήσαμε τα πλεονεκτήματα που αναφέρονται στην βιβλιογραφία σε τέσσερις γενικές κατηγορίες ανάλογα με την προσφορά τους στην επιχείρηση όπως τα αναφέρει και ο Shaw και οι υπόλοιποι συγγραφείς (βλέπε στη βιβλιογραφία για Shaw 2000)

1. Βελτίωση της εσωτερικής λειτουργίας της επιχείρησης: Το ηλεκτρονικό εμπόριο συμβάλλει στην βελτίωση των εσωτερικών διαδικασιών και λειτουργιών μιας επιχείρησης. Πιο συγκεκριμένα:

- Μείωση του κόστους δημιουργίας, επεξεργασίας, αποθήκευσης, διανομής και ανάκτησης έντυπων πληροφοριών
- Αύξηση της παραγωγικότητας
- Μείωση άμεσου κόστους (π.χ. τηλεπικοινωνίες)
- Βελτίωση της δημόσιας εικόνας μίας επιχείρησης
- Απλοποίηση εσωτερικών διαδικασιών
- Διατήρηση χαμηλών αποθεμάτων και γενικών βιομηχανικών εξόδων με την εισαγωγή της διοίκησης εφοδιαστικής αλυσίδας τύπου pull (Just In Time)

2. Βελτίωση της εξωτερικής λειτουργίας της επιχείρησης: Το ηλεκτρονικό εμπόριο συμβάλλει στην βελτίωση των εξωτερικών διαδικασιών και λειτουργιών μιας επιχείρησης (π.χ. πελάτες, προμηθευτές, ανταγωνιστές κ.λπ.). Πιο συγκεκριμένα:

- Χαμηλότερες τιμές προϊόντων υπηρεσιών
- Εξάπλωση της αγοράς σε εθνικό και παγκόσμιο επίπεδο
- Αποδοτικότερη διαχείριση του ανταγωνισμού
- Προώθηση προϊόντων-υπηρεσιών
- Βελτίωση του επιπέδου εξυπηρέτησης των πελατών

3. Μετασχηματισμός των επιχειρήσεων. Το ηλεκτρονικό εμπόριο προσφέρει την δυνατότητα για ουσιαστικό μετασχηματισμό των επιχειρήσεων, μέσω της

- Υιοθέτησης των νέων τεχνολογιών
- Δημιουργίας νέων σχέσεων με τους πελάτες
- Δημιουργία νέων συνεργασιών

4. Δημιουργία νέων μοντέλων: Οι βελτιώσεις και οι μετασχηματισμοί που είδαμε παραπάνω αφορούν μικρές ή μεγάλες αλλαγές μεμονωμένων δραστηριοτήτων μιας επιχείρησης. Η δημιουργία νέων μοντέλων αποτελεί μια εντελώς διαφορετική κατηγορία αλλαγών, που οδηγούν σε νέα προϊόντα, υπηρεσίες και λειτουργικές δομές.

- Νέες δυνατότητες δημιουργίας νέων προϊόντων και υπηρεσιών
- Νέες δυνατότητες διαφοροποίησης των υφιστάμενων προϊόντων και υπηρεσιών
- Νέα μοντέλα λειτουργίας για κάθε τύπο επιχείρησης

ΤΑ ΜΕΙΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ

Στις προηγούμενες ενότητες αναφέραμε διάφορα και αρκετά αλλά και σημαντικά πλεονεκτήματα που μπορεί να προσφέρει το ηλεκτρονικό εμπόριο και το ηλεκτρονικό επιχειρείν σε μια επιχείρηση. Αν έχουμε ως στόχο όμως να το υιοθετήσουμε για παράδειγμα στην επιχείρησή μας, είναι απαραίτητο να γνωρίζουμε και τα διάφορα προβλήματα που δημιουργούνται από την υιοθέτηση και χρήση αυτών των ηλεκτρονικών εργαλείων. Υπάρχει ένα κείμενο των ειδικών της Ευρωπαϊκής Ένωσης που δημοσιεύθηκε στο www.elexportal.com, τον Ιούλιο του 2003 για τα διάφορα προβλήματα και τους πολλούς αρνητικούς παράγοντες που επηρεάζουν τη συμμετοχή των πολιτών και των αγοραστών στο ηλεκτρονικό εμπόριο και στο e-business γενικά και αυτοί οι παράγοντες είναι οι παρακάτω:

- Υπάρχει τεράστια έλλειψη ενημέρωσης για τα ρίσκα και τα οφέλη που προσφέρει το ηλεκτρονικό εμπόριο: Συνήθως οι άνθρωποι είτε δεν έχουν υπολογιστές στα σπίτια τους, είτε δεν διαθέτουν το χρόνο για την αναζήτηση πληροφοριών. Επίσης και για τους αγοραστές αλλά και για τους πωλητές συχνά, πολύ απλά δεν γνωρίζουν καν εάν το ηλεκτρονικό-επιχειρείν τους είναι επωφελές.
- Υπάρχει μεγάλη δυσκολία στην επιλογή της κατάλληλης πλατφόρμας B2B ηλεκτρονικού εμπορίου που να ανταποκρίνεται στις ανάγκες της συγκεκριμένης επιχείρησης που θέλει να οργανωθεί και να κάνει συναλλαγές με μια άλλη επιχείρηση.
- Υπάρχουν διαφορετικά και αλληλοσυγκρουόμενα τεχνικά πρότυπα. Συχνά οι μικρομεσαίες επιχειρήσεις αδυνατούν να ακολουθήσουν τη διαδικασία προτυποποίησης των προϊόντων τους που απαιτείται για τη συμμετοχή στις ηλεκτρονικές αγορές.
- Επίσης υπάρχουν πολλοί κίνδυνοι που προκύπτουν από άγνοια των κανόνων της αγοράς, των εμπορικών εταίρων και πρακτικών: Οι παλαιές και παραδοσιακές επιχειρήσεις στην Ελλάδα αλλά και σε άλλα κράτη του εξωτερικού συχνά δεν γνωρίζουν πως δουλεύουν οι διάφορες ηλεκτρονικές αγορές οι οποίες διέπονται από τους δικούς τους κανόνες για θέματα όπως πχ οι ηλεκτρονικές δημοπρασίες. Οι μικρομεσαίες επιχειρήσεις για να

ελαχιστοποιήσουν τους κινδύνους πρέπει να κατανοήσουν πλήρως αυτούς τους κανόνες και να καταρτίσουν νέες επιχειρηματικές στρατηγικές.

- Υπάρχουν επίσης και διάφοροι άλλοι οικονομικοί παράγοντες όπως για παράδειγμα το κόστος διατήρησης πληροφοριακών συστημάτων και τα χρήματα που χρειάζονται να επενδύσουν για την συμμετοχή τους στο διαδίκτυο. Συνήθως πρέπει να έχουν έναν υπάλληλο που θα φροντίζει για την ιστοσελίδα για την αλλαγή τιμών στα προϊόντα κλπ.
- Γενικά υπάρχει έλλειψη ειδικευμένου προσωπικού.

Συμπερασματικά, οι προκλήσεις είναι πραγματικά πολλές και σημαντικές για τις μικρομεσαίες επιχειρήσεις. Οι περισσότερες από αυτές μπορούν να αντιμετωπισθούν παρέχοντας στις διάφορες μικρομεσαίες επιχειρήσεις ακριβείς, αξιόπιστες, κατάλληλες ανά τομέα και επίκαιρες πληροφορίες. Τελικά το κύριο ερώτημα που πρέπει να απαντηθεί είναι το αν θα είναι επικερδές για την επιχείρησή το ηλεκτρονικό εμπόριο. Οι επιφυλάξεις που έχουν οι διάφορες επιχειρήσεις για το ηλεκτρονικό εμπόριο θα πρέπει να τις ξεπεράσουν, αλλά για να γίνει αυτό πρέπει να πεισθούν πως αξίζει τον κόπο να προβούν σε περαιτέρω επενδύσεις και στις απαραίτητες οργανωτικές αλλαγές.

ΚΕΦΑΛΑΙΟ ΙΙ: ΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ ΚΑΙ ΤΑ ΜΟΝΤΕΛΑ ΤΟΥ

ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο θα μιλήσουμε και θα κάνουμε μια αρκετά λεπτομερή επισκόπηση για τα διάφορα μοντέλα του ηλεκτρονικού εμπορίου όσον αφορά τον τρόπο με τον οποίο προσπαθούν να πουλήσουν οι διάφορες επιχειρήσεις τα προϊόντα τους και να κάνουν περισσότερα κέρδη. Θα αναφερθούμε λοιπόν σε ένα προς ένα στα παρακάτω μοντέλα:

- Ηλεκτρονικό κατάστημα
- Μοντέλο ηλεκτρονικής προμήθειας
- Καταστήματα δημοπρασιών
- Ηλεκτρονικό πολυκατάστημα
- Οι ηλεκτρονικές πύλες και τα
- Μοντέλα δυναμικής διαμόρφωσης τιμών

ΤΑ ΠΛΕΟΝΕΚΤΗΜΑΤΑ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΜΑΡΚΕΤΙΝΓΚ

Όσον αφορά τα διάφορα μοντέλα που θα περιγράψουμε υπάρχουν γενικά αρκετά πλεονεκτήματα και οφέλη για τις εταιρίες ειδικά για την περίπτωση του ηλεκτρονικού μάρκετινγκ. Τα κυριότερα από αυτά είναι:

- Τόπος-Διανομή (place) : Δυνατότητα αγοράς 24/7, κάλυψη απομακρυσμένων περιοχών, ταχύτητα διανομής.
- Τιμή (price) : Δυνατότητα εξατομίκευσης-προσωποποίησης της τιμολογιακής πολιτικής στα ιδιαίτερα χαρακτηριστικά, προτιμήσεις και προφίλ των πελατών (συμφωνίες τιμολόγησης, εκπτώσεις, δυναμική τιμολόγηση)
- Προϊόν (product) : Δυνατότητα εξατομίκευσης - προσωποποίησης προϊόντος στα ιδιαίτερα χαρακτηριστικά των πελατών (ποιότητα, συσκευασία, brand)
- Προώθηση (promotion) : Δυνατότητα εξατομικευμένων μηνυμάτων επικοινωνίας και περιεχομένου της ιστοσελίδας (διαφήμιση – πωλήσεις)

ΤΟ ΜΟΝΤΕΛΟ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ

Το μοντέλο του ηλεκτρονικού καταστήματος ονομάζεται επίσης και Storefront model, δηλαδή συνήθως αντιστοιχεί σε μια εταιρία ή επιχείρηση η οποία έχει παραδοσιακά καταστήματα αλλά επίσης έχει επεκταθεί και στον τομέα του ηλεκτρονικού εμπορίου. Το βασικότερο από τα πλεονεκτήματα μιας τέτοιας επιχειρηματικής κίνησης είναι ότι μπορεί πλέον η εταιρία αυτή να κάνει δικτυακή προβολή των προϊόντων της. Από την άλλη μεριά όλες οι παραδοσιακές λειτουργίες που είχε η εταιρία όπως για παράδειγμα η πολιτική του μαρκετινγκ που ασκούσε η εταιρία αυτή πρέπει πλέον να γίνονται και αυτές μέσω του διαδικτύου. Όμως η εταιρία μπορεί να αποκτήσει πάρα πολλά οφέλη μέσα από αυτή τη διαδικασία. Τα κυριότερα οφέλη είναι τα εξής:

- Αυξημένη ζήτηση : δηλαδή ο τζίρος της εταιρίας επεκτείνεται μέσω της αύξησης πωλήσεων μέσω του διαδικτύου.
- Παγκόσμια παρουσία: Πριν την επέκταση της εταιρίας στο διαδίκτυο, αυτή ήταν γνωστή σε μια μικρή ομάδα πελατών που ενδιαφέρονταν για τα προϊόντα της εταιρίας ή που την γνώρισαν μέσω της άμεσης διαφήμισης. Τώρα μέσω του διαδικτύου μπορεί να γίνει γνωστή σε όλο τον κόσμο ακόμη και παγκόσμια.
- Χαμηλό κόστος: Το κόστος της συντήρησης του διαδικτυακού τόπου της εταιρίας σχετικά με τα αντίστοιχα οφέλη της εταιρίας με την αύξηση των πωλήσεων της μέσω του διαδικτύου είναι σχεδόν μηδαμινά.
- Μείωση εξόδων για διαφήμιση/πωλητές: Η εταιρία πλέον μπορεί να διαφημιστεί μόνη της μέσα από τον δικό της διαδικτυακό τόπο χωρίς να χρειάζεται το επιπρόσθετο έξοδο των διαφημιστών που αναλαμβάνουν μια τέτοια λειτουργία.

Από την άλλη μεριά επίσης υπάρχουν πάρα πολλά οφέλη για τους πελάτες που αγοράζουν προϊόντα από εταιρίες που υπάρχουν στο διαδίκτυο. Τα κυριότερα οφέλη αφορούν κυρίως στο ότι ο καταναλωτής μπορεί να απολαμβάνει χαμηλότερες τιμές (λόγω της μείωσης του λειτουργικού κόστους της εταιρίας) και κατά δεύτερο λόγο ότι μπορεί να αγοράσει προϊόντα από την εταιρία οποιαδήποτε ώρα θέλει χωρίς να υπόκειται στο παραδοσιακό ωράριο λειτουργίας της αγοράς

ΤΟ ΜΟΝΤΕΛΟ ΤΗΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΠΡΟΜΗΘΕΙΑΣ

Το επιχειρηματικό αυτό μοντέλο σχετίζεται στην ηλεκτρονική προσφορά προμηθειών και τον εφοδιασμό αγαθών και υπηρεσιών. Η ηλεκτρονική προσφορά και προμήθεια αγαθών και υπηρεσιών είναι μία υπηρεσία χρήσιμη για τις μεγάλες επιχειρήσεις και τις δημόσιες αρχές. Στα οφέλη της συγκαταλέγονται οι περισσότερες επιλογές προμηθευτών (και πιθανόν χαμηλότερο κόστος), η καλύτερη ποιότητα, οι βελτιωμένες μέθοδοι παράδοσης, το μειωμένο κόστος προμήθειας. Τα οφέλη για τους προμηθευτές είναι περισσότερες ευκαιρίες για υποβολή προσφορών (σε παγκόσμια κλίμακα), χαμηλότερο κόστος υποβολής μιας προσφοράς, συνεταιρικές προσφορές.

Σύμφωνα με αυτό το μοντέλο ο ενδιαφερόμενος «βγάζει στον αέρα» έναν διαγωνισμό. Μαζί με τον διαγωνισμό βγαίνουν επίσης και όλες οι λεπτομέρειες και οι διάφοροι περιοριστικοί όροι που αφορούν στον διαγωνισμό αυτό. Ο διαγωνισμός μπορεί να είναι πλειοδοτικός ή μειοδοτικός. Επίσης πρέπει να πούμε ότι υπάρχουν αυστηροί περιοριστικοί όροι όσον αφορά την ημερομηνία της λήξης κατάθεσης των διαφόρων προσφορών. Υπάρχουν πολλά ωφέλη τόσο για τους ενδιαφερόμενους που διοργανώνουν ένα διαγωνισμό όσο και για τις διάφορες εταιρίες που συμμετέχουν σε τέτοιους διαγωνισμούς. Αυτά τα οφέλη συνοψίζονται παρακάτω:

- Οφέλη του ενδιαφερομένου:
 - Δυνατότητα μεγαλύτερης εκλογής από προμηθευτές
 - Μείωση του κόστους διαχείρισης
 - Βελτίωση της ποιότητας/πιστότητας του διαγωνισμού
 - Μειωμένο κόστος προμηθειών
- Οφέλη των προμηθευτών:
 - Μεγαλύτερες ευκαιρίες διαγωνισμών
 - Βελτίωση των κερδών

Από την άλλη μεριά αυτό το μοντέλο διέπεται από κάποιες βασικές λειτουργίες οι οποίες είναι οι εξής:

Βασικές λειτουργίες του μοντέλου ηλεκτρονικής προμήθειας

- Ηλεκτρονική παρουσίαση των προϊόντων της προμήθειας
- Ηλεκτρονική παρουσίαση των όρων της προκήρυξης
- Ηλεκτρονική λήψη προσφορών
- Ηλεκτρονική μεθοδολογία αξιολόγησης προσφορών
- Ηλεκτρονική διαχείριση παραγγελιών

- Ηλεκτρονική διαχείριση πληροφοριών

Μια από τις γνωστές εταιρίες ηλεκτρονικής προμήθειας είναι και η Ariba της οποίας τον ιστότοπο παρουσιάζουμε στο επόμενο σχήμα

A R I B A®

Buyer Solutions Supplier Solutions Customers Partners Company Support

In this section: Overview | Expertise | Visibility | Sourcing | Contract Management | Procurement |

Spend Management Overview

As competition in the procurement arena intensifies, it is critical that organizations dramatically improve their effectiveness by managing spend and accelerating process performance. Leading companies understand that this can only be achieved through a comprehensive spend management solution—one that delivers cross-company spend visibility, promotes accurate decision making, and employs best practices on which to refine and grow an efficient procurement infrastructure.

Ariba Spend Management Solutions

Visibility

Sourcing

Contract Management

Procurement and Expense

Invoice and Payment

Supplier Management

SPEND MANAGEMENT

Ariba Spend Manage

ΤΟ ΜΟΝΤΕΛΟ ΤΗΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΔΗΜΟΠΡΑΣΙΑΣ

Με τον όρο ηλεκτρονική δημοπρασία (e-auction) εννοούμε την δημιουργία δικτυακών τόπων κατάλληλων για τη διεξαγωγή δημοπρασιών, χωρίς φυσική παρουσία των συμμετεχόντων και με την αξιοποίηση του διαδικτύου, ιδιωτικών ή ιδεατά ιδιωτικών δικτύων. Στις πιο σύγχρονες υλοποιήσεις του μοντέλου η ηλεκτρονική δημοπρασία μπορεί να είναι μια υπηρεσία που επιτρέπει στους χρήστες της να εκθέτουν τα προϊόντα που επιθυμούν προς δημοπράτηση μέσω του Διαδικτύου. Στις περισσότερο διαδεδομένες από αυτές τις υλοποιήσεις, ο διαπιστευμένος χρήστης της υπηρεσίας μπορεί, με πάρα πολύ απλό τρόπο, να προσθέσει στη Web σελίδα του μια υπηρεσία η οποία αναλαμβάνει να εμφανίζει στους επισκέπτες της σελίδας τα προϊόντα του, να δέχεται προσφορές για αυτά και να κρατάει στατιστικά στοιχεία για την κίνηση των προϊόντων και για τους υποψήφιους αγοραστές.

Το μοντέλο αυτό απευθύνεται σε οποιονδήποτε επιθυμεί να βρει αγοραστές για τα προϊόντα του μέσω του Διαδικτύου. Το μόνο που απαιτείται από πλευράς χρήστη είναι να έχει μια Web σελίδα η οποία θα φιλοξενεί την κατάλληλη υπηρεσία. Κατά τα

άλλα η χρήση της υπηρεσίας δεν απαιτεί καθόλου ιδιαίτερες γνώσεις καθώς όλα γίνονται μέσω φιλικών προς τον χρήστη οδηγών (wizards). Αυτό είναι και το μεγάλο πλεονέκτημα της αφού επιτρέπει σε χρήστες χωρίς προγραμματιστικές γνώσεις να απευθυνθούν στο «ηλεκτρονικό» καταναλωτικό κοινό.

Ανεξάρτητα με το αν μιλάμε για το απλοϊκό ή το εξελιγμένο μοντέλο ηλεκτρονικής δημοπρασίας, οι σημαντικότερες και ταυτόχρονα αναγκαίες ενότητες για τη δημιουργία ενός συστήματος για τη διεξαγωγή ηλεκτρονικών δημοπρασιών είναι: ο κατάλογος δημοπρατούμενων προϊόντων, η εγγραφή μέλους, η πιστοποίηση (verification), η υποστήριξη, η επικοινωνία με το σύστημα δημοπρασίας, το περιβάλλον διεπαφής, η παραγγελία και πληρωμή, η παρουσίαση πληροφοριών δημοπρασίας, η παροχή λίστας συχνών ερωτήσεων (FAQ), η επικοινωνία μεταξύ των πελατών μέσα από Chat rooms, η παροχή δυνατότητας αυτόματης προσφοράς και η διανομή των προϊόντων. Οι ενότητες αυτές αναλύονται στη συνέχεια.

Κατάλογος δημοπρατούμενων προϊόντων

Η επιχείρηση έχει τη δυνατότητα να παρουσιάσει στον καταναλωτή μια λίστα με τα προϊόντα ή τις υπηρεσίες που προσφέρει, διευκολύνοντάς τον και παρέχοντάς του άμεσα την πληροφορία για τα προϊόντα της, χωρίς να τον αναγκάσει να χρησιμοποιήσει κάποια μηχανή αναζήτησης. Τα προϊόντα ωφέλιμο είναι να είναι σωστά κατηγοριοποιημένα, να συνοδεύονται από τιμές, ακριβή χαρακτηριστικά, σύντομη αλλά και εκτενή περιγραφή καθώς και σχόλια και κριτικές από άλλους πελάτες που έχουν αγοράσει το προϊόν ή έχουν χρησιμοποιήσει την υπηρεσία. Οι πληροφορίες για τα προσφερόμενα αγαθά αναγκαίο είναι να χαρακτηρίζονται από επάρκεια, έτσι ώστε ο καταναλωτής να παίρνει πλήρη εικόνα γι' αυτά και άρα η απόφαση του να είναι σίγουρη.

Εγγραφή μέλους

Οι πελάτες που επισκέπτονται μια ηλεκτρονική δημοπρασία στο διαδίκτυο είναι είτε περαστικοί, οι οποίοι δεν έχουν δικαίωμα να συμμετέχουν ενεργά, είτε «γνωστοί» (καταχωρημένοι). Η καταχώρηση των χρηστών γίνεται συνήθως με τη συμπλήρωση

μιας ηλεκτρονικής φόρμας στην οποία ο χρήστης προσδιορίζει συγκεκριμένα υποχρεωτικά στοιχεία. Ορισμένα από αυτά είναι: το όνομα, το επώνυμο, η ηλικία και το e-mail του. Στη συνέχεια, η εταιρία που είναι υπεύθυνη για τη διεξαγωγή της ηλεκτρονικής δημοπρασίας, στέλνει στο e-mail του χρήστη την επιβεβαίωση για την εγγραφή του, γνωστοποιώντας του επίσης το usemame και το password που θα χρησιμοποιεί ο χρήστης στο εξής. Σε αρκετές περιπτώσεις ο χρήστης έχει τη δυνατότητα να διαλέξει μόνος του το usemame και το password.

Υποστήριξη

Η υποστήριξη είναι απαραίτητη προϋπόθεση για την επιτυχία ενός συστήματος ηλεκτρονικής δημοπρασίας. Πρώτον, γιατί οι πελάτες νοιώθουν ανασφάλεια από την απουσία φυσικών προσώπων, στους οποίους μπορούν να απευθύνουν τα ερωτήματά τους και δεύτερον γιατί η υποστήριξη οδηγεί σε ευχαριστημένους πελάτες με προοπτικές να επαναλάβουν τις αγορές τους.

Επικοινωνία με το σύστημα δημοπρασίας

Το ηλεκτρονικό ταχυδρομείο χρησιμοποιείται για να προσφέρει την αμεσότητα που λείπει από μια απρόσωπη σχέση μεταξύ πελάτη και ηλεκτρονικού συστήματος δημοπρασίας. Αυτή η μορφή επικοινωνίας σε κάποιες περιπτώσεις προϋποθέτει τη δέσμευση υπαλλήλων που εξυπηρετούν τους πελάτες όταν οι τελευταίοι ζητούν να λάβουν πληροφορίες σχετικά με κάποιο ζήτημα που τους απασχολεί. Σε κάποιες άλλες περιπτώσεις η επικοινωνία είναι στερεότυπη. Για παράδειγμα, όταν ολοκληρωθεί μια ηλεκτρονική δημοπρασία είναι σκόπιμο, κάθε πελάτης να λαμβάνει ένα ηλεκτρονικό μήνυμα με περιεχόμενο τα στοιχεία του είδους που ο ίδιος «χτύπησε» (σκοπός του μηνύματος αυτού είναι η επιβεβαίωση του αποτελέσματος της δημοπρασίας). Στην περίπτωση αυτή, το ηλεκτρονικό μήνυμα δεν πρέπει να συντάσσεται χειρωνακτικά αλλά να συντίθεται και να αποστέλλεται αυτόματα από το web serVer που στηρίζει το σύστημα ηλεκτρονικής δημοπρασίας.

Σχεδίαση του περιβάλλοντος διεπαφής

Κατά την υλοποίηση του περιβάλλοντος διεπαφής πρέπει να ακολουθούνται κάποιες

αρχές έτσι ώστε να διευκολύνεται η πλοήγηση των χρηστών μέσα στις ιστοσελίδες της επιχείρησης e-auction αλλά και η διεκπεραίωση της δημοπρασίας τους. Πρέπει να γίνεται όσο το δυνατόν αναλυτικότερη περιγραφή των προϊόντων και επεξήγηση τεχνικών όρων. Στα συστήματα e-auction, ακριβώς επειδή δεν υπάρχει η δυνατότητα άμεσης επικοινωνίας με το προσωπικό της επιχείρησης, στο οποίο θα μπορούσε ο πελάτης να απευθυνθεί για ερωτήσεις, είναι αναγκαίο το περιεχόμενο των σελίδων να είναι επαρκώς περιγραφικό. Πρέπει κάθε προϊόν να συνοδεύεται με ανάλογη εικόνα, ενώ οι τεχνικοί όροι θα πρέπει αν εξηγούνται πιθανόν με την παραπομπή σε ένα γλωσσάριο. Συνήθως οι δικτυακές επιχειρήσεις μοιάζουν να είναι αχανείς και ακόμα και αν συμμορφώνονται με τις σχεδιαστικές αρχές, δεν είναι λίγοι αυτοί οι χρήστες που δεν μπορούν να βρουν εύκολα αυτό που θέλουν. Επιβάλλεται σε αυτές τις περιπτώσεις η ύπαρξη δυνατότητας αναζήτησης στον ιστότοπο. Επίσης οφείλει η δυνατότητα αναζήτησης να είναι εμφανής από όλες τις ιστοσελίδες της e-auction επιχείρησης και όχι μόνο από την αρχική. Επίσης τα αποτελέσματα της αναζήτησης πρέπει να είναι περιγραφικά και καθοδηγητικά ιδίως σε περιπτώσεις λάθους.

Παραγγελία και πληρωμή

Όταν ο πελάτης κερδίζει τη δημοπρασία του προϊόντος που τον ενδιαφέρει, ολοκληρώνει τη συναλλαγή συμπληρώνοντας τα προσωπικά του στοιχεία μαζί με τον τόπο προορισμού και τον τρόπο αποστολής και διάθεσης που επιθυμεί και καταβάλλει το αντίτιμο με έναν από τους υπάρχοντες-ηλεκτρονικούς τρόπους πληρωμής που διαθέτει η επιχείρηση e-auction. Οι μέθοδοι πληρωμής είναι είτε με αντικαταβολή είτε, κυρίως, με ηλεκτρονική πληρωμή. Η ηλεκτρονική πληρωμή περιλαμβάνει τις πιστωτικές κάρτες (όπου ο καταναλωτής καλείται να επιδείξει την ικανότητα του να πληρώσει, παρουσιάζοντας τον αριθμό της πιστωτικής του κάρτας στην επιχείρηση e-auction) και τις ηλεκτρονικές επιταγές που στην ουσία είναι ένα μήνυμα προς την τράπεζα του καταναλωτή να μεταφέρει κεφάλαιο από το λογαριασμό του στο λογαριασμό της e-auction επιχείρησης. Το μήνυμα αυτό δεν στέλνεται απευθείας στην τράπεζα, αλλά στο μελλοντικό αποδέκτη του κεφαλαίου, ο οποίος πρέπει να εμφανίσει την επιταγή αυτή στην τράπεζα προκειμένου να εισπράξει το ποσό του κεφαλαίου που αναγράφεται. Αφού λάβει χώρα η μεταφορά κεφαλαίου, η επικυρωμένη και εξοφλημένη επιταγή επιστρέφει στον αποστολέα, και μπορεί έτσι να χρησιμοποιηθεί ως απόδειξη της πληρωμής. Η τρίτη μέθοδος ηλεκτρονικής

πληρωμής είναι το ψηφιακό χρήμα.

Πληροφορίες δημοπρασίας

Περιλαμβάνουν όλες της απαραίτητες πληροφορίες που πρέπει να γνωρίζει ο χρήστης προκειμένου να πάρει μέρος σε μια ηλεκτρονική δημοπρασία. Έτσι λοιπόν θα πρέπει να γνωρίζει τουλάχιστον την αρχική τιμή, την τελική (επιδιωκόμενη) τιμή και τη διάρκεια της ηλεκτρονικής δημοπρασίας.

Λίστα Συχνών Ερωτήσεων (FAQ)

Τα ερωτήματα των πελατών επαναλαμβάνονται με μεγάλη συχνότητα. Οι λίστες συχνών ερωτήσεων (FAQ) αποτελούνται από έναν κατάλογο ερωτήσεων που έχουν καταγραφεί ως ερωτήσεις, οι οποίες υποβάλλονται συχνά και συνοδεύονται από κατατοπιστικές απαντήσεις.

Chat rooms

Πρόκειται για χώρους διαδικτυακής συζήτησης, όπου οι πελάτες ανταλλάσσουν απόψεις και εμπειρίες σε σχέση με προϊόντα / υπηρεσίες και διαδικασίες του e-auction.

Αυτόματη προσφορά

Μπορεί ένας πελάτης (A) να θέσει ένα άνω όριο χρημάτων και όταν κάποιος άλλος πελάτης (B) χτυπήσει ένα προϊόν που ενδιαφέρει τον A, τότε αυτόματα το σύστημα να υποβάλει μία νέα πρόσφορα για λογαριασμό του A, εφόσον η τελική τιμή δεν ξεπερνά το άνω όριο.

Διανομή

Η διανομή των προϊόντων μπορεί να γίνει είτε φυσικά, είτε με ηλεκτρονικό τρόπο. Στις ηλεκτρονικές δημοπρασίες χρησιμοποιείται κυρίως ο πρώτος τρόπος. Σε αυτήν χρησιμοποιούνται όλοι οι υπάρχοντες τρόποι όπως το ταχυδρομείο (air-mail,

συστημένο, απλό), η ταχυαποστολή (courier) ακόμη κάποια υπηρεσία διανομής που μπορεί να ανήκει στην e-auction επιχείρηση (εφόσον πρόκειται για μεγάλη εταιρία).

Η διαδικασία της ηλεκτρονικής δημοπρασίας

Στη συνέχεια παρουσιάζεται η διαδικασία διεξαγωγής ηλεκτρονικής δημοπρασίας. Στα βήματα που παραθέτουμε, ο όρος δημοπράτης αναφέρεται στον παροχέα και διαχειριστή του συστήματος ηλεκτρονικής δημοπρασίας και όχι στον εκάστοτε πωλητή.

Ο δημοπράτης καλεί τους πωλητές να καταθέσουν μια πρόταση δημοπρασίας, προσδιορίζοντας όλες τις αναγκαίες παραμέτρους και τους ενημερώνει για την ημέρα και ώρα ανοίγματος (έναρξης) της δημοπρασίας.

Οι αγοραστές ενημερώνονται για τις επιμέρους παραμέτρους της δημοπρασίας. Οι παράμετροι συνήθως είναι:

1. Τιμή Ανοίγματος (Opening Price)
2. Επιδιωκόμενη Τιμή (Reserved Price)
3. Βήμα δημοπρασίας (Bid Increment)
4. Διάρκεια δημοπρασίας - παράταση
5. Νόμισμα
6. Proxy Bidding (αυτόματη υποβολή προσφορών)

Ο δημοπράτης παρέχει το κατάλληλο λογισμικό και την αντίστοιχη εκπαίδευση .

Από την έναρξη και μετά, οι αγοραστές υποβάλουν τις προσφορές τους. Για να γίνουν δεκτές ελέγχεται αν αυτές τηρούν το ελάχιστο όριο διαφοράς (ή το βήμα) της μιας προσφοράς από την άλλη. Όταν παρέλθει ο χρόνος λήξης, ο δημοπράτης διακόπτει τη δημοπρασία. Η διάρκειά της ορίζεται, συνήθως, στη μισή ή μία ώρα, με δυνατότητα ολιγόλεπτων παρατάσεων.

Ο δημοπράτης έχει τη δυνατότητα να εξετάσει τις προσφορές και να τις συγκρίνει μεταξύ τους, λαμβάνοντας υπόψη και άλλες παραμέτρους πέραν της τιμής.

Ο πωλητής έχει τη δυνατότητα να ελέγξει τις προσφορές, ώστε να βεβαιωθεί για την ποιότητα, τις υπηρεσίες και τα άλλα κριτήρια αξιολόγησης της προσφοράς.

Όλοι οι συμμετέχοντες ενημερώνονται για τις εξελίξεις και τα τελικά αποτελέσματα.

Κατά τη διάρκεια της δημοπρασίας μόνο ο αριθμός των συμμετεχόντων στην δημοπρασία είναι ορατός στους συμμετέχοντες (αγοραστές και πωλητές), ενώ η

ταυτότητά τους παραμένει άγνωστη

Στους περισσότερους δικτυακούς τόπους δημοπρασιών η συμμετοχή είναι δωρεάν για τους αγοραστές. Οι πωλητές συνήθως καταβάλλουν στο δημοπράτη, ένα ποσοστό της τελικής τιμής που πέτυχαν.

Πλεονεκτήματα ηλεκτρονικών δημοπρασιών

Οι ηλεκτρονικές δημοπρασίες μπορούν να αποφέρουν τεράστια οικονομικά οφέλη τόσο τους πωλητές όσο και τους αγοραστές

Πλεονεκτήματα για τον αγοραστή

1. Βελτιωμένη προσβασιμότητα
2. Μεγαλύτερη αμεσότητα
3. Αυτόματη διαχείριση προσφορών (ανώτατο όριο δαπάνης)
4. Παράλληλο, έναντι σειριακού, μοντέλο δημοπρασίας (οικονομία χρόνου, ποικιλία προϊόντων)
5. Αωνυμία (εμπιστοσύνη, αποφυγή δημιουργίας προφίλ, κλπ)
6. Αυξημένες δυνατότητες προσφορών (χρονικά περιθώρια για έρευνα αγοράς)
7. Εργαλεία για τον εντοπισμό προϊόντων (φιλτράρισμα σύμφωνα με τα κριτήρια του χρήστη)

Πλεονεκτήματα για τον πωλητή

1. Έλλειψη άγχους με το παράλληλο, έναντι του σειριακού, μοντέλου δημοπρασίας
2. Ποσοτικοποίηση επιθυμίας καταναλωτή για αγορά με την εφαρμογή της τιμολογιακής τεχνικής "one to one pricing"
3. Συνεχής διάθεση προϊόντων
4. Συμπίεση κόστους διαδικασίας (χαρτί, γραφειοκρατικά έξοδα)

Μια από τις πιο γνωστές εταιρίες διενέργειας ηλεκτρονικών δημοπρασιών είναι και η παλαιά fleamarket.gr που μετονομάστηκε σε emarket.gr της οποίας τον ιστότοπο δείχνουμε στο παρακάτω σχήμα



Κατηγορίες

- Αθλητικά είδη (3.299)
- Ακίνητα (35)
- Βιβλία & Περιοδικά (9.054)
- Διάφορα (135)
- Δώρα & Gadgets (2.450)
- Είδη Επιχειρήσεων & Έξοπλισμός (1.127)
- Εποχιακά Είδη (1.547)
- Ερωτικά (3.146)
- Ηλεκτρονικά Είδη (173)
- Ήχος & Εικόνα (16.662)
- Κινητή & Σταθερή τηλεφωνία (6.709)
- Κοσμήματα - Ρολόγια (4.609)
- Μουσικά Όργανα - Εξοπλισμός DJ (1.192)
- Μουσική (482)
- Νομίσματα (7.636)
- Οικία & Κήπος (6.916)

Το Emarket.gr προτείνει:

<p>CALVIN KLEIN CK849S 010 - ... Από: Drees24.gr Λήξη: 09/11/2010 7,00 € Δημοπρασία</p>	<p>ΡΟΛΟΙ ΤΥΠΟΥ PANERAI PAM 17... Από: anti-radar Λήξη: 09/11/2010 300,00 € Δημοπρασία</p>
<p>GAS MASK + FILTER NBC- Ιθα... Από: ingeo2010 Λήξη: 15/11/2010 20,00 € Πώληση</p>	<p>Senseo Espresso Lavazza In... Από: MR.Gadget Λήξη: 09/11/2010 1,00 € Δημοπρασία</p>
<p>HELMET AVEX-Rally 8 Avex... Από: ingeo2010 Λήξη: 10/11/2010 75,00 € Πώληση</p>	<p>ΚΑΤΑΓΛΗΚΤΙΚΟ ΕΠΙΠΛΟ ΚΟΥΡΔΙ... Από: lowprices Λήξη: 15/11/2010 245,00 € Δημοπρασία</p>
<p>OSRAM KEPIA 9W - 40W F14</p>	<p>Panasonic PT-AF2000</p>

Επισκεφτείτε τα καταστήματα στο Emarket.gr

Τι κινείται;

ΤΟ ΜΟΝΤΕΛΟ ΤΟΥ ΕΜΠΟΡΙΚΟΥ ΚΕΝΤΡΟΥ Ή ΠΟΛΥΚΑΤΑΣΤΗΜΑΤΟΣ

Το Ηλεκτρονικό Εμπορικό Κέντρο είναι ένα σύνολο-συνάθροιση ηλεκτρονικών καταστημάτων, όπου εφαρμόζεται κοινή μέθοδος πληρωμής και όλα τα ηλεκτρονικά καταστήματα είναι κάτω από μια κοινή "ομπρέλα" (επωνυμία). Τα οφέλη για τα μέλη του ηλεκτρονικού εμπορικού κέντρου είναι το χαμηλότερο κόστος και λιγότερο πολύπλοκες διαδικασίες εισαγωγής στον Παγκόσμιο Ιστό, εξειδικευμένες δυνατότητες (για παράδειγμα, ηλεκτρονικές πληρωμές), περισσότερη κίνηση. Τα πλεονεκτήματα για τον πελάτη είναι εύκολη πρόσβαση σε άλλα ηλεκτρονικά καταστήματα, κοινό περιβάλλον χρήσης (πιθανόν πρόσθετες υπηρεσίες προστιθέμενης αξίας).

Επιπλέον, τα οφέλη για τον διαχειριστή του ηλεκτρονικού εμπορικού κέντρου είναι ο διαφημιστικός χώρος, η προώθηση των μαρκών, η αύξηση των πωλήσεων των υποστηριζόμενων τεχνολογιών (π.χ. η IBM με τη World Avenue), Στα έσοδα περιλαμβάνονται συνδρομές μελών, διαφημίσεις και πιθανόν τέλη συναλλαγών.

Ένα παράδειγμα εμπορικού κέντρου ή ηλεκτρονικό πολυκατάστημα φαίνεται στο παρακάτω σχήμα και ανήκει στην εταιρία www.emall.com




Shop, Compare, Buy

ipod

Search

Search Results for: ipod

All
 Cars
 Clothing and Accessories
 Computers
 Electronics
 Flowers and Gifts
 Gift Ideas
 Health and Beauty
 Home and Garden
 Jewelry and Watches
 Kids and Family
 Magazine and Subscriptions
 Media
 Miscellaneous
 Musical Instruments & Accessories
 Office
 Sports and Outdoors
 Video Games

	<p>Apple iPod Touch 8 GB Di</p> <p>MPN: MA623LLA, Apple iOS, 8 Files: MP3, WAV, AAC, MP3 VBR</p> <p>Price Range: \$155.00 - \$275.00</p> <p>at 12 Stores</p> <p>Compare Prices >></p>
	<p>Apple iPod touch 3rd Gen</p> <p>MPN: MC011LL/A, Apple iOS, 8 Audio Files: MP3, WAV, AAC, A</p> <p>Price Range: \$285.00 - \$371.00</p> <p>at 4 Stores</p> <p>Compare Prices >></p>
	<p>Apple iPod Shuffle 2nd Generation Blue (2 GB) MP3 Player</p> <p>MPN: SHUFFLE2GOB, 2 GB (Built-in Memory), 500 Songs, 0.55 oz., Audio Files: MP3, WAV, AAC, MP3 VBR, Audible. Compare ></p> <p>Price Range: \$49.00 - \$99.00</p> <p>at 2 Stores</p> <p>Compare Prices >></p>

Υπάρχουν αρκετά οφέλη και για τους πελάτες αλλά και για τα καταστήματα στην περίπτωση του μοντέλου του ηλεκτρονικού πολυκαταστήματος. Τα κυριότερα από αυτά είναι τα εξής:

- Οφέλη για τους πελάτες:
 - Ελαττώνεται η περιήγηση στο διαδίκτυο
 - Χρήση του ίδιου περιβάλλοντος περιηγήσεως
- Οφέλη για τα καταστήματα:
 - Μειωμένο κόστος σε σχέση με την ανεξάρτητη λειτουργία του καταστήματος
 - Αυξημένη κίνηση λόγω των γειτονικών καταστημάτων

ΤΟ ΜΟΝΤΕΛΟ ΤΗΣ ΗΛΕΚΤΡΟΝΙΚΗΣ ΠΥΛΗΣ

Το μοντέλο της ηλεκτρονικής πύλης περιλαμβάνει ιστοσελίδες οι οποίες λειτουργούν ως σημεία έναρξης χρηστών στη σύνδεσή τους με το διαδίκτυο. Συνήθως αυτές οι ιστοσελίδες παρέχουν πολλές πληροφορίες από διάφορα θέματα της καθημερινότητας

για τα οποία όμως οι περισσότεροι χρήστες έχουν πάντα μεγάλο ενδιαφέρον. Τα περισσότερα θέματα αφορούν επίσης την επικαιρότητα. Οι βασικότερες από τις λειτουργίες και υπηρεσίες που προσφέρουν οι ηλεκτρονικές πύλες είναι οι εξής:

- Παρέχουν καταλόγους από άλλα web-sites
- Δίνουν ή έχουν σύνδεση με διάφορα εργαλεία αναζήτησης
- Προσφέρουν διάφορες ειδήσεις
- Παρέχουν μετεωρολογικές ειδήσεις σχετικά με τον καιρό
- Παρέχουν πληροφορίες σχετικές με διάφορους τηλεφωνικούς καταλόγους ή χάρτες
- Πολλές τέτοιες ιστοσελίδες σήμερα παρέχουν επίσης τη δυνατότητα σε χρήστες που το επιθυμούν, να δημιουργήσουν λογαριασμούς e-mail
- Μια από τις βασικές παροχές είναι επίσης και η online σύνδεση με το Χρηματιστήριο
- Παρέχουν διάφορα θέματα που αφορούν τη διασκέδαση
- Πολλά θέματα που αφορούν την τεχνολογία
- Επίσης θέματα από άλλες διάφορες επιστήμες

Υπάρχουν 2 ειδών ηλεκτρονικές πύλες:

- Οι Γενικές ή όπως αλλιώς λέγονται οριζόντιες. Αυτές περιέχουν λίγα πράγματα για πολλά θέματα
- Οι Ειδικές ή αλλιώς κάθετες. Αυτές περιέχουν πολλά πράγματα από λίγα θέματα (ή μια θεματική ενότητα)

Ένα παράδειγμα μιας οριζόντιας ηλεκτρονικής πύλης είναι η ιστοσελίδα www.in.gr ενώ ένα χαρακτηριστικό παράδειγμα μιας κάθετης διαδικτυακής πύλης είναι το



ΙΑΤΡΙΚΗ ΟΝ ΛΙΝΕ - ΚΑΛΩΣΗΡΩΑΤΕ
 :: Αν είστε μέλος, προχωρήστε κάνοντας [click εδώ](#).
 :: Μπορείτε να εγγραφείτε (δωρεάν) ως μέλος του Iatriki On Line κάνοντας [click εδώ](#).

ΠΕΡΙΟΔΙΚΑ

- Ιατρικό Βήμα
- Δελτίο Α' Παιδιατρικής Κλινικής Πανεπιστημίου Αθηνών
- Ελλ. Μαιευτική & Γυναικολογία
- Ελλ. Χειρουργική Ορθοπαιδική και Τραυματολογία
- Θέματα Μαιευτ. & Γυναικολογίας
- Ελλ. Δερματοχειρουργική
- Ελλ. Αγγειοχειρουργική
- Πανελλήνια Οφθαλμορηθμολογία
- Ελευθεροπροπνευμολογίας Ιατρος
- Ελληνική Ρευματολογία
- Εργοθεραπεία

ΕΦΗΜΕΡΙΔΕΣ

- Info Derma
- Info Respiratory & Internal Medicine
- Info Orthopaedics & Traumatology
- Info Urology
- Info Gastroenterology
- Βιολογική Ψυχιατρική
- Εν χρό
- Σύγχρονη ΩΡΛ Ενημέρωση

Medline /PubMed
[Ιατρικά άρθρα](#)
[Φαρμακευτική ενημέρωση](#)
[Ανασκόπηση διεθνούς Ιατρικού τύπου](#)
[Επικοινωνία](#)
[Links](#)
[Μηχανή Αναζήτησης](#)

Συνεχίζει τις επενδύσεις η Φαρμακευτική Εταιρεία Kleva
(27/10/10)

2ο Σεμινάριο του Εργαστηρίου Διγνωστικής Κυτταρολογίας της Ιατρικής Σχολής ΕΚΠΑ στο Π.Γ.Ν. «ΑΤΤΙΚΟΝ», με θέμα: Σύγχρονα Θέματα Κυτταρολογίας: «Η νέα ταξινόμηση αλλοιώσεων του θυρεοειδούς κατά Bethesda 2009. Τι Νέοτερο στην Πρόληψη Καρκίνου Τροχήλου της Μήτρας»
(29/9/10)

ΔΕΛΤΙΟ ΤΥΠΟΥ «ΠΥΣΙΔΑ» Πειραιάς, 23.9.2010
1η Επιστημονική Ημερίδα «ΨΥΧΑΡΓΩΣΗ» (29/9/10)

ΔΕΛΤΙΟ ΤΥΠΟΥ ΦΑΡΜΑΚΕΥΤΙΚΟΥ ΣΥΛΛΟΓΟΥ ΑΤΤΙΚΗΣ
Αθήνα, 7.9.10. (22/9/10)

ΔΕΛΤΙΟ ΤΥΠΟΥ Ι.Σ.Α. Αθήνα, 6.9.2010
ΠΕΡΙΚΟΠΗ ΔΑΠΑΝΩΝ ΣΤΟΥΣ ΔΙΣΦΑΛΙΣΜΕΝΟΥΣ (22/9/10)

ΔΕΛΤΙΟ ΤΥΠΟΥ Αθήνα, 15 Σεπτεμβρίου 2010
Εκστρατεία Ενημέρωσης και Πρόληψης με αφορμή την Παγκόσμια Ημέρα κατά του Καρκίνου του Προστάτη από το Διαδραστικό Δίκτυο Υγείας (22/9/10)

ALAPIS - Synthetic Genomics: Αποκωδικοποιώντας το Μέλλον
Σε σύναψη συνεργασίας με την εταιρεία του γενεσπύ που αποκωδικοποίησε το ανθρώπινο γονιδίωμα προώρησε η Alaris (17/9/10)

ΔΕΛΤΙΟ ΤΥΠΟΥ Ι.Σ.Α. Αθήνα, 19.7.2010
ΘΕΜΑ: 24ωρη ΠΑΝΕΛΛΑΔΙΚΗ ΠΑΝΙΑΤΡΙΚΗ ΑΠΕΡΓΙΑ ΤΕΤΑΡΤΗ 21.7.2010 (20/07/10)

ΔΕΛΤΙΟ ΤΥΠΟΥ ΣΦΕΕ, Χαλάνδρι, 16.7.10
Ανακοίνωση για τις τιμές των φαρμάκων (20/07/10)

ΔΕΛΤΙΟ ΤΥΠΟΥ Ι.Σ.Α. Αθήνα, 15.7.10
ΘΕΜΑ: «ΣΧΕΔΙΟ ΝΟΜΟΥ ΑΝΑΒΑΘΜΙΣΗ ΤΟΥ Ε.Σ.Υ. ΚΑΙ ΛΟΙΠΕΣ ΔΙΑΤΑΞΕΙΣ» (20/07/10)

ΔΕΛΤΙΟ ΤΥΠΟΥ Ι.Σ.Α. Αθήνα, 6.7.10 (29/06/10)

Προκηρύξεις

Συνέδρια

Αγγελίες

Πολιτική της Υγείας

Ενημέρωση από τον Ε.Ο.Φ.

Νέα από τις φαρμακευτικές εταιρείες

Βιοτεχνολογία και Ιατρική

Νέες Θεραπείες & εφαρμογές

Νέα προϊόντα

Ιατρικά βιβλία

Ιατρική και νέες τεχνολογίες (ηλεκτρονική κ.λπ.)

Quiz



Τελευταία ενημέρωση: 27-10-2010

* Κ: Χιλιάδες



"Κάντε κλικ δίπλα για να κατεβάσετε τον τελεματρίκι"

<http://www.iatrikionline.gr/>

ΜΟΝΤΕΛΑ ΔΥΝΑΜΙΚΗΣ ΔΙΑΜΟΡΦΩΣΗΣ ΤΙΜΩΝ

Όλο και περισσότερες εταιρίες δίνουν τη δυνατότητα στους πελάτες τους να πραγματοποιούν τις αγορές τους μέσω internet με τρόπους που στηρίζονται στα λεγόμενα «μοντέλα δυναμικής διαμόρφωσης τιμών» (dynamic pricing models). Αυτά τα μοντέλα παρέχουν στους καταναλωτές ευκαιρίες για να διαπραγματευτούν τις τιμές για τα προϊόντα που ενδιαφέρονται να αγοράσουν. Στο παρελθόν, κάποιος που ενδιαφερόταν να βρει ένα συγκεκριμένο προϊόν στην καλύτερη δυνατή τιμή έπρεπε να επισκεφτεί πολλά site που πουλούν το προϊόν και να διαπραγματευτεί στο κάθε site για τυχόν έκπτωση για την αγορά του. Τώρα, αρκεί να επισκεφτεί ένα site που παρέχει κάποιο μοντέλο διαμόρφωσης τιμών και με εξοικονόμηση χρόνου και μεγαλύτερες πιθανότητες επίτευξης καλύτερης τιμής, να διαπραγματευθεί για το προϊόν που επιθυμεί. Οι περισσότερες επιχειρήσεις αγοράζουν προϊόντα μαζικά, γι' αυτό έχουν τη δυνατότητα να αγοράσουν αρκετά πιο φθηνά τα προϊόντα ανάλογα και με τις ποσότητες που αγοράζουν. Τέτοιες τιμές δεν θα μπορούσε να έχει κάποιος μεμονωμένος καταναλωτής που θα ήθελε να αγοράσει ένα συγκεκριμένο προϊόν, σε

πολύ μικρή ποσότητα. Όμως, τώρα υπάρχουν site που επιτρέπουν στους καταναλωτές να συνενωθούν με άλλους αγοραστές και έτσι να παραγγείλουν προϊόντα σε μεγαλύτερες ποσότητες και καλύτερες τιμές. Παρακάτω θα παρουσιάσουμε τα πιο γνωστά και συνήθως χρησιμοποιούμενα μοντέλα.

Name Your-Price Model

Αυτό το μοντέλο επιτρέπει στους καταναλωτές να ονομάσουν την τιμή που προτίθενται να πληρώσουν για τα προϊόντα και τις υπηρεσίες που τους ενδιαφέρουν (π.χ. το site της priceline, www.Priceline.com, που παρατίθεται παρακάτω). Οι εταιρίες έχουν τη δυνατότητα να το κάνουν αυτό επειδή έχουν σχηματίσει συνεταιρισμούς με βιομηχανίες που παρέχουν αυτά τα προϊόντα ή τις υπηρεσίες. Εταιρίες όπως η Priceline λειτουργούν σαν μεσολαβητής για διαπραγμάτευση με τους προμηθευτές προς όφελος του αγοραστή. Η διαδικασία που ακολουθείται είναι η προσφορά τιμής από τον αγοραστή η οποία μέσω της εταιρίας μεσολαβητή φτάνει στον προμηθευτή, ο οποίος είτε αποδέχεται την προσφορά και τότε ο αγοραστής υποχρεούται να ολοκληρώσει τη συνδιαλλαγή, είτε την απορρίπτει και ο αγοραστής, αν θέλει, κάνει καινούργια προσφορά.

Συγκεκριμένα η Priceline παρέχει τη δυνατότητα «name your price» για αεροπορικά εισιτήρια, δωμάτια ξενοδοχείων, ενοικιάσεις αυτοκινήτων, κ.α.. Για να κατανοήσουμε καλύτερα πως δουλεύει το μοντέλο, θα χρησιμοποιήσουμε για παράδειγμα την αγορά ενός αεροπορικού εισιτηρίου μέσω της Priceline. Εάν ενδιαφερόμαστε για εγχώρια πτήση (στην έδρα της Priceline), κατ' αρχάς εισάγουμε την τοποθεσία αναχώρησης, τον προορισμό, τον αριθμό εισιτηρίων που θέλουμε να αγοράσουμε και την τιμή που προσφέρουμε για την αγορά του κάθε εισιτηρίου. Μετά επιλέγουμε τις ημερομηνίες που θέλουμε να ταξιδέψουμε και τα αεροδρόμια μέσα ή κοντά στις πόλεις αναχώρησης ή προορισμού. Όσο πιο ευέλικτοι είμαστε με τους διακανονισμούς του ταξιδιού τόσο μεγαλύτερες πιθανότητες έχουμε να αποκτήσουμε τα εισιτήρια στη τιμή που επιθυμούμε. Η Priceline λαμβάνει την προσφορά μας και προσπαθεί να διαπραγματευθεί με την αεροπορική εταιρία μια τιμή μικρότερη από αυτή που έχουμε προσφέρει εμείς. Εάν η αεροπορική εταιρία αποδεχθεί την προσφορά της Priceline, η διαφορά της προσφερόμενης από την Priceline τιμής από τη δική μας προσφερόμενη τιμή παραμένει ως κέρδος στην Priceline. Εάν η αεροπορική εταιρία δεν αποδεχθεί την προσφορά οι διαπραγματεύσεις θα

συνεχιστούν και ανάλογα με την κατάληξή τους θα έχουμε φυσικά και εμείς πιο αυξημένο κόστος από αυτό που προτεινάμε. Για τις εγχώριες πτήσεις η όλη διαδικασία διαρκεί περίπου μία ώρα.

Η Priceline είναι ένα πολύ καλό παράδειγμα πώς το Internet έχει αλλάξει ριζικά τον τρόπο που διεξάγονται οι συναλλαγές. Στην περίπτωση των αεροπορικών εταιριών, καθημερινά μεγάλος αριθμός θέσεων μένει άδειος. Με τη βοήθεια site όπως της Priceline αυξάνουν τα κέρδη τους πουλώντας εισιτήρια που θα έμεναν απούλητα, ενώ ταυτόχρονα κερδίζουν και η Priceline και οι καταναλωτές που καταφέρνουν αρκετά καλύτερες τιμές από τις κανονικές

The screenshot shows the priceline.com website interface. At the top, there is a navigation bar with links for 'sign in', 'my profile', 'my trips', 'check your request', and 'help'. Below this is a main menu with categories like 'flights', 'hotels', 'rental cars', 'vacation packages', 'cruises', 'tours & attractions', 'pricebreakers', 'city guides', and 'groups'. The main content area is divided into several sections:

- shop for discount travel:** This section features a search form for flights, hotels, cars, vacations, and cruises. It includes fields for 'from (city or airport)', 'to (city or airport)', 'departing' and 'returning' dates, 'number of tickets', and 'seating class'. A 'search now!' button is prominently displayed. Below the search form, there are 'more search options' like '> one way', '> multi-city', and '> by airline'. To the left of the search form, there is a promotional image of a man pointing and the text 'NO ONE DEALS LIKE WE DO™™'. Above the search form, there are radio buttons for 'flight only', 'flight + hotel', and 'flight + hotel + car', with a note 'package and save up to \$325:'. There are also tabs for 'flights', 'hotels', 'cars', 'vacations', and 'cruises'.
- freshly negotiated travel deals:** This section features a large banner for 'find HALF PRICE HOTELS EVERY DAY' with a 'search now!' button. Below the banner, there is a link for 'vacations on sale + \$25 rewards card'.
- my travel deals for you:** This section lists various deals, including 'Flights Under \$200 Round-Trip -- No Booking Fees! Ends in 9 days', 'Round-Trip Flight + Mid Size Rental Car from \$263! Ends in 9 days', 'American Airlines U.S. Sale from \$49! OW based on R/T travel! Ends in 28 days', 'Up to 20% off a Las Vegas Hotel + Up to \$900 in Shopping Coupons! Ends Today', 'Up to 77% off a 4-flight Baja Mexico Cruise! Ends Today', and 'Up to 71% off a 7-flight Eastern Caribbean Cruise! Ends Today'. A 'See all PriceBreakers' link is also present.
- name your own price® - get deeper discounts:** This section offers three bidding options: 'save up to 40% on flights', 'save up to 50% on hotels', and 'save up to 30% on cars', each with a 'bid now' button.
- priceline city guides:** This section is powered by ZAGAT SURVEY and lists various cities: New York, Boston, San Francisco, Dallas, Seattle, Los Angeles, Chicago, Las Vegas, Miami, Orlando, San Diego, London, Atlanta, Denver, Phoenix, and Explore City Guides.





Comparison Pricing Model

Αυτό το μοντέλο επιτρέπει στους πελάτες να επιλέξουν από μια μεγάλη ποικιλία εμπορών για να βρουν το επιθυμητό προϊόν ή υπηρεσία στη φθηνότερη τιμή. Ένα δημοφιλές site που βασίζεται σε αυτό το μοντέλο είναι το www.BottomDollar.com το οποίο παραθέτουμε πιο κάτω. Αυτού του είδους τα site διαπραγματεύονται συνεταιρισμούς με πολλούς εμπόρους για να πετύχουν τις καλύτερες όσο το δυνατόν τιμές και να τις προσφέρουν στους καταναλωτές. Πρέπει όμως να είμαστε προσεκτικοί επειδή μερικά site μπορεί να βγάζουν κέρδη από συμφωνίες με συνεταιρισμούς με δικούς τους εμπόρους, έτσι ίσως να μη βλέπουμε όλους τους διαθέσιμους εμπόρους που υπάρχουν σε ολόκληρο το Web και μπορεί να μην καταφέρνουμε να βρούμε την καλύτερη τιμή που υπάρχει διαθέσιμη. Τα μεγάλα και δημοφιλή site όπως το BottomDollar ψάχνουν στα μεγαλύτερα καταστήματα που υπάρχουν στο Internet, προσπαθώντας να προσελκύσουν το ευρύ κοινό.

COMPUTERS CAMERAS ELECTRONICS FURNITURE MOVIES VIDEO GAMES JEWELRY CLOTHING MORE

bottomdollar.com™ December 1, 2008 [HOME](#)

Find Products

BROWSE CATEGORIES	TOP RATED PRODUCTS
<p>Appliances Small Kitchen Appliances, Large Kitchen Appliances, Air Cleaners, Heating & Cooling, Vacuums, More</p> <p>Auto Parts Wheels & Tires, Motorcycle, Engine, More</p> <p>Babies & Kids Nursery, Strollers & Other Gear, Girls Clothing, More</p> <p>Books Computers, Self-help, Health & Fitness, Political Science, Study Aids, More</p> <p>Cameras Digital Cameras, Camcorders, 35mm & SLR, Flashes, More</p> <p>Clothing Women, Men, Accessories, More</p> <p>Computers Laptops & Accessories, Printers & Accessories, Storage Devices, More</p> <p>Electronics TVs, GPS, Portable Audio, More</p> <p>Flowers & Gourmet Wines, Desserts & Sweets, Gift Baskets, Flowers & Plants, More</p> <p>Furniture</p>	<p>Top Laptop Computers More Popular Products</p> <p>Acer Aspire One AOA110-1295 Notebook \$299.00</p> <p> 1.6GHz Intel Atom N270, 512MB DDR2, 8GB SSD, Linux, 8.9" LCD (4 Sellers)</p> <p>Acer - LUS020A012 <input type="button" value="Compare Prices"/></p> <p>Rating: ★★★★★ (2 Reviews)</p> <hr/> <p>Apple 15" MacBook Pro Notebook \$1,449.99</p> <p> 2.4GHz Intel Core 2 Duo, 2GB DDR2, 200GB, DVD±RW DL, Mac OS X v10.5 Leopard, 15.4" LCD (12 Sellers)</p> <p>Apple - MB133LLA <input type="button" value="Compare Prices"/></p> <p>Rating: ★★★★★ (13 Reviews)</p> <hr/> <p>Acer Aspire ONE A150-1447 Notebook \$379.00</p> <p> 1.6GHz Intel Atom N270, 1GB DDR2, 160GB, Windows Vista Home Premium, 8.9" LCD (8 Sellers)</p> <p>Acer - LUS050B133 <input type="button" value="Compare Prices"/></p> <p>Rating: ★★☆☆☆ (3 Reviews)</p> <hr/> <p>Apple 13.3" MacBook Notebook \$849.99</p> <p> 2.1GHz Intel Core 2 Duo Mobile, 1GB DDR2, 120GB, DVD-</p>

Το site BottomDollar χρησιμοποιεί την τεχνολογία intelligent-agent για να ψάχνει σε ολόκληρο το Web και να βρίσκει τα προϊόντα που επιθυμούμε στις χαμηλότερες τιμές. Τα intelligent-agents είναι προγράμματα που ψάχνουν και διαχειρίζονται πολύ μεγάλο αριθμό δεδομένων και παρουσιάζουν τα στοιχεία που βρήκαν βάση αυτών των δεδομένων. Ο πελάτης μπορεί να ψάξει απ' ευθείας για το προϊόν που τον ενδιαφέρει ή να κοιτάξει σε κάποια από τις ομαδοποιημένες κατηγορίες που υπάρχουν στο site όπως φαίνεται στην εικόνα. Η υπηρεσία ψάχνει στους καταλόγους περισσότερων από 1.000 online μικροεμπόρων για να βρει την καλύτερη διαθέσιμη τιμή. Η όλη διαδικασία εύρεσης διαρκεί λιγότερο από ένα λεπτό (χρόνος σαφέστατα μικρότερος από εκείνον που θα απαιτούταν για την επίσκεψη στο site καθενός από τους εμπόρους που διαθέτουν το προϊόν που θέλουμε, προκειμένου να συγκρίνουμε τις τιμές τους). Η δυνατότητα της πολύ εύκολης απ' ευθείας σύγκρισης των τιμών όλων των προμηθευτών, τους αναγκάζει να κρατάνε τις τιμές τους σε ανταγωνιστικά επίπεδα.

Το μοντέλο αυτό προσεγγίζει (ή θυμίζει) το μοντέλο των e-malls και συγκεκριμένα την υποκατηγορία των «ηλεκτρονικών μεσιτών».

Demand-Sensitive Pricing Model

Όπως οτιδήποτε στην οικονομία το web είναι ευαίσθητο στην ζήτηση για τη διαμόρφωση των τιμών. Οι πελάτες-καταναλωτές απαιτούν καλύτερες και πιο γρήγορες υπηρεσίες σε φθηνότερες τιμές. Με τη χρήση του internet, έχει ενδυναμωθεί η συγκρότηση ομάδων μεμονωμένων αγοραστών για να επιτευχθεί έκπτωση ανάλογη με αυτή που παρέχεται σε μεγάλες επιχειρήσεις.

Η λογική αυτού του μοντέλου είναι ότι όσο περισσότεροι πελάτες κάνουν μια μεμονωμένη αγορά ενός συγκεκριμένου προϊόντος, τόσο.

μικρότερο είναι το κόστος ανά άτομο. Η πώληση προϊόντων ξεχωριστά είναι ακριβή διαδικασία επειδή ο πωλητής δίνει τιμή στο προϊόν του, έτσι ώστε να καλύπτει το κόστος πώλησης ανά άτομο και φυσικά να βγάζει ένα κέρδος. Εάν οι πελάτες αγοράζουν το προϊόν μαζικά, αυτό το κόστος μοιράζεται και το όριο κέρδους αυξάνεται. Ένα site που βασίζεται σε αυτή τη λογική και παρέχει μια πολύ μεγάλη γκάμα προϊόντων που μπορούμε να αγοράσουμε, είναι το www.mercata.com. Επειδή η ποικιλία και η κοστολόγηση των προϊόντων διαφέρουν στα παρόμοια με το mercata

site, καλό θα ήταν να επισκεφτούμε αρκετά site πριν κάνουμε την αγορά μας.

Mercata.com
What you need, when you need it

English

Search

Search

Related Searches

- Laptop Computer
- Cheap Digital Camera
- Laptops
- Golf Bags
- Work From Home
- Notebook
- Airline Tickets
- Mortgage
- Cheap Laptop

Related Searches

- Houses For Sale
- Bread Maker
- Foreclosure
- Zojirushi Bbcc V20
- Wholesale Clothing
- Tiller

Related Searches: [Zojirushi Bbcc V20](#) | [Wholesale Clothing](#) | [Tiller](#) | [Ryobi Tiller](#) | [Tillers](#) | [Hotels](#)

[Bookmark this page](#) | [Make this your homepage](#)

Τα παραπάνω είναι τα κυριότερα και πλέον διαδεδομένα μοντέλα δυναμικής διαμόρφωσης τιμών. Βέβαια υπάρχουν και πολλά άλλα μοντέλα, τα οποία θα αναφέρουμε συνοπτικά πιο κάτω.

Bartering Model

Είναι το μοντέλο που στηρίζεται στην ανταλλαγή ενός προϊόντος για κάποιο άλλο. Τόσο μεμονωμένοι καταναλωτές όσο και επιχειρήσεις μπορούν να προσφέρουν ένα προϊόν και περιμένουν προσφορές ανταλλαγής με κάποιο άλλο προϊόν. Το site www.ubarter.com είναι ένα τέτοιο site.

Rebates

Πολλά sites προσφέρουν εκπτώσεις στα προϊόντα τους για να προσελκύσουν πελάτες. Το www.eBates.com είναι ένα site που προσφέρει έκπτωση σε κάθε αγορά. Το eBates έχει συνάψει συνεταιρισμούς με χονδρέμπορους και μικροέμπορους που προσφέρουν καλύτερες τιμές και το site τις παρουσιάζει (πασάρει) στους πελάτες του με τη μορφή εκπτώσεων. Βάζοντας αντίτιμο στις επισκέψεις των πελατών στο site, το eBates εξασφαλίζει την πίστη αλλά και την ικανοποίηση των πελατών τους, ενώ ταυτόχρονα

βγάξει και κέρδη. Το μοντέλο αυτό προσεγγίζει (ή θυμίζει) το μοντέλο «name your price». Η διαφορά είναι ότι δεν κάνει διαπραγμάτευση ανά περίπτωση αλλά βασίζεται σε συμφωνίες (εκ των προτέρων διαπραγματεύσεις) που εξασφαλίζουν καλύτερες τιμές λόγω των προσδοκιών που έχουν οι συμβαλλόμενοι έμποροι για αυξημένες πωλήσεις.

Offering Free Products and Services

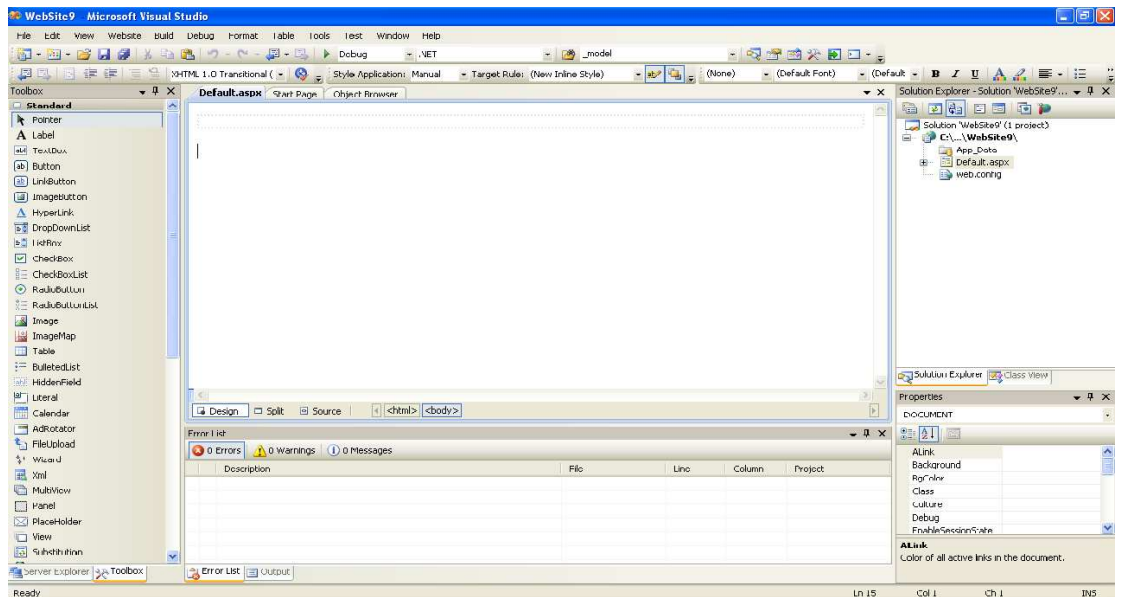
Η διαφήμιση είναι ένας τρόπος που επιτρέπει στα site να προσφέρουν τις υπηρεσίες τους δωρεάν. Τα site αυτά δίνουν χώρο στις σελίδες τους για διαφημίσεις ή ακόμα έχουν συνάψει συμφωνίες με χορηγούς και έχουν links στις σελίδες τους. Με το εισόδημα που αποκτούν από τις διαφημίσεις και τις χορηγίες, τα site αυτά εξασφαλίζουν τη βιωσιμότητα τους και μπορούν να προσφέρουν δωρεάν υπηρεσίες.

ΚΕΦΑΛΑΙΟ III: Η ΧΡΗΣΗ ΤΟΥ ΕΡΓΑΛΕΙΟΥ VISUAL STUDIO.NET ΚΑΙ ΤΗΣ ASP.NET

ΕΙΣΑΓΩΓΗ ΚΑΙ ΓΡΗΓΟΡΗ ΜΑΤΙΑ ΣΤΟ ΛΟΓΙΣΜΙΚΟ

Όταν ανοίγουμε το λογισμικό βλέπουμε ότι αποτελείται από μια πλειάδα παραθύρων ενσωματωμένων μέσα σε ένα container παράθυρο. Κάθε ένα από τα παράθυρα αυτά χρησιμοποιείται για διαφορετικούς σκοπούς. Θα δείξουμε ποια είναι τα βασικότερα παράθυρα του λογισμικού και θα κάνουμε μια μικρή επεξήγηση για το καθένα από αυτά.

- **Εργαλειοθήκη:** Όταν ανοίγουμε μια σελίδα ή ένα άλλο στοιχείο για επεξεργασία, η εργαλειοθήκη (Toolbox) προσφέρει εργαλεία που δίνουν τη δυνατότητα να ενσωματώσουμε στοιχεία ελέγχου στη σελίδα.
- **Επιφάνεια σχεδίασης:** Η αλλιώς *design grid*. Εδώ δημιουργούνται και υπόκεινται σε επεξεργασία οι σελίδες. Αρχικά εμφανίζεται μια αρχική σελίδα η οποία ονομάζεται `default.aspx` και είναι η αντίστοιχη του `index.html` που συνήθως βλέπουμε στα home pages των διαφόρων ιστοτόπων.
- **Solution Explorer/Database Explorer:** Κάθε νέα σελίδα που δημιουργείται οργανώνεται σε φακέλους στον Solution Explorer. Κάθε βάση δεδομένων που δημιουργείται για την σελίδα εμφανίζεται στον Database Explorer. Χρησιμοποιούμε τα tabs στο κάτω μέρος του pane για να αλλάξουμε ανάμεσα στα 2 Explorer προγράμματα.
- **Ιδιότητες:** Εμφανίζει ιδιότητες που σχετίζονται με την σελίδα ή με το τρέχων αντικείμενο.
- **Χρήση των Panes:** Τα panes μπορούν να μετακινούνται, να αποκρύπτονται και να αλλάζουν μέγεθος, ώστε να αυξηθεί ο εκμεταλλεύσιμος χώρος της οθόνης. Για να μεγαλώσουμε ή να μικρύνουμε ένα pane, μπορούμε να σύρουμε το εσωτερικό περίγραμμα δεξιά ή αριστερά. Αν έχουμε 2 ή περισσότερα panes, το ένα πάνω στο άλλο, στην άκρη της οθόνης, μπορούμε να μεγαλώσουμε το pane που βρίσκεται πιο χαμηλά σέρνοντας το πάνω περίγραμμα πάνω ή κάτω. Μια τυπική φωτογραφία των pane από το λογισμικό φαίνεται στο παρακάτω σχήμα.



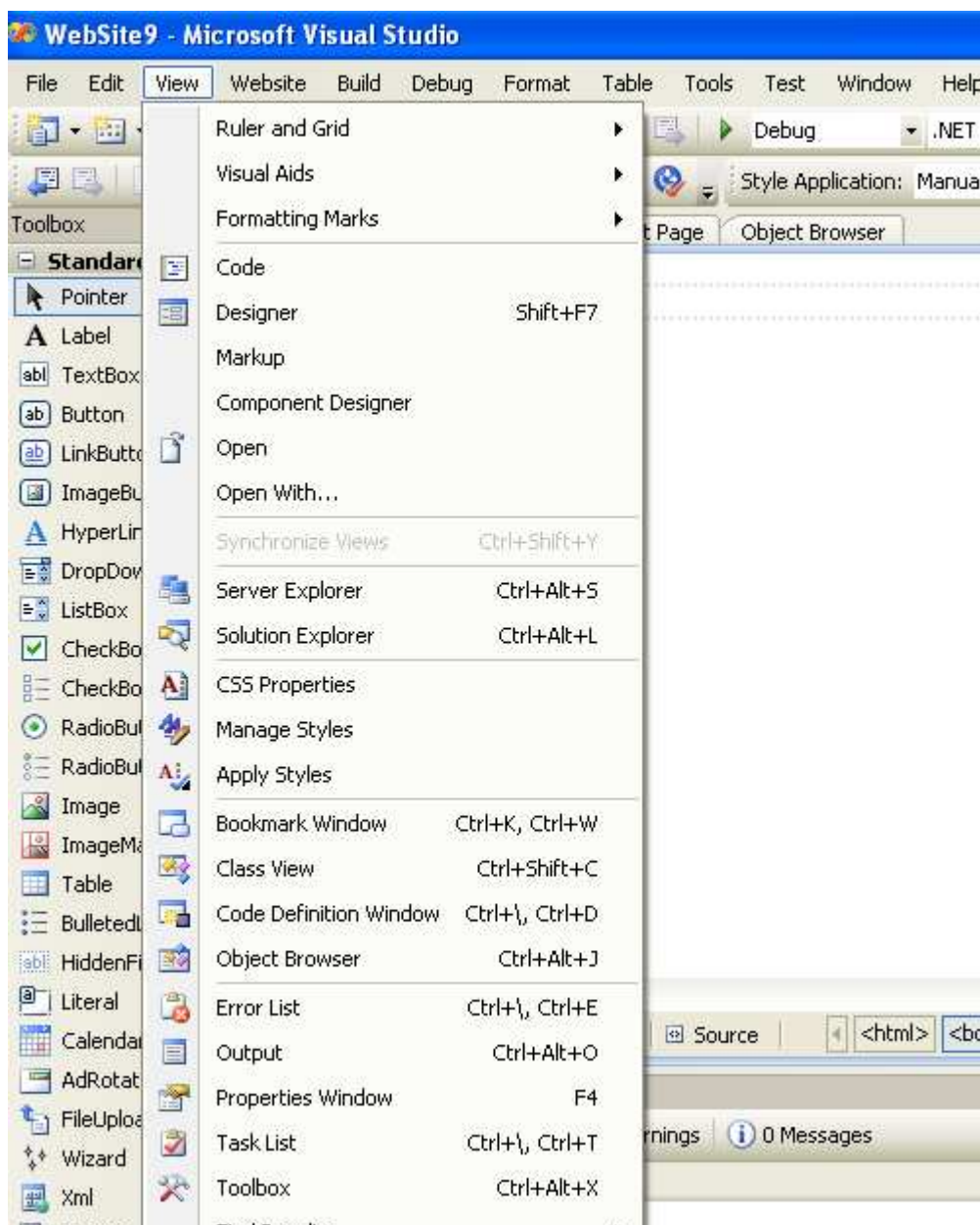
Το περίγραμμα των Panes του λογισμικού Visual Studio.NET

Τα panes μπορούν να πάρουν διαφορετικές ιδιότητες έτσι ώστε να είναι:

- **Floating:** Μετατρέπει ένα pane σε free-floating παράθυρο με δυνατότητα αλλαγής μεγέθους και μετακίνησης στο παράθυρο του προγράμματος.
- **Dockable:** Αποσυνδέει ένα παράθυρο που είναι σε καρτέλα.
- **Tabbed Document:** Μετακινεί το pane στην περιοχή Editing, και δημιουργεί ένα tab.
 - Κάντε κλικ στο tab για να το εμφανίσετε
 - Κάντε δεξί κλικ στο tab και επιλέξτε Dockable για να το επαναφέρετε στο πρόγραμμα.
- **Auto Hide:** Εναλλάσσει τα panes από ανοιχτά σε κρυμμένα και αντίστροφα. Για να το επαναφέρετε κάντε κλικ ή απλά αφήστε το δείκτη του ποντικιού πάνω στο όνομά του.
- **Hide:** Κρύβει το pane και εμφανίζεται μόνο το όνομά του στο περιθώριο. Για να το επαναφέρετε κάντε κλικ ή απλά αφήστε το δείκτη του ποντικιού πάνω στο όνομά του.

Το μενού View: Η επιλογή **View** στο μενού, παρέχει πρόσβαση σε όλα τα προαιρετικά panes (ή αλλιώς windows επειδή μπορούν να είναι και έτσι). Αν κλείσουμε ένα pane από το Close (X), μπορούμε να το επαναφέρουμε από το μενού View επιλέγοντάς το όνομά του. Μερικές επιλογές στο μενού View όπως Object Browser και Error List, θα έχουν χρησιμότητα μόνο όταν αρχίσουμε να φτιάχνουμε μια ιστοσελίδα. Στις περισσότερες περιπτώσεις εμφανίζονται αυτόματα όταν είναι απαραίτητο. Στο μενού View επίσης, υπάρχει η δυνατότητα να εμφανίζουμε και να

αποκρύπτουμε διάφορα toolbars. Ένα snapshot του μενού view φαίνεται στο παρακάτω σχήμα



Μια απεικόνιση του μενού View

Συμβατότητα των ιστοσελίδων με τα διάφορα Web Browsers : Κάθε web author μπορεί να επιλέξει ανάμεσα σε 2 αντικρουόμενα πράγματα: Συμβατότητα με όλους τους web browsers και εντυπωσιακά στοιχεία. Αν θέλουμε η σελίδα να είναι προσβάσιμη πρακτικά από όλους τότε πρέπει να είναι συμβατή με πολύ αρχικές εκδόσεις των web browsers, όπως Internet Explorer 3 και Netscape Navigator 3. Αυτοί όμως δεν υποστηρίζουν νέες και εντυπωσιακές τεχνολογίες. Αν θέλουμε συμβατότητα με όλους τους νέους browsers τότε πρέπει να περιοριστούμε μόνο σε

αυτούς. Σχεδόν όλοι πλέον έχουν νέες εκδόσεις. Πολύ λίγα sites είναι συμβατά με εκδόσεις πριν της 4 και οι μεγαλύτερες εταιρίες έχουν τις πιο πρόσφατες XHTML specifications. Η XHTML είναι το μέλλον των browsers οπότε όλοι προσπαθούν να την ακολουθούν. Στο παράθυρο διαλόγου **Επιλογές** στο Visual Studio ρυθμίζεται η συμβατότητα των browsers. Επιλέξτε **Tools** ⇨ **Options** από το μενού. Ανοίγει το παράθυρο διαλόγου Επιλογές.. Κάντε κλικ στο + (αν υπάρχει) δίπλα στο Text Editor HTML. Κάντε κλικ στο **Validation**. Επιλέξτε τον επιθυμητό Browser

Δημοσιοποίηση ιστοσελίδων: Όπως θα γνωρίζετε ήδη το να φτιάξετε μια σελίδα και να δουλεύει στον υπολογιστή σας δεν την κάνει προσβάσιμη στο ευρύ κοινό. Αυτό μπορεί να συμβεί κατοχυρώνοντας ένα domain name και δημοσιεύοντάς το σε έναν Web server. Οι εταιρίες που παρέχουν τέτοιες υπηρεσίες ονομάζονται

hosting service

hosting provider

Web presence provider

WPP εν συντομία

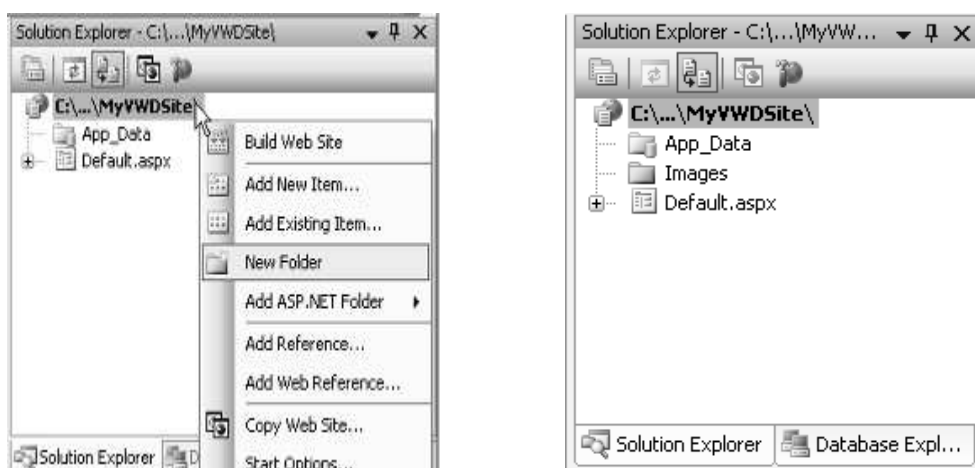
Τα hosting services που χρησιμοποιούν συγκεκριμένα την τεχνολογία που έχουμε στο VS για να φτιάξουμε ένα Web site είναι τα *ASP.NET 2.0 Hosters*. Θα χρειαστούμε ένα hosting service που να υποστηρίζει ASP.NET 2.0 και SQL Server 2005/2008. Μπορούμε να βρούμε μια λίστα με τέτοια hosting services στο www.asp.net/hosters/.

ΔΗΜΙΟΥΡΓΙΑ ΕΝΟΣ WEB SITE

Από την μεριά του Visual Studio, ένα Web site είναι βασικά ένας φάκελος ο οποίος περιέχει ίσως και άλλους φακέλους και όλα τα αρχεία που χρειάζονται για την δημιουργία ενός ολοκληρωμένου Web site. Από το μενού του Visual Studio διαλέγουμε **File** ⇨ **New Web Site** και κάνουμε **Click ASP.NET Web Site**. Από το drop-down list για την τοποθεσία που θα το σώσουμε, διαλέγουμε την θέση. Από το drop-down list για την γλώσσα. Έχουμε επιλέξει να χρησιμοποιήσουμε την C# . Κάνουμε **Click OK**.

ΕΠΕΞΕΡΓΑΣΙΑ ΣΕΛΙΔΩΝ

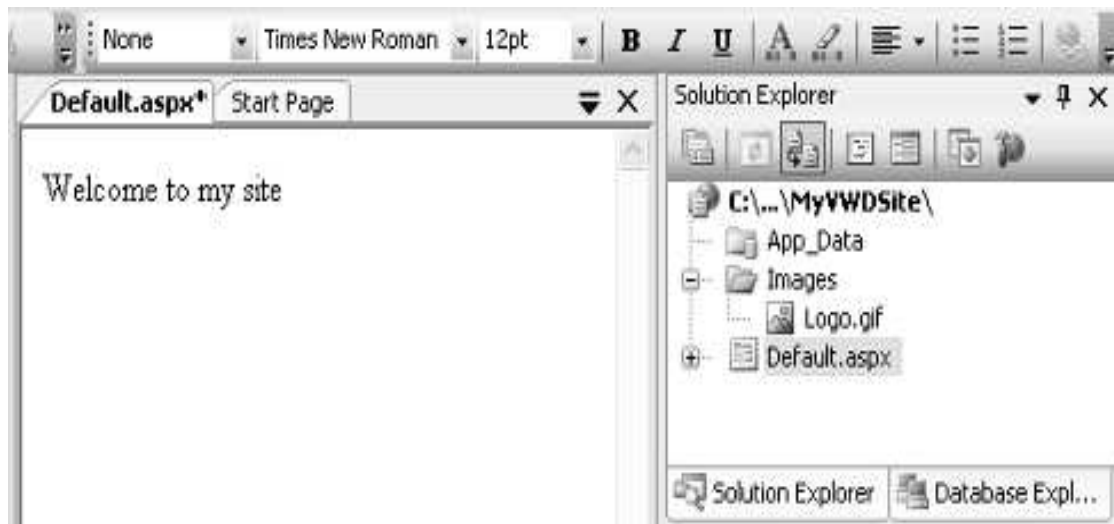
Μπορούμε να χρησιμοποιήσουμε φακέλους για να οργανώσουμε τις σελίδες και άλλα συστατικά στον ιστοχώρο μας με τον ίδιο σχεδόν τρόπο που χρησιμοποιούμε τους φακέλους στα Windows για να οργανώσουμε τα αρχεία μας. Παραδείγματος χάριν, εάν θελήσουμε να δημιουργήσουμε έναν φάκελο για την αποθήκευση των εικόνων όλου του site κάνουμε τα εξής βήματα: Σιγουρευόμαστε ότι το site είναι ανοικτό και πάμε στο Solution Explorer pane. Κάνουμε Right-click στο όνομα του site στην κορυφή του δέντρου του Solution Explorer και Επιλέγουμε New Folder, όπως φαίνεται στο παρακάτω σχήμα.



Δημιουργία και χρήση φακέλων

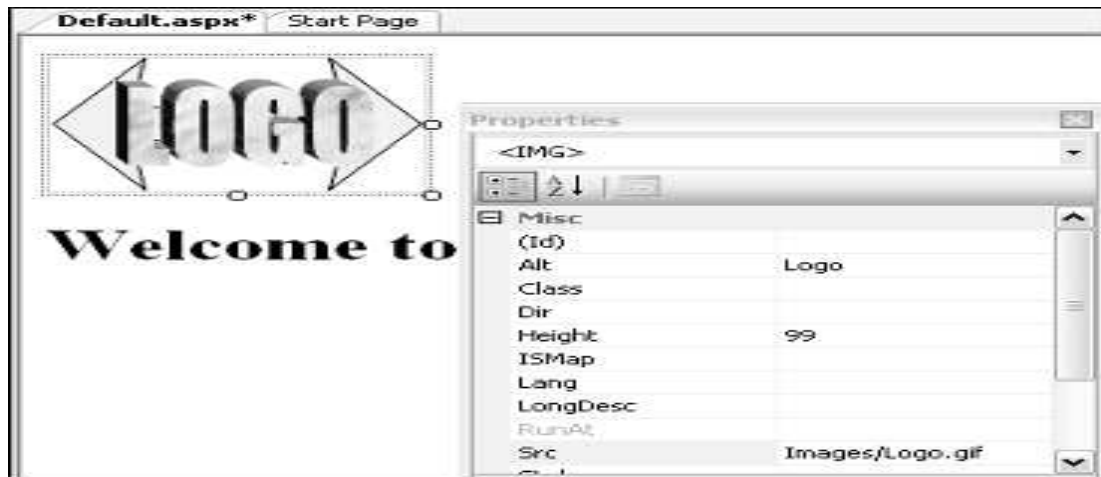
Για να επεξεργαστούμε μια υπάρχουσα σελίδα μέσα στο VS, πρέπει αρχικά να ανοίξουμε τη σελίδα έτσι ώστε να είναι ορατή στην επιφάνεια του καμβά σχεδίασης. Για να ανοίξουμε μια σελίδα, κάνουμε διπλό κλικ στο εικονίδιο του στο solution explorer. Όταν ανοίγουμε μια σελίδα .aspx (όπως Default.aspx), βλέπουμε τα κουμπιά Design και Source στο κατώτατο σημείο της επιφάνειας του καμβά σχεδίασης και έτσι μπορούμε να μεταβούμε από το ένα mode στο άλλο.

Το VS σας επιτρέπει να εκδώσετε μια σελίδα .aspx είτε χρησιμοποιώντας το mode του Design είτε στο mode του Source (ακατέργαστα HTML και ASP.NET). Το κείμενο στο Design γίνεται όπως και η δακτυλογράφηση στο Microsoft Word και στο Frontpage. Η επιλογή και η επεξεργασία text δουλεύει το ίδιο στο VS όπως και στα προγράμματα επεξεργασίας κειμένου. Για να κάνουμε format ένα κομμάτι κειμένου, πρώτα το επιλέγουμε (highlight) με το ποντίκι. Κατόπιν επιλέγουμε μια επιλογή από την γραμμή εργαλείων μορφοποίησης. Στο παρακάτω σχήμα για παράδειγμα, επιλέγουμε το κείμενο «Welcome to my site».



Κατόπιν από το κουμπί Block Format στα αριστερά της γραμμής εργαλείων μορφοποίησης αλλάζουμε το heading με την επιλογή Heading1 <H1>. Για να προσθέσουμε μια εικόνα στην ιστοσελίδα μας, πρώτα σιγουρευόμαστε ότι έχουμε αποθηκεύσει την εικόνα σε έναν φάκελο μέσα στο Solution Explorer. Σιγουρευόμαστε ότι όλα τα αρχεία τα οποία συνθέτουν το Web site που δημιουργούμε— συμπεριλαμβανομένων των εικόνων — είναι μέσα σε φακέλους στο Solution Explorer. Αλλιώς, όταν κάνουμε upload το site μας σε κάποιο Web server, οι εικόνες δεν θα περιλαμβάνονται στο upload, το οποίο σημαίνει ότι οποιοσδήποτε προσπαθεί να ανοίξει αυτή την ιστοσελίδα δεν θα μπορεί να τις δει (αντ' αυτού θα βλέπει ένα μικρό κόκκινο X εκεί όπου έπρεπε να εμφανίζεται η εικόνα. Για να προσθέσουμε λοιπόν μια εικόνα απλώς πρέπει να σύρουμε το εικονίδιο της από τον φάκελο στο οποίο βρίσκεται στην ιστοσελίδα.

Σχεδόν οτιδήποτε και αν προσθέσετε σε μια ιστοσελίδα είναι ένα *object*. Και όπως τα objects στον πραγματικό κόσμο έτσι και τα objects σε μια ιστοσελίδα έχουν ιδιότητες. Οι ιδιότητες ενός αντικειμένου είναι settings τα οποία εκφράζουν τα χαρακτηριστικά του — όπως πχ size, shape, location πάνω στην ιστοσελίδα κλπ. Για να δούμε και ίσως για να αλλάξουμε τις ιδιότητες ενός object απλά επιλέγουμε το αντικείμενο και κοιτάμε το παράθυρο ιδιοτήτων (Properties pane) ή κάνουμε right-click στο object και επιλέγουμε Properties. Οι ιδιότητες του αντικειμένου εμφανίζονται σε ένα *properties sheet*. Το παρακάτω σχήμα δείχνει ένα παράδειγμα όπου βλέπουμε τις ιδιότητες μιας εικόνας.



Η ιδιότητες μιας εικόνας

Το tag κοντά στην κορυφή του παραθύρου ιδιοτήτων είναι επίσης η ένδειξη ότι όλες οι εικόνες σε όλα τα web sites στην HTML εμφανίζονται με το tag.

Κάθε ιστοσελίδα έχει ένα τίτλο ο οποίος εμφανίζεται στο πάνω μέρος (μπλε μπάρα) καθώς βλέπουμε την ιστοσελίδα. Αυτός ο ίδιος τίτλος εμφανίζεται σαν link σε αυτή την ιστοσελίδα από τις μηχανές αναζήτησης (εφόσον το web site είναι αναρτημένο σε κάποιο web server). Στην HTML, ο τίτλος της ιστοσελίδας πρέπει να τοποθετηθεί μεταξύ των tag <title>...</title>, το οποίο tag με την σειρά του πρέπει να είναι εξ' ολοκλήρου εντός των tag <head>...</head>. Στο Visual Studio, μπορούμε να ακολουθήσουμε τα παρακάτω βήματα για να δημιουργήσουμε τον τίτλο μιας ιστοσελίδας. Από το drop-down list στην κορυφή του Properties sheet, επιλέγουμε <DOCUMENT>. Κάνουμε Scroll προς το τέλος του Properties sheet και γράφουμε τον τίτλο της ιστοσελίδας στην ιδιότητα Title.

Στο Design view μιας ιστοσελίδας μας δίνεται μια αρκετά καλή ιδέα για το πώς θα φαίνεται μια ιστοσελίδα σε ένα Web browser. Αλλά ποτέ δεν θα δούμε την ακριβή προβολή της ιστοσελίδας. Για να κάνουμε testing για το πώς φαίνονται οι ιστοσελίδες που δημιουργούμε σε ένα Web browser κάνουμε τα εξής (οτιδήποτε σας βολεύει από τα παρακάτω):

Right-click σε κάποιο άδειο μέρος της ιστοσελίδας στο design view → View In Browser.

Ή Click στο **View in Browser** κουμπί στο toolbar (αριστερή πλευρά στο παρακάτω σχήμα)

Ή Πατήστε το Ctrl+F5.

ΔΙΑΜΟΡΦΩΣΗ ΙΣΤΟΧΩΡΟΥ ΜΕ GUEST ΧΡΗΣΤΕΣ ΚΑΙ REGISTERED ΧΡΗΣΤΕΣ

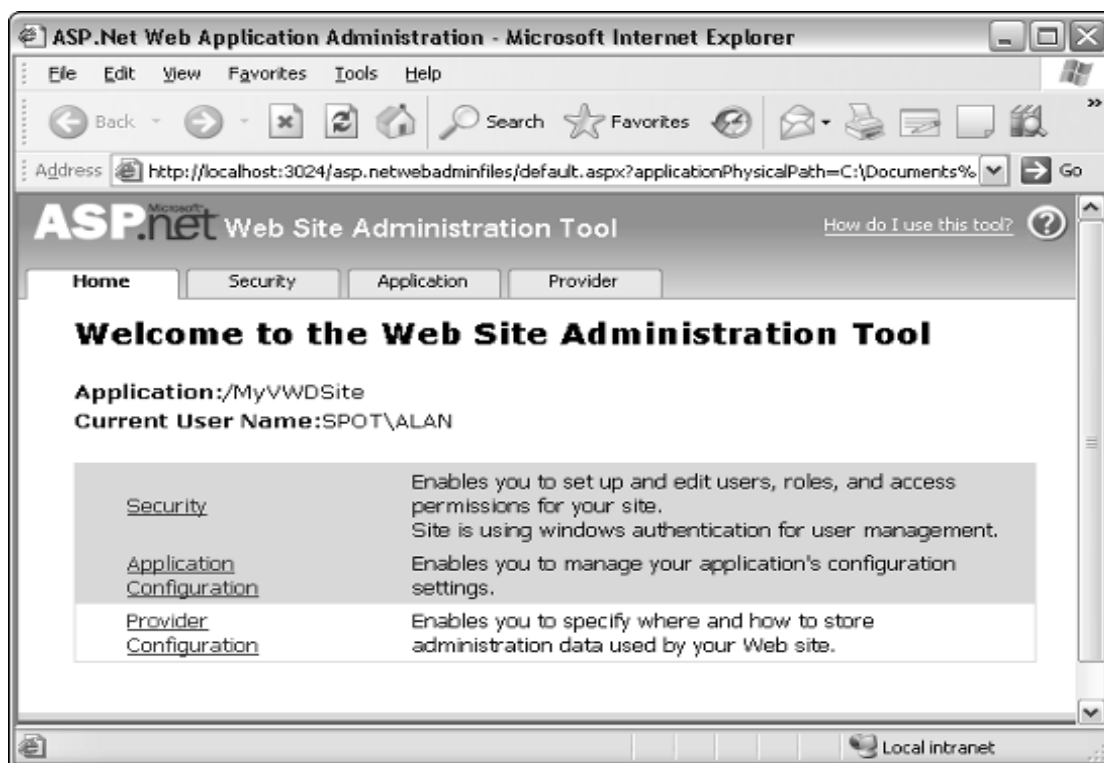
Με το VS είναι πολύ εύκολο να δημιουργήσουμε ένα site με χρήστες-μέλη. Μπορούμε να ελέγχουμε ποιοι θα έχουν πρόσβαση στο περιεχόμενο της σελίδας μας. Για παράδειγμα μπορούμε να έχουμε περιεχόμενο για ανώνυμους χρήστες που απλά κάνουν απλή περιήγηση στο site και περιεχόμενο για τους χρήστες με προνόμια, χρήστες που έχουν λογαριασμό στο site μας. Η βασική ιδέα είναι απλή. Δημιουργούμε ένα φάκελο με το όνομα Member-Pages, όπου θα βάλουμε μέσα όλες τις πληροφορίες που θα εμφανίζονται στα μέλη (members-only content). Στη συνέχεια ορίζουμε ένα role, με το όνομα SiteMembers. Τέλος δημιουργούμε έναν κανόνα που να ορίζει ότι οι ανώνυμοι χρήστες δεν μπορούν να έχουν πρόσβαση στο φάκελο MemberPages, μόνο χρήστες του SiteMember role μπορούν να έχουν πρόσβαση εκεί. Με άλλα λόγια, αν κάποιος τυχαία βρεθεί στο site μας να μην μπορεί να δει το περιεχόμενο που προορίζεται για τα μέλη, παρά μόνο δημιουργώντας έναν λογαριασμό χρήστη. Αρχικά αποφασίζουμε πώς να οργανώσουμε το περιεχόμενο. Κάποιο μέρος των πληροφοριών πολύ πιθανό πρέπει να είναι προσβάσιμο σε ανώνυμους χρήστες. Ανώνυμος χρήστης θεωρείται αυτός που δεν έχει κάνει Log in στο site ή δεν έχει δημιουργήσει λογαριασμό. Επίσης μπορεί να υπάρχει περιεχόμενο μόνο για χρήστες με προνόμια, χρήστες δηλαδή που έχουν κάνει λογαριασμό στο site μας. Για αυτό το διαχωρισμό του περιεχομένου θα πρέπει να δημιουργήσουμε ένα ξεχωριστό φάκελο με το περιεχόμενο των προνομιούχων χρηστών. Έστω ότι τον ονομάζουμε αυτό τον φάκελο MemberPages. Κρατούμε το γενικό περιεχόμενο του ιστοχώρου έξω από αυτό τον φάκελο. Για να δημιουργήσουμε ένα τέτοιο φάκελο κάνουμε δεξί κλικ στο όνομα του site μας πάνω στον Solution Explorer και επιλέγουμε New Folder. Πληκτρολογούμε το όνομα του φακέλου και πατάμε Enter.

Η ΧΡΗΣΗ ΤΟΥ WEBSITE ADMINISTRATION TOOL

Το WSA (Web Site Administration Tool) είναι το εργαλείο που χρησιμοποιείται για τη διαχείριση των σελίδων στο website μας. Επιλέγουμε **Website** → **ASP.NET Configuration** από το μενού. Θα ανοίξει σε ένα Web browser. Τα tabs Home, Security, Application και Provider όπως και τα links στο Home tab, παρέχουν εργαλεία για τη δημιουργία και τη διαχείριση ενός site με χρήστες-μέλη. Πρέπει να

βεβαιωθούμε ότι το Visual Studio συνδέεται με τον SQL Server, όπου αποθηκεύονται οι πληροφορίες για τους χρήστες. Για να κάνουμε test στη σύνδεση:

- Κλικ στο **Provider tab** ή την επιλογή **Provider Configuration** στο **Home tab**.
- Επιλέγουμε **Select A Single Provider For All Site Management Data**
- Βεβαιωνόμαστε ότι είναι επιλεγμένο το **AspNetSqlProvider**



- Κάνουμε κλικ στο **Test link** για να βεβαιωθούμε ότι συνδέεται
- Θα πρέπει να εμφανιστεί το μήνυμα **“Successfully established a connection to the database.”**.

Σε αυτό το σημείο έχουμε πει στο Visual Studio ότι θέλουμε να χρησιμοποιήσουμε τον SQL Server σαν βάση διαχείρισης της πληροφορίας για τους χρήστες και τους λογαριασμούς χρηστών. Το επόμενο βήμα είναι η επιλογή του authentication type. Για να μπει κάποιος σε ένα site πρέπει να παρέχει 2 πληροφορίες.

- Το user name (ή το e-mail)
- Το password.

Το user name ορίζει ποιος χρήστης είσαι. Το password πιστοποιεί ότι είσαι αυτός που λες – υποθέτοντας ότι δεν έχεις αποκαλύψει τον κωδικό σου σε άλλους ανθρώπους.

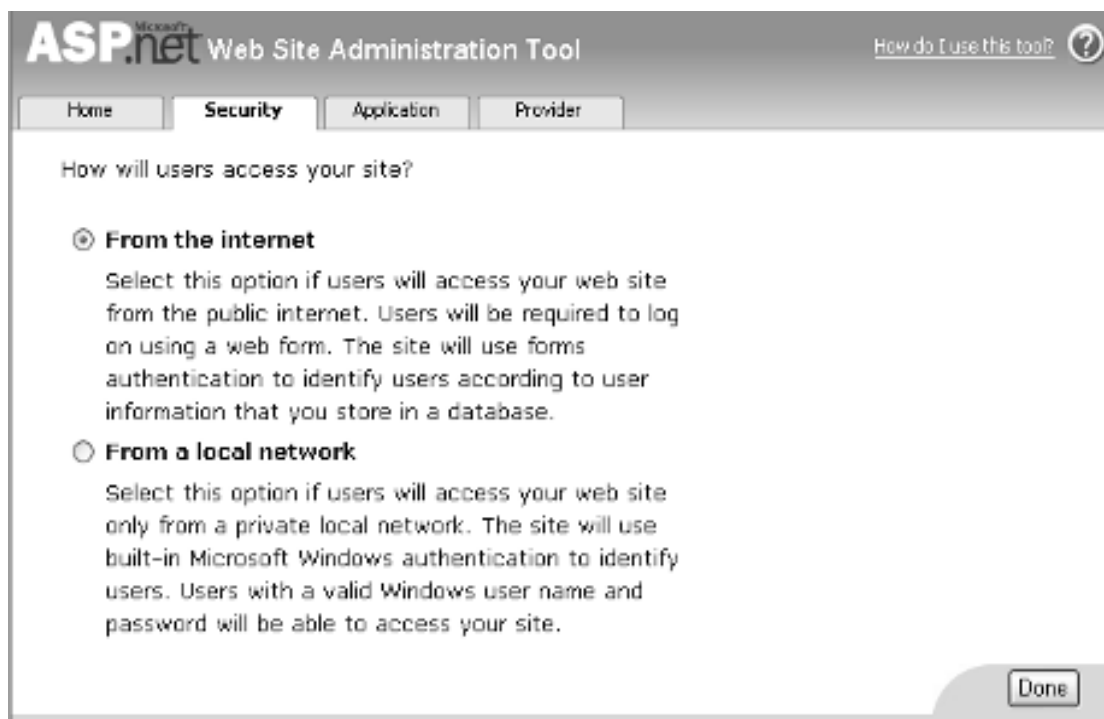
Αυτή η διαδικασία των 2 βημάτων λέγεται **authentication** γιατί πιστοποιεί αν ο χρήστης δίνει αληθινά δεδομένα. Στο VS υπάρχουν 2 επιλογές μορφής authentication:

- **Windows authentication** ή
- **Forms authentication.**

Το Windows authentication μπορεί να δουλέψει μόνο σε ένα μικρό Web site σε τοπικό επίπεδο. Αν το site θα δημοσιευτεί στο Internet, χρειαζόμαστε Forms authentication: Στο WSA εργαλείο, κάνουμε κλικ στο **Security tab**.

- Κλικ στο **Select Authentication Type option under Users** στο **Security tab**.
- Στην επόμενη σελίδα επιλέγουμε **From the Internet option**.
- Κλικ στο **Done**.

Έχοντας επιλέξει data provider και authentication mode, είμαστε έτοιμοι να θέσουμε τους κανόνες ασφάλειας του site. Το πρώτο βήμα είναι να δημιουργήσουμε ένα *role* που θα διαχωρίζει τους χρήστες μέλη από τους ανώνυμους χρήστες.



ΔΗΜΙΟΥΡΓΙΑ ΡΟΛΩΝ ΓΙΑ ΤΗΝ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΧΡΗΣΤΩΝ

Αν σκεφτούμε έναν επαγγελματικό οργανισμό, η πρόσβαση στα επαγγελματικά resources και πληροφορίες βασίζεται στους ρόλους των ανθρώπων. Για παράδειγμα τα διοικητικά στελέχη (executives) μπορούν να έχουν πρόσβαση στα πάντα, οι managers έχουν πρόσβαση σε λεπτά ζητήματα στο τμήμα τους, και οι εργαζόμενοι

έχουν πρόσβαση στα υπόλοιπα. Ο ρόλος μέσα στην επιχείρηση ορίζει και σε ποια στοιχεία μπορούν να έχουν πρόσβαση. Σε ένα site χρηστών, η πρόσβαση βασίζεται σε roles. Ένας χρήστης που απλά κοιτάει τη σελίδα και δεν έχει κάνει sign in είναι ανώνυμος. Για το διαχωρισμό των χρηστών σε ανώνυμους και επώνυμους χρειαζόμαστε έναν role (SiteMembers). Για να θεωρείται κάποιος SiteMember, πρέπει να έχει λογαριασμό στη σελίδα και να είναι συνδεδεμένος. Για τη χρήση ρόλων στο site, πρέπει να τους ενεργοποιήσουμε στο WSA.

- Κλικ στο **Security tab** στο **Web Site Administration Tool**.
- Αν οι ρόλοι δεν είναι ενεργοποιημένοι κάντε κλικ στο **Enable Roles**

Το κεντρικό pane κοντά στο κάτω μέρος του Security tab έχει ένα Link που λέει Create ή Manage Roles. Δεν χρειάζεται να δημιουργήσουμε ρόλο για ανώνυμους χρήστες. Οποιοσδήποτε που επισκέπτεται τη σελίδα είναι εξ' ορισμού ανώνυμος χρήστης.

Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
Existing users: 0 Create user Manage users Select authentication type	Existing roles: 0 Disable Roles Create or Manage roles	Create access rules Manage access rules

Μόλις έχουμε ενεργοποιήσει τους ρόλους μπορούμε να ορίσουμε ρόλους για τους χρήστες της σελίδας μας. Ένα απλό site μελών χρειάζεται απλά μόνο ένα ρόλο, όπως το SiteMembers.

You can optionally add roles, or groups, that enable you to allow or deny groups of users access to specific folders in your Web site. For example, you might create roles such as "managers," "sales," or "members," each with different access to specific folders.

Create New Role	
New role name: <input type="text" value="SiteMembers"/>	<input type="button" value="Add Role"/>

Σε αυτό το σημείο μπορείτε να δημιουργήσουμε και άλλους ρόλους βάζοντας το όνομα και κάνοντας κλικ στο Add Role για κάθε έναν καινούργιο που δημιουργούμε.

ΔΗΜΙΟΥΡΓΙΑ ΚΑΝΟΝΩΝ ΠΡΟΣΒΑΣΗΣ

Ο σκοπός του να δημιουργείς ρόλους είναι για να διαχωρίσεις τους διάφορους τύπους ανθρώπων που επισκέπτονται το site σας, ώστε να μπορούμε να ελέγχουμε την πρόσβαση. Ο τρόπος για να γίνει αυτό είναι ορίζοντας τα *access rules* για διάφορους ρόλους. Για το Member Pages. Βάζουμε όλο το περιεχόμενο για τους προνομιούχους χρήστες σε αυτό το φάκελο κρατώντας το χωριστά από το υπόλοιπο. Χρειαζόμαστε έναν κανόνα που να αρνείται την πρόσβαση στους ανώνυμους χρήστες. Αυτό γίνεται με το Web Site Administration tool.

- Στο **Web Site Administration Tool**, κάνουμε κλικ στο **Security tab**
- Στην στήλη **Access Rules**, κάνουμε κλικ στο link **Create Access Rules**
- Στην αριστερή στήλη της σελίδας που εμφανίζεται **Add New Access Rule**, κάνουμε κλικ στο όνομα του φακέλου για τον οποίο θέλουμε να δημιουργήσουμε τον κανόνα.

Ο όρος *content folder* σημαίνει για φακέλους που δημιουργούμε όπως ο MemberPages δεν υπάρχει λόγος να ορίσουμε κανόνες στον root folder, ή για τους ειδικούς φακέλους όπως App_Data. Αυτοί οι φάκελοι έχουν ήδη τους απαραίτητους κανόνες ορισμένους από την αρχή. Αν αλλάξουμε αυτούς τους κανόνες το σίγουρο είναι ότι θα δημιουργηθούν προβλήματα. Στην κεντρική στήλη επιλέγουμε το ρόλο για τον οποίο θα δημιουργήσουμε κανόνα πρόσβασης. Στην επόμενη εικόνα έχουμε επιλέξει το φάκελο MemberPages. Στην κεντρική στήλη επιλέξαμε το ρόλο Anonymous Users για να δημιουργήσουμε κανόνα.

- Στην στήλη **Permission**, επιλέγουμε **Allow** ή **Deny** για τους χρήστες του ρόλου στον επιλεγμένο φάκελο.
- Στην επόμενη εικόνα βλέπουμε ότι επιλέξαμε την επιλογή Deny. Με άλλα λόγια οι ανώνυμοι χρήστες δεν θα έχουν πρόσβαση στο φάκελο MemberPages.

Add New Access Rule

Select a directory for this rule: <ul style="list-style-type: none"> MyVWDSite <ul style="list-style-type: none"> App_Data MemberPages 	Rule applies to: <input type="radio"/> Role <input checked="" type="radio"/> SiteMembers <input type="radio"/> user <input type="text"/> <input type="radio"/> All users <input checked="" type="radio"/> Anonymous users	Permission: <input type="radio"/> Allow <input checked="" type="radio"/> Deny
	<input type="button" value="OK"/>	

ΔΙΑΧΕΙΡΙΣΗ ΚΑΝΟΝΩΝ ΠΡΟΣΒΑΣΗΣ

Για να αναθεωρήσουμε ή να αλλάξουμε ή να διαγράψουμε κανόνες που έχουμε ορίσει στο παρελθόν, κάνουμε κλικ στο link **Manage Access Rules** στο **Security tab** του **Web Site Administration Tool**. Οι κανόνες πρόσβασης ορίζονται σε μια βάση folder-by-folder. Γι'αυτό στην αριστερή στήλη της σελίδας που ανοίγει πρέπει να επιλέγουμε το όνομα του φακέλου για τον οποίο θα επεξεργαστούμε τους κανόνες.

Στην επόμενη εικόνα έχουμε κάνει κλικ στο Manage Access Rules. Στην συνέχεια κάναμε κλικ στο MemberPages στα αριστερά για να εμφανιστούν οι κανόνες πρόσβασης του συγκεκριμένου φακέλου. Οι κανόνες του φακέλου εμφανίζονται στην κεντρική στήλη.

Manage Access Rules

<ul style="list-style-type: none"> MyVWDSite <ul style="list-style-type: none"> AdminPages App_Data MemberPages 	<table border="1"> <thead> <tr> <th>Permission</th> <th>Users and Roles</th> <th>Delete</th> </tr> </thead> <tbody> <tr> <td>Deny</td> <td> [anonymous]</td> <td>Delete</td> </tr> <tr> <td>Allow</td> <td> [...]</td> <td>Delete</td> </tr> </tbody> </table>	Permission	Users and Roles	Delete	Deny	[anonymous]	Delete	Allow	[...]	Delete
	Permission	Users and Roles	Delete							
Deny	[anonymous]	Delete								
Allow	[...]	Delete								
<input type="button" value="Add new access rule"/>										

Προσέξτε ότι αφού έχουμε επιλέξει το MemberPages, ο πρώτος κανόνας που εμφανίζεται μας λέει ότι απαγορεύεται στους ανώνυμους χρήστες να έχουν πρόσβαση

σε αυτό το φάκελο. Είναι σημαντικό να καταλάβουμε ότι οι κανόνες εφαρμόζονται με σειρά από πάνω προς τα κάτω. Το Deny εμφανίζεται πρώτο, με το everyone else θα έχουν πρόσβαση στο φάκελο. Στο συγκεκριμένο παράδειγμα “everyone else” σημαίνει αυτοί που ανήκουν στο SiteMembers, γιατί μόνο αυτό τον ρόλο έχουμε ορίσει.

- Ο κανόνας Allow [all] βασίζεται στο εξής: αν έχετε δημοσιεύσει κάτι σημαίνει ότι θέλετε να το μοιραστείτε με άλλους.
- Ο κανόνας Deny σημαίνει ότι : **“Θέλω να μοιραστώ κάποιες πληροφορίες στο φάκελο MemberPages με άλλους ανθρώπους. Μόνο με μερικούς από αυτούς που επισκέπτονται το site μου. Δε θέλω να το μοιραστώ με όλους του ανώνυμους χρήστες του site μου.”**

Αυτό είναι αρκετό για να αποτρέψουμε την πρόσβαση στους ανώνυμους χρήστες στον φάκελο MemberPages.

ΔΗΜΙΟΥΡΓΙΑ ΛΟΓΑΡΙΑΣΜΟΥ ΧΡΗΣΤΗ

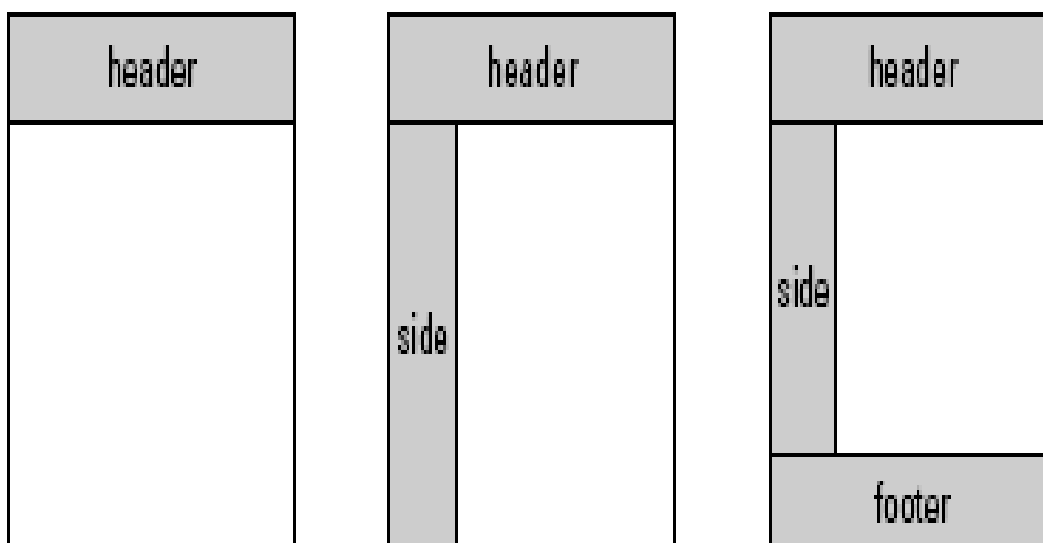
Θα χρειαστούμε τουλάχιστον έναν λογαριασμό χρήστη κατά την ανάπτυξη του site μας, που θα χρησιμοποιούμε για testing και debugging. Δημιουργήστε ένα λογαριασμό για τον εαυτό σας.

- Μπείτε στο Web Site Administration tool. **Website⇒ASP.NET Configuration**
- Στο Web Site Administration tool, κάντε κλικ στο **Security tab**. Στη συνέχεια κάντε κλικ στο **Create User** στα αριστερά κάτω από το Users heading. Θα μεταφερθείτε στην σελίδα Create User page όπου μπορείτε να φτιάξετε έναν χρήστη.
- Επειδή ο λογαριασμός είναι υποθετικός, δεν χρειάζεται να δώσετε ιδιαίτερη προσοχή στο User Name, E-mail, και το Security Question, και Security Answer.
- **Ωστόσο, το password πρέπει να είναι τουλάχιστον 7 χαρακτήρων και πρέπει να περιέχει τουλάχιστον ένα μη αλφαριθμητικό χαρακτήρα.**

Create User	Roles
<p>Sign Up for Your New Account</p> <p>User Name: <input type="text" value="TestMember"/></p> <p>Password: <input type="password" value="••••••••"/></p> <p>Confirm Password: <input type="password" value="••••••••"/></p> <p>E-mail: <input type="text" value="alan@coolnerds.com"/></p> <p>Security Question: <input type="text" value="Favorite guitarist"/></p> <p>Security Answer: <input type="text" value="hendrix"/></p> <p><input type="button" value="Create User"/></p>	<p>Select roles for this user:</p> <p><input checked="" type="checkbox"/> SiteMembers</p>
<input checked="" type="checkbox"/> Active User	

ΔΗΜΙΟΥΡΓΙΑ MASTER PAGES

Τα Master Pages είναι ένας πολύ μεθοδικός τρόπος που δίνει στο web site σταθερή μορφοποίηση ως προς το design διότι μας επιτρέπουν να ορίζουμε περιεχόμενο το οποίο θα εμφανίζεται σε κάθε σελίδα του site. Ένα Master Page μας επιτρέπει να ορίσουμε μια γενική μορφοποίηση σε όλες τις σελίδες. Για παράδειγμα, θα θέλαμε ένα σταθερό header να δείχνουμε το logo, θα θέλαμε ένα bar στο πλάι που εμφανίζουμε links και navigation controls, θα θέλαμε ένα footer στο κάτω μέρος κάθε σελίδας για να δείχνουμε και άλλα links, ή θα θέλαμε μια σύνθεση από header + sidebar + footer όπως φαίνεται στο παρακάτω διάγραμμα.



Για να κάνουμε λοιπόν τα παραπάνω, δημιουργούμε ένα φάκελο για τα master pages διότι θα είναι πιο εύκολο να τις βρούμε και να τις διαχειριζόμαστε. Από το menu, διαλέγουμε Layout → Insert Table. Στο Insert Table παράθυρο διαλόγου που ανοίγει διαλέγουμε Template. Διαλέγουμε το κατάλληλο template από το drop-down list. Για παράδειγμα, διαλέγουμε το Header and Side and Footer. Κατόπιν, μετακινούμε το ContentPlaceHolder όπως μας βολεύει. Για το ContentPlaceHolder pane δεν πολύ-χρειάζεται να δημιουργήσουμε κάποιο στυλ διότι θα γεμίσει με κάποια ιστοσελίδα. Ωστόσο αν κάνουμε στοίχιση στο pane στο πάνω μέρος του table cell θα είναι πιο εύκολο να διαχειριστούμε τις σελίδες που θα βάζουμε μέσα.

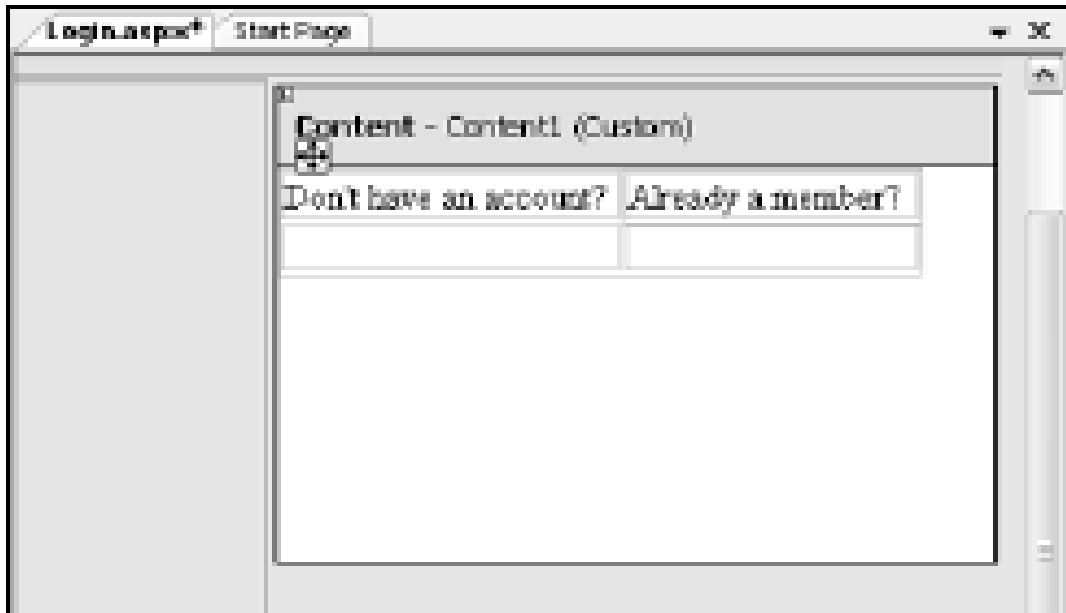
Όσον αφορά τη χρήση ενός Master Page, για να χρησιμοποιήσουμε ένα Master page πρέπει να δημιουργήσουμε ένα νέο Web Form. Κάνουμε Right-click στο όνομα του site στην κορυφή του Solution Explorer και διαλέγουμε: Add New Item. Στο Add New Item παράθυρο διαλόγου που ανοίγει διαλέγουμε Web Form και δίνουμε ένα όνομα στην ιστοσελίδα που δημιουργούμε. Κατόπιν, επιλέγουμε το Select Master Page check box. Στο παράθυρο διαλόγου Select a Master Page που ανοίγει κάνουμε click στο όνομα του φακέλου που περιέχει το Master Page (Masters, για παράδειγμα έστω ότι είναι το όνομα του φακέλου). Μετά κάνουμε click στο όνομα του Master Page που θέλουμε να χρησιμοποιήσουμε. Όταν διαλέγουμε View in Browser για να δούμε μια σελίδα που χρησιμοποιεί το Master page, οι φωτογραφίες που υπάρχουν μέσα στο Master page πρέπει να φαίνονται στο browser. Αν δούμε ένα κόκκινο X εκεί που έπρεπε να φαίνεται η φωτογραφία, πρέπει να κάνουμε μια μικρή αλλαγή στο Master page:

- Κλείνουμε τον browser, μετά κάνουμε double-click στο Master Page στο Solution Explorer και το ανοίγουμε.
- Με ανοικτή τη Master Page, κάνουμε click στο Source button στο κάτω μέρος για να μεταβούμε στο Source view.
- Βρίσκουμε το <img...> tag για την φωτογραφία στο οποίο θα λέει κάτι σαν: `` Προσθέτουμε το `runat="server"` στο tag.
- Κλείνουμε και σώζουμε το Master page.

Το Visual Studio προσθέρει δύο ειδών σελίδες για να δημιουργήσουμε:

- **Web Forms:** Μπορούν να περιέχουν **περιεχόμενο HTML** και ASP.NET Server controls. Το όνομα τους έχει επέκταση .aspx.

- **HTML Pages:** Μπορούν να περιέχουν περιεχόμενο HTML αλλά ΟΧΙ ASP.NET controls. Το όνομα τους έχει επέκταση .html.



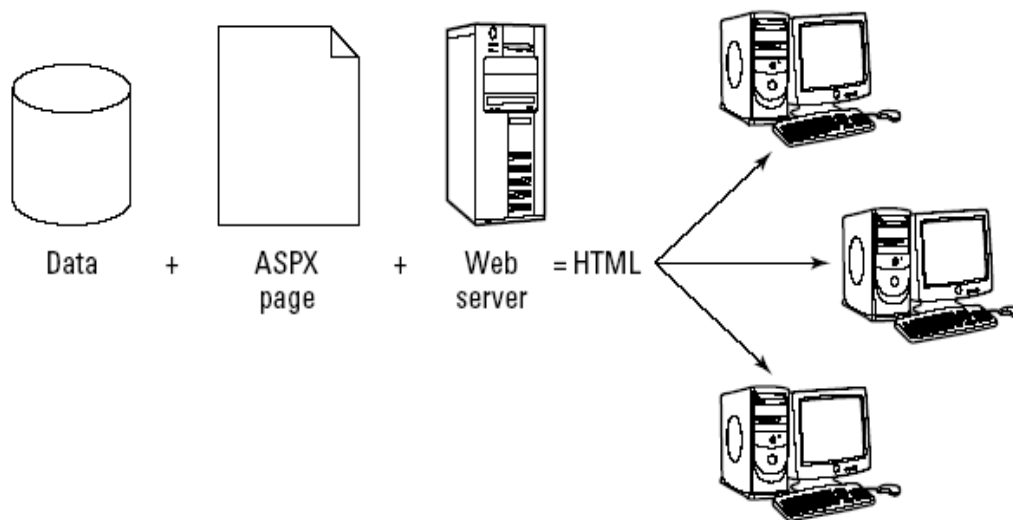
Μπορούμε να αλλάξουμε ένα πίνακα ή τα περιεχόμενα του οποιαδήποτε στιγμή: Για να προσθέσουμε ένα link στη σελίδα μας μαρκάρουμε κάποιο text και κάνουμε δεξί κλικ και κατόπιν επιλέγουμε Format→Convert to HyperLink. **Διαλέγουμε το link Type.** Για παράδειγμα, διαλέγουμε http: για μια ιστοσελίδα. Έχουμε την δυνατότητα να έχουμε διαφορετικούς τύπους για links (ftp, http, κλπ.)

Ο ευκολότερος τρόπος για να προσθέσουμε εικόνες σε ένα Web site είναι πρώτα να τις βάλουμε όλες σε ένα φάκελο στο Solution Explorer. Για να διαλέξουμε μια φωτογραφία την κάνουμε click. Αφού έχει επιλεγεί η φωτογραφία εμφανίζεται με κάποιο border που δείχνει 3 dragging handles. Επίσης, στο παράθυρο ιδιοτήτων δείχνει ένα tag. Η ιδιότητα Src (source) είναι το path για το αρχείο της εικόνας.

ΒΑΣΙΚΕΣ ΙΔΙΟΤΗΤΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ ΤΗΣ ASP.NET

Η ASP.NET είναι ένα σετ από controls για το χτίσιμο μιας δυναμικής data-driven ιστοσελίδας (Web site). Ας υποθέσουμε ότι πληκτρολογούμε κάτι στο www.google.com. Ένα λεπτό αργότερα το Google στέλνει μια σελίδα με διάφορα links για άλλα Web sites. **Πώς συνέβει αυτό? Πώς λειτουργεί αυτό?** Ο Web server της Google έχει έναν ενεργό ρόλο στο να δημιουργεί τη σελίδα που επιστρέφει ο Web browser. Οι σελίδες που επιστρέφονται από τον Web server της Google είναι

dynamic και κάθε σελίδα περιέχει μόνο links που ταιριάζουν με τις λέξεις της αναζήτησης του χρήστη. Αυτά τα αποτελέσματα είναι *datadriven* και όλα αυτά τα links αποθηκεύονται στη βάση δεδομένων της Google. Η ASP.NET μπορεί να κάνει τον Web server να έχει ενεργό ρόλο στο να δημιουργεί τις σελίδες αυτές. Δεν μπορούμε να στείλουμε σελίδες που περιέχουν server controls απευθείας στους clients διότι οι clients δεν μπορούν να εκτελέσουν server controls. Ο server χρησιμοποιεί δεδομένα από τη βάση δεδομένων και πληροφορίες από την.aspx σελίδα για να αποφασίσει ακριβώς ποια HTML χρειάζεται το PC του client. Στη συνέχεια ο server δημιουργεί την κατάλληλη σελίδα και την στέλνει στον client. Τα παραπάνω φαίνονται επίσης διαγραμματικά στο παρακάτω σχήμα.



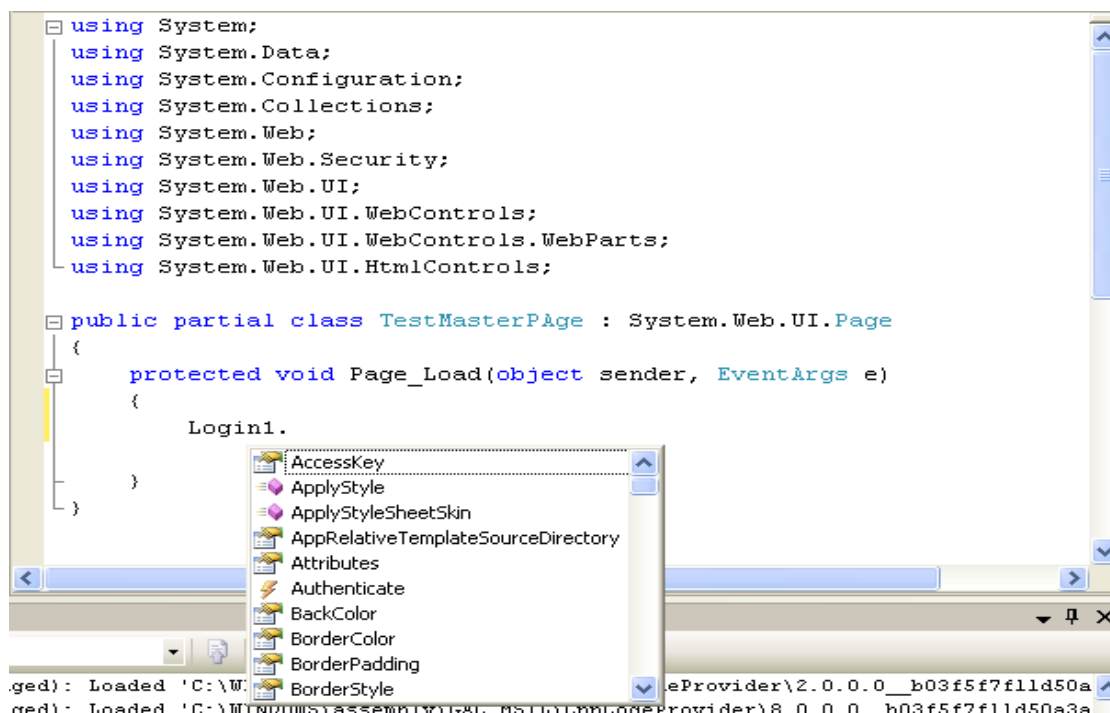
Στο Visual Web Developer, το Toolbox δείχνει τα ονόματα που χρησιμοποιούνται συνήθως από .NET Server controls, κατηγοριοποιημένα σε groups όπως

- Data,
- Validation,
- Navigation,
- Login κτλ.

Αυτά τα εργαλεία είναι ορατά μόνο στο editing ενός Web form (.aspx page). Τα server controls είναι κρυφά κατά το editing μιας .HTML σελίδας. Στο Design view μπορούμε να επιλέξουμε ένα ASP.NET control με κλικ. Το Properties φύλλο δείχνει τα ονόματα και τα properties του control.

Τα Controls στο .NET Framework οργανώνονται σε ένα ιεραρχικό *namespace*, κάτι σαν την ιεραρχική δομή στους φακέλους του σκληρού δίσκου. Στο Properties window, **System.Web.UI.WebControls.Login** είναι το πλήρες όνομα του Login

control στο .NET Framework namespace. Αν για παράδειγμα δημιουργήσουμε μια σελίδα TestMaster.aspx αυτή συνοδεύεται και από μία σελίδα TestMaster.cs. Αυτή η σελίδα προορίζεται να περιέχει τον απαραίτητο κώδικα που πρέπει να γράψουμε για την διαχείριση των controls που η σελίδα εμφανίζει. Κάτι δηλαδή παρόμοιο με την ιδέα του event programming στον προγραμματισμό. Για την προηγούμενη λοιπόν συζήτησή μας που αφορά τον ενσωματωμένο κώδικα της γλώσσας C# για την διαχείριση των controls βλέπουμε στο παρακάτω διάγραμμα ένα snapshot από μια τέτοια σελίδα κώδικα. Παρατηρούμε την δομή που μοιάζει πολύ με την δομή της γλώσσας Java.



```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;

public partial class TestMasterPage : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Login1.
    }
}
```

The screenshot shows a code editor with the above C# code. A property window is open over the `Login1.` property access, displaying a list of properties such as `AccessKey`, `ApplyStyle`, `ApplyStyleSheetSkin`, `AppRelativeTemplateSourceDirectory`, `Attributes`, `Authenticate`, `BackColor`, `BorderColor`, `BorderPadding`, and `BorderStyle`.

ASP.NET LOGIN CONTROLS

Σε αυτή την ενότητα θα μιλήσουμε για τα πιο βασικά ASP.NET εργαλεία που περιλαμβάνει η εργαλειοθήκη του Visual Studio.NET. Δεν εξαντλούμε την λίστα των εργαλείων αλλά δίνουμε μια σύντομη περιγραφή για τα πιο βασικά από αυτά:

- **Login:** Ένα control που επιτρέπει στους χρήστες να συνδεθούν στο λογαριασμό τους με όνομα χρήστη και password.
- **LoginView:** Επιτρέπει να δείχνουμε διαφορετικά πράγματα σε διάφορους χρήστες.
- **PasswordRecovery:** Παρέχει πεδία για την ανάκτηση του password.

- **LoginStatus:** Εμφανίζει ένα Login link στους ανώνυμους χρήστες ή ένα Logout link σε authenticated χρήστες.
- **LoginName:** Σε ανώνυμους χρήστες δεν εμφανίζει τίποτα, σε έναν authenticated χρήστη το login name του.
- **CreateUserWizard:** Παρέχει μια φόρμα fill-in-the-blanks για τη δημιουργία νέου λογαριασμού χρήστη. Χρησιμοποιούμε αυτό το εργαλείο για να επιτρέπουμε στους χρήστες να δημιουργούν τον δικό τους λογαριασμό. Προφανώς τα δεδομένα των χρηστών θα αποθηκευτούν στη βάση δεδομένων που συνοδεύει τον ιστοχώρο.
- **ChangePassword:** Παρέχει μια φόρμα που επιτρέπει στο χρήστη να αλλάξει password.

Για να είναι το site εύχρηστο, χρειαζόμαστε ένα login link σε κάθε σελίδα. Ένας εύκολος τρόπος είναι να βάλουμε ένα link για την Login.aspx σελίδα στο Master Page του site. Για τους χρήστες που είναι ήδη όμως συνδεδεμένοι αυτό θα μπερδεύει. Το link θα πρέπει αν δείχνει **Logout** ή κάτι άλλο για τους ήδη logged in χρήστες. Εδώ λοιπόν χρησιμοποιείται το LoginStatus control. Αρχικά δημιουργούμε μια σελίδα **Login.aspx** στο root folder πριν χρησιμοποιήσουμε οποιοδήποτε control που αναφέρουμε παρακάτω. Αλλιώς θα παίρνουμε το μήνυμα λάθους ότι η σελίδα Login.aspx δεν βρέθηκε. Στη συνέχεια κάνουμε drag ένα LoginStatus control στο Master Page και την σώζουμε. Στη συνέχεια κάνουμε δεξί κλικ σε μια σελίδα που χρησιμοποιεί το Master Page και επιλέγουμε View in Browser. Στον browser, κάνουμε κλικ στο Login link και θα μας μεταφέρει στη σελίδα Login.aspx. Στο Default.aspx page που χρησιμοποιεί το Master Page, μπορούμε να επαληθεύσουμε ότι το link έχει αλλάξει από Login σε Logout, επειδή είμαστε συνδεδεμένοι.

Το LoginView control μπορεί να διαχωρίσει τους ανώνυμους με τους authenticated χρήστες. Δεν περιορίζεται όμως στο να δείχνει κείμενο ή εικόνα. Μπορεί να χρησιμοποιηθεί για να εμφανίζει οτιδήποτε, ακόμα και ASP.NET server controls. Για να το χρησιμοποιήσουμε κάνουμε drag το εργαλείο σε μια Web page ή στο Master page. Το control έχει σχήμα κουτιού. Έχει 2 διαφορετικές views, αλλά μπορούμε να δούμε μόνο μια κάθε φορά. Επιλέγουμε λοιπόν μια μορφή από τις παρακάτω:

- **Anonymous Template:** Περιεχόμενο για ανώνυμους χρήστες.
- **LoggedIn Template:** Περιεχόμενο για authenticated χρήστες.

ΙΔΙΟΤΙΚΟΠΟΙΗΣΗ ΧΡΗΣΤΩΝ ΣΤΗΝ ASP.NET

Όταν ορίζουμε το Web site να υποστηρίζει χρήστες-μέλη, το Visual Studio αυτόματα δημιουργεί πίνακες στη βάση δεδομένων για την αποθήκευση πληροφοριών των χρηστών. Ωστόσο δημιουργεί τον ελάχιστο αριθμό πεδίων όπως το User Name, E-mail Address, και Password. Αν επιθυμούμε την αποθήκευση περισσότερων πληροφοριών για κάθε χρήστη, όπως όνομα, διεύθυνση και τηλέφωνο, πρέπει να ορίσουμε τα λεγόμενα *profile properties*. Η βασική ιδέα στο Visual Studio είναι ότι κάθε authenticated χρήστης έχει ένα profile που περιέχει πληροφορίες για αυτό τον χρήστη. Στον κώδικα μπορούμε να χρησιμοποιήσουμε την ακόλουθη απλή σύνταξη για να λάβουμε ή να αποθηκεύσουμε πληροφορίες για κάθε χρήστη. Η λέξη Profile (με κεφαλαίο P) πάντα σημαίνει “whichever user happens to be viewing this page.” Δηλαδή οποιοσδήποτε χρήστης συμβαίνει να βλέπει αυτή τη σελίδα. Για κάθε άτομο που δημιουργεί έναν λογαριασμό χρήστη στο Web site σας θα πρέπει να παρέχει τουλάχιστον τα ακόλουθα δεδομένα:

- **First Name**
- **Last name**
- **Address1**
- **Address 2** (προαιρετική 2^η γραμμή).
- **City**
- **StateProvince**
- **ZIP-PostalCode**
- **Country**

Για να ορίσουμε user profiles, πρέπει να διορθώσουμε το Web.config file στο root του Web site:

- Ανοίγουμε το Web.config με διπλό κλικ. Θα πρέπει να δούμε το όνομα του αρχείου να βρίσκεται στον Solution Explorer.
- Προσθέτουμε πάνω από `</system.Web>` tag,
 - `<profile>` και Enter.
- Προστίθεται αυτόματα `</profile>`.
- Προσθέτουμε ανάμεσα στα `<profile>` και `</profile>` :
 - `<properties>` και πατήστε Enter.
 - `</properties>` προστίθεται

- Ανάμεσα στο <properties> και </properties> πρέπει να προσθέσουμε ένα tag για κάθε property που θέλουμε να ορίσουμε :
 - ο <add propertyName /> όπου *propertyName* το όνομα του profile property.

Ένα ολοκληρωμένο κομμάτι του web.config αρχείου για τα profiles των χρηστών φαίνεται στην παρακάτω εικόνα.



```

web.config
<?xml version="1.0" encoding="utf-8"?>
<configuration xmlns="http://schemas.microsoft.com/.NetConfiguration/2.0">
  <system.web>
    <roleManager enabled="true" />
    <authentication mode="Forms" />

    <profile>
      <properties>
        <add name="FirstName"/>
        <add name="LastName"/>
        <add name="Address1"/>
        <add name="Address2"/>
        <add name="City"/>
        <add name="StateProvince"/>
        <add name="ZIPPostalCode"/>
        <add name="Country" defaultValue="USA"/>
      </properties>
    </profile>

  </system.web>
</configuration>

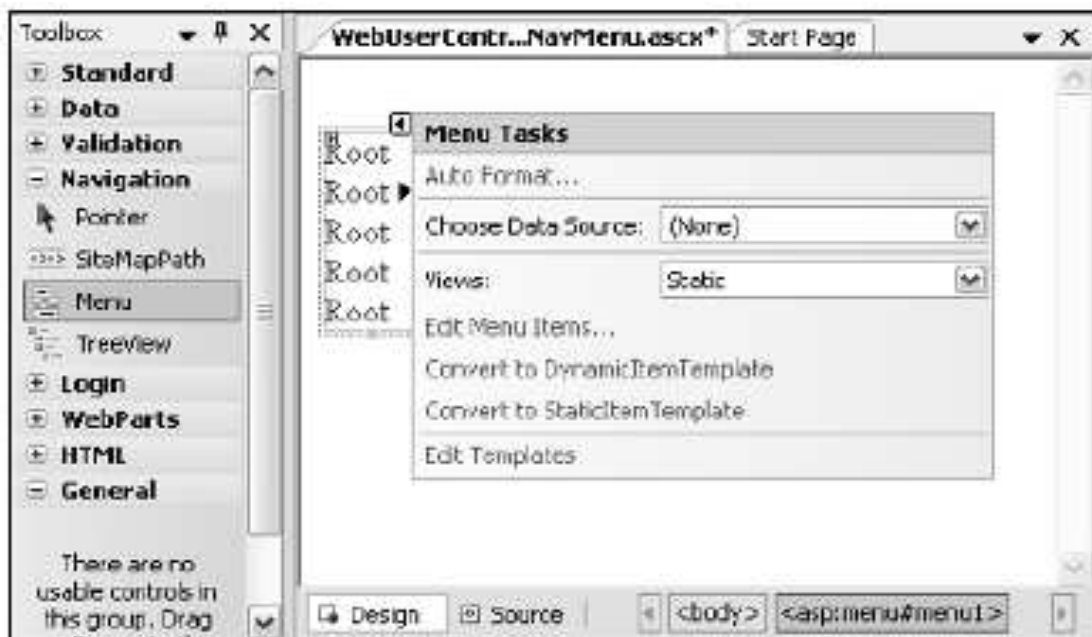
```

ΕΡΓΑΛΕΙΑ ΠΛΟΗΓΗΣΗΣ

Το site πρέπει να είναι εύκολο να πλοηγηθεί. Εάν ο ιστοχώρος είναι μεγάλος, η πλοήγηση μεταξύ των σελίδων που δεν χρησιμοποιούν τίποτα άλλο παρά links μπορεί να είναι κουραστική για το χρήστη. Τα εργαλεία Navigation controls προσφέρουν ένα πολύ καλό τρόπο για να δομήσουμε την πλοήγηση του χρήστη στο site μας. Μπορούμε να ορίσουμε όλα τα links σε ένα σημείο και μόνο. Με αυτό τον τρόπο αν προσθέσουμε μια νέα σελίδα (ή διαγράψουμε μια παλιά) δεν χρειάζεται να πάμε και να διαγράψουμε όλα τα links. Απλώς κρατάμε όλα τα links σε ένα αρχείο. Πρέπει να σκεφτούμε πως θα οργανώσουμε το site. Κατόπιν πρέπει να δημιουργήσουμε μια περιοχή πλοήγησης. Τέλος ίσως χρειαστεί να δημιουργήσουμε ένα ξεχωριστό αρχείο για αυτές τις περιοχές πλοήγησης. Τα δύο βασικά εργαλεία για την πλοήγηση είναι το **Menu** και το **TreeView**.

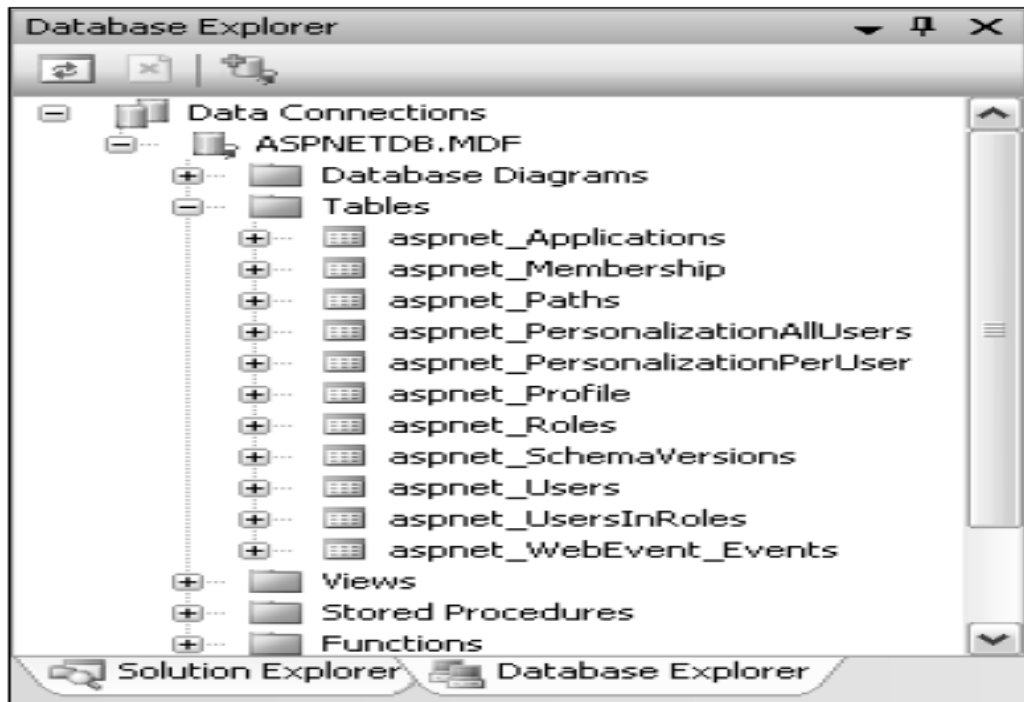
- Το Menu προσφέρει ένα απλό drop-down menu των links. Όταν το Menu control φαίνεται στην σελίδα στο browser, μόνο το **Home** link και το ΒΕΛΑΚΙ φαίνονται. Το Menu control είναι καλό για μικρά menu τα οποία τα θέλουμε να μην φαίνονται την ώρα που πλοηγούμαστε στον ιστοχώρο.
- Το TreeView control δείχνει την δομή πλοήγησης σε μορφή δέντρου. Ο χρήστης κάνει click + ή στο – και αναλόγως έχουμε εμφάνιση και απόκρυψη. Το TreeView control είναι καλό για μεγάλες και βαθιές πλοηγήσεις-διασχίσεις σε sites

Και το Menu αλλά και το TreeView μπορούν να χρησιμοποιηθούν και για στατικά αλλά και για δυναμικά δεδομένα (*static data* or *dynamic data*). Όταν χρησιμοποιείται το εργαλείο με static data, ορίζεται η πλοηγητική δομή σαν μέρος του control. Η μεθοδολογία είναι έτσι ευκολότερη διότι έτσι μπορούμε απλώς γεμίζοντας τα κενά του εργαλείου να ορίσουμε όλη τη δομή του site. Με τα dynamic data, αποθηκεύουμε τα δεδομένα σχετικά με την πλοήγηση σε ένα αρχείο το οποίο ονομάζεται *site map*, και το οποίο είναι εξωτερικό από το control. Το πλεονέκτημα με αυτή τη μεθοδολογία είναι ότι η δομή πλοήγησης του site είναι όλη αποθηκευμένη σε ένα σημείο. Επομένως αν χρειαστεί να αλλάξουμε τη δομή πλοήγησης του site αρκεί και μόνο να αλλάξουμε μόνο αυτό το εξωτερικό αρχείο χωρίς να πειράζουμε το control σε κάθε σελίδα του site. Τα εργαλεία του server TreeView και Menu βρίσκονται και τα δύο στην κατηγορία **Navigation category** του Toolbox. Είναι τόσο παρόμοια μεταξύ τους που διαβάζοντας της οδηγίες χρήσης του ενός μπορούμε επίσης να χρησιμοποιήσουμε μετά και το άλλο. Πρώτα περιγράφουμε τη χρήση τους με στατικά δεδομένα. Αυτός είναι ο ευκολότερος τρόπος να δημιουργήσουμε ένα «χάρτη» ειδικά όταν είναι μικρός και ειδικά όταν δεν γνωρίζουμε τίποτα σχετικά με την γλώσσα XML. Το μόνο μειονέκτημα με αυτή τη μέθοδο είναι ότι είναι χρονοβόρα. Και αν βάλουμε το control σε μια σελίδα μετά πρέπει να το κάνουμε copy-paste σε όλες τις άλλες σελίδες που θέλουμε να φαίνεται. Εκτός και εάν το βάλουμε στο Master Page. Αν δεν θέλουμε να το βάλουμε στο Master Page, μπορούμε να το τοποθετήσουμε σε ένα Web User Control και να το χρησιμοποιήσουμε όπου και όπως το χρειαζόμαστε μέσα στο site. Το παρακάτω σχήμα δείχνει την χρήση του εργαλείου αυτού



ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ SQL ΓΙΑ ΤΗΝ ΔΙΑΧΕΙΡΙΣΗ ΔΕΔΟΜΕΝΩΝ ΣΕ ΙΣΤΟΧΩΡΟΥΣ

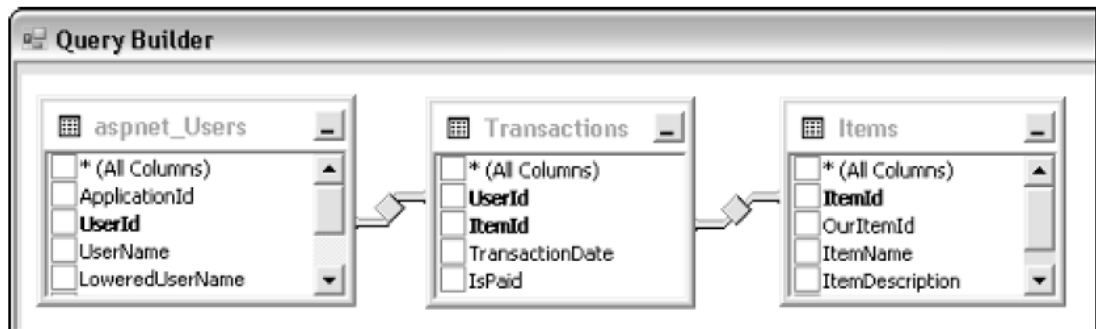
Το Visual Studio δέχεται αρχεία ACCESS (.mdb) ή SQL-Server (.mdf). Ο SQL-Server παρέχει καλύτερη κλιμάκωση και υποστηρίζει πολλαπλούς χρήστες. Η ACCESS έχει από την άλλη πλευρά διεπαφή χρήστη ενώ ο SQL-Server όχι. Είναι απλά ένας server. Ως DATA (δεδομένα) ορίζουμε όλη τη συλλογή πληροφοριών σε κάποια δομημένη μορφή. Στη δημιουργία ενός ιστοτόπου το Visual Studio δημιουργεί από μόνο του κάποιους πίνακες μέσα στη βάση δεδομένων. Όλοι οι πίνακες που δημιουργούνται αυτόματα ξεκινούν το όνομά τους με aspnet_ Δεν πρέπει να διαγράφουμε ή να προσθέτουμε δεδομένα στα αρχεία aspnet_. Πάντα πρέπει να χρησιμοποιούμε το WEB SITE ADMINISTRATION TOOL για τη διαχείριση των Membership Data. Ένα διάγραμμα της βάσης δεδομένων για ένα απλό ιστότοπο φαίνεται στο παρακάτω σχήμα.



Ο SQL-Server είναι ένα RDBMS (relational database management system). Όπως συζητήσαμε και πιο πάνω η διαχείριση των χρηστών του site γίνεται είτε με την χρήση του Web Administration Tool είτε με την βοήθεια του profile object για την εισαγωγή και αυτόματη διαχείριση των προσωπικών στοιχείων τους. Στις περιπτώσεις που χρειαζόμαστε να βάλουμε τα προϊόντα στο site πρέπει να δημιουργήσουμε δικούς μας πίνακες. Υπάρχουν όμως και περιπτώσεις που θα πρέπει να εξαγάγουμε δεδομένα από πίνακες. Ειδικότερα όταν ενδιαφερόμαστε να διαβάσουμε δεδομένα που προέρχονται από δύο ή περισσότερους πίνακες τότε για αυτές τις περιπτώσεις δημιουργούμε ένα νέο πίνακα. Στην πραγματικότητα δηλαδή δεν συνδέουμε πίνακες, δεν κοιτάμε καν στους πίνακες. Αντίθετα φτιάχνουμε ερωτήματα (queries) για να τους διαχειριστούμε, να τους ενημερώσουμε, να τους εμφανίσουμε, να εισάγουμε εγγραφές, να διαγράψουμε εγγραφές, κτλ. Κάθε φορά που εξαγάγουμε δεδομένα από πίνακες (π.χ. Users, Items, Transactions) το ερώτημά μας πρέπει να περιέχει πεδία και από τους 3 πίνακες. Επίσης πρέπει να ισχύουν τα εξής:

- Τα **primary** keys και τα **foreign** keys που συνδέουν τους πίνακες πρέπει να συνδέονται με join-lines στο query
- Κάθε πίνακας πρέπει να έχει ένα primary key

- Ο πίνακας των συναλλαγών για τους χρήστες πρέπει να περιέχει τουλάχιστον 2 πεδία που τα ονόματά τους και ο τύπος δεδομένων τους πρέπει να είναι ακριβώς ίδιος με τα αντίστοιχα στους πίνακες Users και Items



Οι βασικοί τύποι δεδομένων που υποστηρίζονται μέσω της SQL είναι οι εξής

- Text
- Number (Scalar)
- DateTime
- Boolean
- Binary (Εικόνες, ήχοι κτλ)
- Άλλοι (Ειδικοί τύποι δεδομένων : UniqueIdentifier, xml, timestamp, sql-variant)
- Unicode Text (16-bit / character)
- Non-Unicode Text (8-bit/character)

Οι παρακάτω πίνακες δίνουν μια ολοκληρωμένη εικόνα για όλους τους τύπους δεδομένων που υποστηρίζει η SQL μέσω του SQL-Server

<code>bigint</code>	Integers from -2^{63} ($-9,223,372,036,854,775,808$) to $2^{63} - 1$ ($9,223,372,036,854,775,807$)	8 bytes
<code>binary(n)</code>	Fixed-length binary data, where n can be any value from 1 to 8,000	n bytes
<code>bit</code>	1, 0, or Null	1 byte per 8 bit fields
<code>char(n)</code>	Fixed-length text, where n can be from 1 to 8,000	n bytes
<code>datetime</code>	Dates from January 1, 1753, through December 31, 9999, to 3.33 milliseconds of accuracy	8 bytes
<code>decimal(p, s)</code>	Numbers from $-10^{38} + 1$ to $10^{38} - 1$	Depends on p
<code>float(n)</code>	Numbers from $-1.79E + 38$ to $-2.23E - 38$, 0, and $2.23E - 38$ to $1.79E + 38$	Depends on n
<code>image</code>	Variable-length binary data to $2^{31} - 1$ (2,147,483,647) bytes	$length + 2$ bytes
<code>int</code>	Integers from -2^{31} ($-2,147,483,648$) to $2^{31} - 1$ (2,147,483,647)	4 bytes
<code>money</code>	Dollar amount from $-922,337,203,685,477.5808$ to $922,337,203,685,477.5807$	8 bytes
<code>nchar(n)</code>	Fixed-length Unicode text to 4,000 characters	$2 * length$
<code>ntext</code>	Variable-length Unicode data or binary data to $2^{30} - 1$ bytes	$2 * length$ bytes
<code>numeric(p, s)</code>	Numbers from $-10^{38} + 1$ to $10^{38} - 1$	Depends on p
<code>nvarchar(n)</code>	Variable-length Unicode text up to 4,000 characters	$2 * length + 2$ bytes
<code>nvarchar(MAX)</code>	Variable-length Unicode data to $2^{31} - 2$ bytes	$2 * length + 2$ bytes
<code>real</code>	$-1.18E - 38$, 0 and $1.18E - 38$ to $3.40E + 38$	4 bytes

<code>smalldatetime</code>	Dates from January 1, 1900, through June 6, 2079, accurate to 1 minute	4 bytes
<code>smallint</code>	Integers from -2^{15} ($-32,768$) to $2^{15} - 1$ ($32,767$)	2 bytes
<code>smallmoney</code>	Dollar amounts from $-214,748.3648$ to $214,748.3647$	4 bytes
<code>sql_variant</code>	Non-specific data type up to 8,016 bytes	Depends on data stored
<code>text</code>	Variable-length text to $2^{31} - 1$ (2,147,483,647 characters)	Depends on code page of server
<code>timestamp</code>	Time of last change to row	8 bytes
<code>tinyint</code>	Integers from 0 to 255	1 byte
<code>unique-identifier</code>	GUIDs (Globally Unique Identifiers)	16 bytes
<code>Varbinary (n)</code>	Variable-length binary data, where n can be 1 to 8,000	$length+2$ bytes
<code>varbinary (MAX)</code>	Variable-length data up to $2^{31} - 1$ bytes in length	$length + 2$ bytes
<code>varchar (n)</code>	Variable-length text where n can be from 1 to 8,000	$n+2$ bytes
<code>varchar (MAX)</code>	Variable-length text up to $2^{31} - 1$ bytes in length	$length + 2$ bytes
<code>xml</code>	Typed or untyped XML data	Depends on data stored

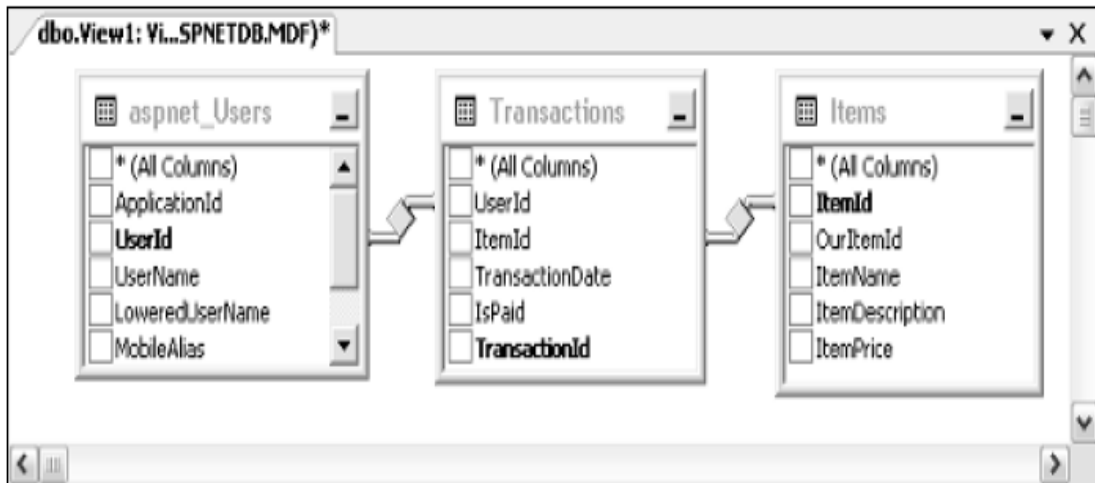
Συνδέοντας πίνακες

- Παρόλο που αποθηκεύονται τα δεδομένα στους πίνακες ο μόνος τρόπος να πάρετε μόνο αυτά που θέλετε είναι με την **Structured Query Language**, γνωστή ως **SQL**
- **Πααδείγματα**
 - `SELECT * FROM Items`
 - `SELECT UserId, UserName FROM aspnet_Users`

- `SELECT * FROM Items WHERE ItemId = 10002`
- `SELECT ItemName, ItemDescription FROM ItemsTable WHERE ItemId = 10002`
- Δεν χρειάζεται να γράψετε SQL ερωτήματα .
- Μπορείτε να χρησιμοποιήσετε το **Query Builder** για περίπλοκα SQL ερωτήματα. Το Query Builder επιτρέπει να επιλέξετε τι θέλετε να πάρετε από τους πίνακες.
- Επιλέξτε Options, το Query Builder γράφει το ερώτημα για εσάς. Το Query Builder εμφανίσετε αυτόματα όταν κάνετε μια ενέργεια από μια βάση.
- Θα χρησιμοποιούμε το Query Builder για να συνδέσουμε τους πίνακες σε *view*.
- Στον SQL Server, το view είναι ένα αποθηκευμένο query.
- Όταν ορίζετε το Membership, το VWD δημιουργεί αυτόματα διάφορα views. Εμφανίζονται κάτω από το Views στον Database Explorer. Το κάθε όνομα αρχίζει με **vw_aspnet_**

Δημιουργία ενός view

- Μην διαγράφετε αλλάξτε ή μετονομάσετε κάποιο view που το όνομά του ξεκινάει από vw_aspnet_. Αυτά τα views δημιουργούνται από το membership system, για το membership system.
- Ας δημιουργήσουμε ένα view με το Query Builder για να δείξουμε πως ενώνονται οι πίνακες users, transactions, και items, με όνομα **UsersAndItemsView**.
- Δεξί κλικ στο φάκελο των Views στον Database Explorer και επιλέξτε Add New View. Ανοίγει ένα παράθυρο διαλόγου Add Table.
- Το πρώτο βήμα στο να φτιάξουμε ένα query είναι μέλος να επιλέξουμε τους πίνακες από τους οποίους θα πάρει δεδομένα το query. Αν υπάρχει σχέση many-to-many στους επιλεγμένους πίνακες το view πρέπει να περιέχει και τον πίνακα **Transactions** που δείχνει αυτές τις σχέσεις μεταξύ των πινάκων.
- Για να προσθέσετε ένα πίνακα στο query, κάντε κλικ στο όνομά του στο παράθυρο διαλόγου Add Table, και πατήσετε OK.
- (Για παράδειγμα, προσθέστε τον **aspnet_Users**, **Transactions**, και **Items**) Πατήστε το κουμπί Close στο παράθυρο διαλόγου Add Tables αφού επιλέξετε τους πίνακες.



- Επιλέξτε ποια πεδία θέλετε να δείτε από αυτούς τους πίνακες. Μπορείτε να επιλέξετε ποια πεδία θέλετε να δείτε στο view επιλέγοντάς. Κάθε πεδίο που επιλέγετε εμφανίζεται στο Criterion pane, κάτω από το Diagram pane.
- Κάτω από το Criterion pane είναι το Show SQL pane, που δείχνει το ερώτημα SQL που δημιουργεί το Query Builder.

ΚΑΛΑΘΙ ΑΓΟΡΩΝ ΚΑΙ ΠΑΡΑΓΓΕΛΙΣ

Υπάρχουν πολλά στάδια στη διαδικασία μιας παραγγελίας, ξεκινώντας με την σελίδα της που επιτρέπει στους πελάτες να προσθέσουν είδη στο καλάθι αγορών. Όταν οι πελάτες επιλέξουν είδη μπορούν να κάνουν checkout, και να ορίσουν διεύθυνση αποστολής, στοιχεία πιστωτικής κάρτας και άλλες λεπτομέρειες. Στο τέλος δημιουργείται η παραγγελία στη βάση δεδομένων και όλα τα στοιχεία της παραγγελίας αποθηκεύονται εκεί.

Πριν δημιουργήσουμε μια σελίδα παραγγελίας, πρέπει να ξεκαθαρίσουμε την διαδικασία παραγγελίας ειδών. Χρειαζόμαστε τα παρακάτω εργαλεία για την σωστή διαχείριση του ιστοχώρου μας όσον αφορά τις παραγγελίες και τις συναλλαγές των πελατών:

- Μια σελίδα για την παραγγελία, όπου μπορούμε να επιλέγουμε διάφορα είδη και προϊόντα για αγορά.
- Ένα shopping cart, για να αποθηκεύουμε τα είδη προς αγορά
- Μια σελίδα για τη συλλογή δεδομένων αποστολής και πληρωμής με πιστωτική κάρτα.

- Θα πρέπει να έχουμε μια ξεχωριστή σελίδα για παραγγελία και τουλάχιστον ξεχωριστή σελίδα για τον κατάλογο των προϊόντων. Αν όχι πρέπει να προσθέσουμε δίπλα στο κάθε είδος ένα σύνδεσμο με το καλάθι αγορών.
- Εναλλακτικά μπορούμε να προσθέσουμε ένα πλαίσιο κειμένου για να προσθέτει ο πελάτης την ποσότητα των ειδών που θέλει να αγοράσει.

Πρέπει επίσης να αποφασίσουμε που θα αποθηκεύονται οι παραγγελίες. Συνήθως το πιο δημοφιλές εργαλείο που χρησιμοποιείται είναι τα shopping carts. Οι πληροφορίες ενός καλαθιού αγορών μπορούν να αποθηκεύονται:

- **Σε μια database**—Όταν ο χρήστης προσθέτει είδη στο καλάθι του, αποθηκεύονται σε ένα πίνακα της βάσης δεδομένων (π.χ. OrderItems). Όταν επιβεβαιωθεί η παραγγελία, αντιγράφονται οι εγγραφές στον πίνακα της παραγγελίας. Μπορεί να προκύψει όμως το εξής πρόβλημα: Αν ο χρήστης δεν επιβεβαιώσει την παραγγελία, υπάρχουν εγγραφές που πρέπει να διαγραφούν.
- **Σε ένα Profile** —Το Profile της ASP.NET 2.0 επιτρέπει την αποθήκευση δεδομένων σε σχέση με έναν χρήστη. Ενδείκνυται για μακράς διάρκειας αποθήκευση δεδομένων.
- **Σε ένα Session** —Το Session περιέχει δεδομένα για ένα ενεργό session. Ξεκινάει όταν γίνεται η πρόσβαση στο site και τελειώνει με την έξοδο.

ΚΕΦΑΛΑΙΟ IV: ΥΛΟΠΟΙΗΣΗ ΕΝΟΣ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ ΜΕ ΘΕΜΑ «ΚΑΤΟΙΚΙΔΙΑ ΖΩΑ» e-Pet-Shop

ΕΡΓΑΛΕΙΑ ΥΛΟΠΟΙΗΣΗΣ

Η υλοποίηση του ηλεκτρονικού καταστήματος έγινε με την χρήση του ολοκληρωμένου εργαλείου ανάπτυξης λογισμικού Microsoft Visual Studio .Net. Η γλώσσα προγραμματισμού που επιλέχθηκε για την υλοποίηση των λειτουργιών στο κατάστημα ήταν η C#. Το ολοκληρωμένο λογισμικό ανάπτυξης εφαρμογών Microsoft Visual Studio στάθηκε πολύ χρήσιμο για την πραγματοποίηση του project δεδομένου του γεγονότος ότι περιέχει ενσωματωμένες μια πλειάδα εργαλείων έτοιμα προς χρήση για διάφορες λειτουργίες όπως για παράδειγμα το στοιχείο ελέγχου δημιουργίας νέων χρηστών, τα data grids που μπορούν να χρησιμοποιηθούν για την παρουσίαση των αποτελεσμάτων από ένα query της βάσης δεδομένων με συστηματικό τρόπο κλπ. Επιπροσθέτως το εργαλείο αυτό επιτρέπει το σχεδιασμό του front end interface των χρηστών χρησιμοποιώντας εντελώς νέες τεχνολογίες όπως για παράδειγμα την τεχνολογία AJAX. Συμπληρωματικά, οι νέες εκδόσεις του λογισμικού έρχονται με ενσωματωμένη μια απλή αλλά λειτουργική έκδοση του SQL server που ονομάζεται SQL-EXPRESS και μπορεί να χρησιμοποιηθεί για την δημιουργία των απαραίτητων πινάκων της βάσης δεδομένων που περιέχουν τα δεδομένα του ηλεκτρονικού καταστήματος. Αυτά τα δεδομένα είναι είτε δεδομένα προϊόντων, είτε δεδομένα χρηστών αλλά είτε δεδομένα επίσης που αφορούν την διαχείριση του καταστήματος, (συναλλαγές, ενημερωτικά δελτία κλπ).

Το βασικότερο παραδοτέο της παρούσας πτυχιακής εργασίας πέρα από το θεωρητικό της κομμάτι που αποτελεί μια σύνοψη και επισκόπηση των κυριότερων μεθοδολογιών ανάπτυξης του ηλεκτρονικού εμπορίου, είναι η πειραματική επισφράγιση και απόδειξη για το πώς αυτές οι μεθοδολογίες υλοποιούνται. Μέχρι σήμερα υπάρχουν τόσες πολλές τοποθεσίες ιστοτόπων στο διαδίκτυο που εγγίζουν ίσως και τον αριθμό του μισού δισεκατομμυρίου. Γνωρίζουμε ότι οι περισσότερες περιπτώσεις από αυτές αποτελούν παραδείγματα εντελώς πειραματικών προσπαθειών και στις οποίες δεν θα επικεντρωθούμε. Εκείνες όμως οι περιπτώσεις ιστοτόπων οι

οποίες αποτελούν κομμάτι του extranet μιας επιχείρησης και βασικό εργαλείο μέσα από το οποίο παράγουν οι επιχειρήσεις πλούτο είναι και το πιο ενδιαφέρον κομμάτι.

Ένα από τα πρώτα και βασικά ερωτήματα που μας απασχόλησε από την αρχή της προσπάθειας αυτής ήταν και κομμάτι της περιοχής υλοποίησης καθώς και ο τρόπος με τον οποίο θα έπρεπε να προσεγγίσουμε την υλοποίηση αυτή. Θέλαμε από την αρχή να δημιουργήσουμε ένα ολοκληρωμένο ιστότοπο που να αφορά την πώληση μέσω του διαδικτύου προϊόντων αλλά επίσης θέλαμε μια τέτοια προσπάθεια να είναι όσο το δυνατό πιο ολοκληρωμένη και να αγγίζει όλες τις πτυχές μιας επιχείρησης τόσο στην προώθηση ορισμένων προϊόντων όσο και στο κομμάτι της διαχείρισης των προϊόντων αυτών από θέμα αποθήκης και λογιστηρίου. Για αυτό ακριβώς το λόγο και επικεντρωθήκαμε επίσης όχι μόνο στο κομμάτι υλοποίησης που φαίνεται στον επισκέπτη στις ιστοσελίδας αλλά και στο διαχειριστικό κομμάτι της επιχείρησης που αφορά την ενοποίηση νέων προϊόντων, την αλλαγή τιμών στα προϊόντα, την δημιουργία μεθοδολογιών μαζικής ενημέρωσης των πελατών του ιστοτόπου όσον αφορά τα νέα προϊόντα της επιχείρησης κλπ.. Επομένως δώσαμε μεγάλη προσοχή στα κομμάτια υλοποίησης που συνήθως δεν είναι ορατά στους επισκέπτες μιας ιστοσελίδας αλλά επίσης δεν είναι ορατά και στους καθημερινούς προγραμματιστές. Σε ότι αφορά στην επιλογή της προγραμματιστικής μεθοδολογίας για την πειραματική υλοποίηση της πτυχιακής μας δυσκολευτήκαμε επίσης να διαλέξουμε μια συγκεκριμένη μεθοδολογία. Αν παρατηρήσει κανείς στο διαδίκτυο θα ανακαλύψει ότι οι πιο κυρίαρχες τεχνολογίες τα τελευταία χρόνια είναι οι εξής:

- Php με την χρήση της html, flash και MySQL
- CMS (όπως για παράδειγμα μέσω Joomla, Drupal και άλλων πακέτων διαχείρισης περιεχομένου) και τέλος
- ASP.NET με την χρήση Silverlight, SQL-Server και διαφόρων εργαλείων της Microsoft Corporation

Η χρήση της πρώτης επιλογής θα μας έδινε την δυνατότητα να χρησιμοποιήσουμε μια ευρέως γνωστή γλώσσα προγραμματισμού καθώς και όλα τα απαραίτητα εργαλεία τα οποία είναι ελεύθερα προς χρήση. Επίσης το γεγονός ότι η μεγάλη πλειοψηφία των ιστοσελίδων στο διαδίκτυο χρησιμοποιούν αυτή την επιλογή μας λέει ότι είναι μια από τις πιο καλές επιλογές. Όμως θέλαμε να κάνουμε κάτι το οποίο δεν θα πατούσε πάνω στα περισσότερα πρότυπα των παλαιών τεχνολογιών αλλά θα χρησιμοποιούσε

νέα και πιο σύγχρονα εργαλεία προγραμματισμού. Ένας ακόμη λόγος για τον οποίο δεν θέλαμε να διαλέξουμε μια σχετικά παλαιή τεχνολογία είναι να μπορέσουμε μέσα από την παρούσα πτυχιακή εργασία να αποκτήσουμε την εκμάθηση ενός νέου εργαλείου με σκοπό να είναι πιο προσιτό τα επόμενα χρόνια και επίσης αυτό το εργαλείο να προέρχεται από μια εταιρία λογισμικού που να είναι φημισμένη.

Η δεύτερη επιλογή που είχαμε ήταν να διαλέξουμε την μεθοδολογία υλοποίησης μέσω πακέτων διαχείρισης περιεχομένου (content management systems, CMS) όπως για παράδειγμα τα πακέτα Joomla, Drupal κλπ. Όμως αυτά τα πακέτα δημιουργίας ιστοσελίδων έχουν ιδιαιτερότητες που είναι προσανατολισμένες περισσότερο ως προς την διαχείριση περιεχομένου πληροφορίας. Από την άλλη πλευρά επίσης εμφανίζονται κατά πού λιγότερο ευέλικτα ως προς την διαχείριση εικόνων τη δημιουργία δυναμικών σελίδων και λοιπά.

Η τελευταία επιλογή ήταν και αυτή που ακολουθήσαμε. Το ολοκληρωμένο πακέτο ανάπτυξης λογισμικού της Microsoft που ονομάζεται Visual Studio .NET δεν παρέχει απλώς τη δυνατότητα δημιουργίας ενός ολοκληρωμένου ιστότοπου αλλά επίσης δίνει τη δυνατότητα στον προγραμματιστή μέσω μιας πλειάδας ενσωματωμένων εργαλείων να αναπτύξει με εύκολο τρόπο:

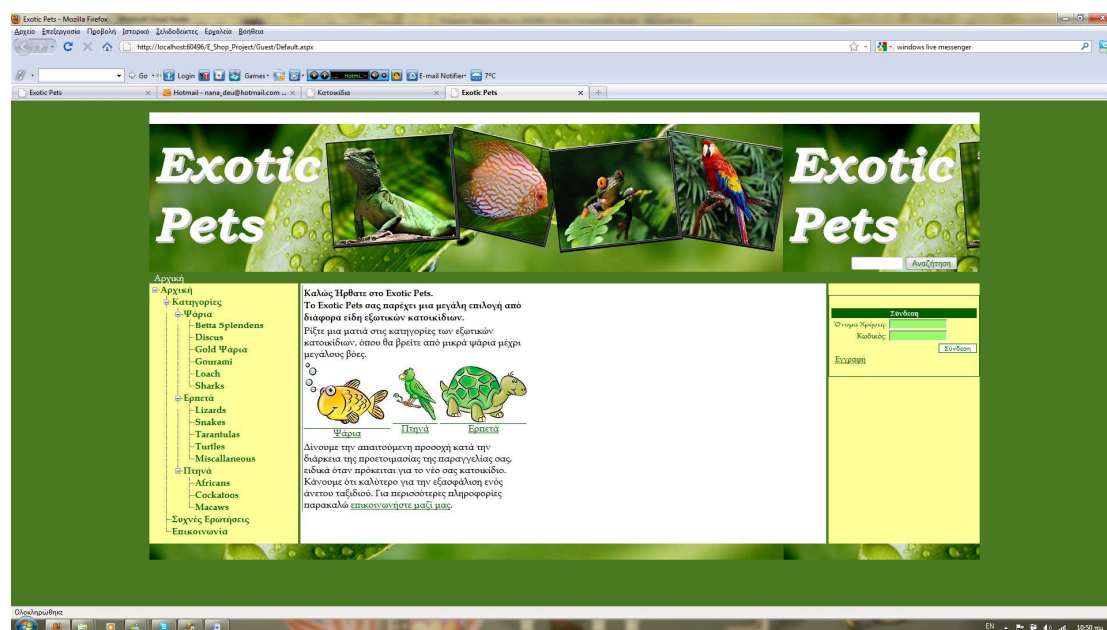
1. Δυναμικές ιστοσελίδες
2. Μια ολοκληρωμένη βάση δεδομένων
3. Εύκολη Διαχείριση χρηστών στο site
4. Πλοήγηση στον ιστότοπο μέσω ενσωματωμένων εργαλείων πλοήγησης
5. Εύκολη διατήρηση των συνεδριών συναλλαγής των χρηστών με την χρήση των εργαλείων session και cookies

Για τους παραπάνω λόγους λοιπόν καταλήξαμε στην τελευταία επιλογή. Στις παρακάτω ενότητες αυτού του κεφαλαίου θα περιγράψουμε την φιλοσοφία του ηλεκτρονικού καταστήματος που αναπτύξαμε, τις διάφορες λειτουργικότητες που παρέχονται στο χρήστη από πλευράς πλοήγησης και εγγραφής, την δημιουργία δυναμικών σελίδων που εμφανίζουν τις διάφορες κατηγορίες προϊόντων κλπ. Ιδιαίτερη όμως μνεία θα γίνει για το κομμάτι του διαχειριστή του ηλεκτρονικού καταστήματος στο οποίο δώσαμε και ιδιαίτερη βαρύτητα. Αυτό το κομμάτι περιλαμβάνει επί της ουσίας μια διαφορετική ομάδα ιστοσελίδων οι οποίες είναι

αφιερωμένες στη διαχείριση των χρηστών, στη διαχείριση των προϊόντων καθώς και σε περαιτέρω δυνατότητες όπως για παράδειγμα την αποστολή μαζικών ηλεκτρονικών μηνυμάτων, την αποστολή newsletter σε εγγεγραμμένους χρήστες κλπ.

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΟΜΑΔΑ ΤΩΝ ΜΗ ΕΓΓΕΓΡΑΜΜΕΝΩΝ ΧΡΗΣΤΩΝ

Το ηλεκτρονικό κατάστημα που αναπτύξαμε παρέχει ολοκληρωμένες υπηρεσίες που αφορούν την πώληση κατοικίδιων. Αν και το συγκεκριμένο προϊόν δεν παρέχεται ως και το πιο ενδεδειγμένο για την δημιουργία μιας πειραματικής εφαρμογής, εν τούτοις, προσπαθήσαμε να ξεφύγουμε από τα τετριμμένα όπως για παράδειγμα, τις ηλεκτρονικές συσκευές, τα ηλεκτρονικά βιβλιοπωλεία κλπ. Ένας περαιτέρω λόγος που διαλέξαμε την συγκεκριμένη ομάδα προϊόντων πώλησης είναι επίσης η ιδιαίτερη μας αγάπη στα ζώα καθώς και η προώθηση αυτής της φιλοσοφίας στην ελληνική κοινωνία. Η αρχική σελίδα που ονομάζεται Default.aspx φαίνεται στο παρακάτω σχήμα.

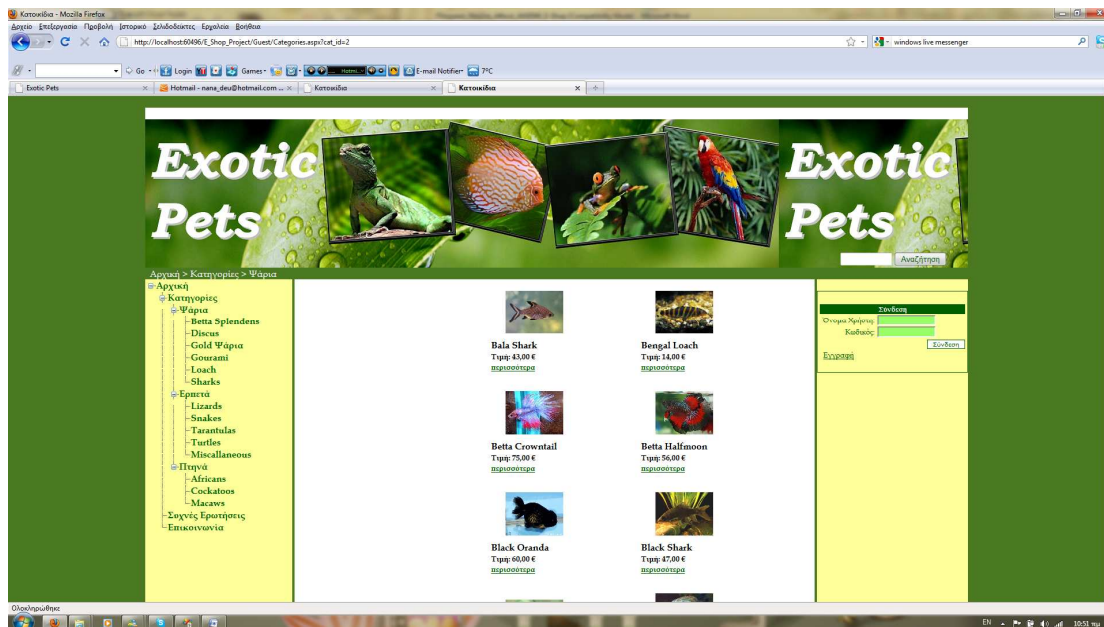


Σε αυτό σχήμα φαίνεται το βασικό template της εφαρμογής καθώς και επίσης δύνεται η βασική δυνατότητα στους μη εγγεγραμμένους χρήστες να περιηγηθούν μέσω του μενού δεινδρικής μορφής στα αριστερά σε διάφορες κατηγορίες προϊόντων. Όμως μη εγγεγραμμένοι χρήστες δεν μπορούν να επιλέξουν κανένα προϊόν προς αγορά αν δεν προσθέσουν τα προσωπικά τους στοιχεία και πάρουν ένα όνομα χρήστη και έναν κωδικό πρόσβασης στον ιστότοπο. Αυτό γίνεται όπως φαίνεται στο παραπάνω σχήμα

στα δεξιά της αρχικής σελίδας. Ενσωματώσαμε στην αρχική σελίδα ένα login control το οποίο είναι ένα βασικό εργαλείο από την εργαλειοθήκη του Visual Studio .NET για την εγγραφή νέων χρηστών στον ιστότοπο. Μια άλλη πολύ βασική λειτουργία που επίσης εμφανίζεται στην αρχική σελίδα και η οποία αφορά επίσης στους ανώνυμους (μη –εξουσιοδοτημένους) χρήστες είναι η δυνατότητα αναζήτησης προϊόντων μέσα στον ιστότοπο. Αυτό γίνεται με την μετάβαση στην σελίδα search.aspx η οποία επί της ουσίας παίρνει την ομάδα από τις λέξεις κλειδιά που τοποθετεί ο χρήστης στο text box της υποτυπώδους μηχανής αναζήτησης και κάνει ένα SELECT query statement στη βάση δεδομένων και στον πίνακα Pets (Products). Εκείνο που ψάχνουμε για ταυτοποίηση είναι οι λέξεις που τοποθέτησε ο χρήστης με λέξεις από το κομμάτι της περιγραφής του κάθε προϊόντος στο συγκεκριμένο πίνακα. Ο ανώνυμος χρήστης μέσα από το εργαλείο πλοήγησης που παρέχουμε μπορεί να δει τον ηλεκτρονικό κατάλογο προϊόντων, να περιηγηθεί σε κάθε μία από τις διάφορες κατηγορίες προϊόντων και να δει για αυτές τις κατηγορίες διαφορετικές σελίδες που αφορούν κατοικίδια. Θα πρέπει να επισημάνουμε ότι επί της ουσίας δεν υπάρχουν διαφορετικές ιστοσελίδες στατικού χαρακτήρα για κάθε μια από τις διαφορετικές κατηγορίες προϊόντων. Αντιθέτως η τεχνική που ακολουθήσαμε είναι να δημιουργήσουμε μια δυναμική σελίδα για αυτό το λόγο. Ο τρόπος με τον οποίο το κάναμε είναι μέσω της χρήσης ενός sitemap αρχείου.

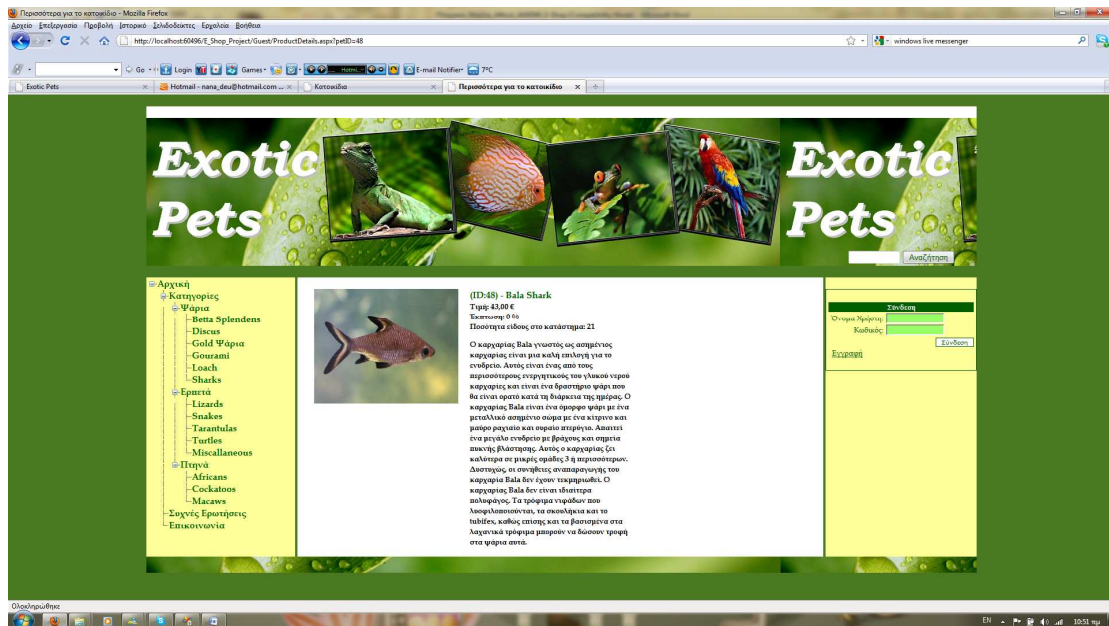
Όπως έχουμε αναφέρει και σε προηγούμενο κεφάλαιο η χρήση των sitemap αρχείων δίνει την δυνατότητα στον προγραμματιστή να μπορέσει να κάνει αφενός το μενού πλοήγησης αλλά και να συγχρονίσει το μενού πλοήγησης με τρόπο ώστε να εμφανίζει μόνο τα προϊόντα της κατηγορίας που επιλέγεται. Κάθε κατηγορία προϊόντων αποτελεί ένα node στο αρχείο sitemap της εφαρμογής μας που δεν είναι τίποτε άλλο παρά ένα xml αρχείο. Κάθε λοιπόν εγγραφή στο αρχείο αυτό συνοδεύεται και με ένα μοναδικό αριθμό (στην περίπτωση μας ονομάζεται sub_id). Αυτός ο μοναδικός ακέραιος αποτελεί επίσης και ένα ξεχωριστό πεδίο στον πίνακα των προϊόντων στη βάση δεδομένων. Η επιλογή λοιπόν από το μενού πλοήγησης στη μέθοδο Form_load() της δυναμικής σελίδας πηγαίνει και κάνει ένα query στη βάση δεδομένων για τα προϊόντα της κατηγορίας αυτής. Το αποτέλεσμα του query είναι τα προϊόντα της συγκεκριμένης κατηγορίας και τα οποία γεμίζουν ένα list view control το οποίο είναι ένα βασικό εργαλείο για την παράσταση των δεδομένων από μια βάση δεδομένων στο Visual Studio.NET. Τα προϊόντα λοιπόν εμφανίζονται μέσω του list view control το οποίο είναι γενικά κενό (χωρίς δεδομένα) μέχρι τη στιγμή που ο

χρήστης θα διαλέξει μια κατηγορία προϊόντων από το μενού πλοήγησης. Ένα τέτοιο σχήμα το οποίο δείχνει τον τρόπο με τον οποίο εμφανίζονται τα προϊόντα του ηλεκτρονικού καταλόγου φαίνεται στο παρακάτω σχήμα

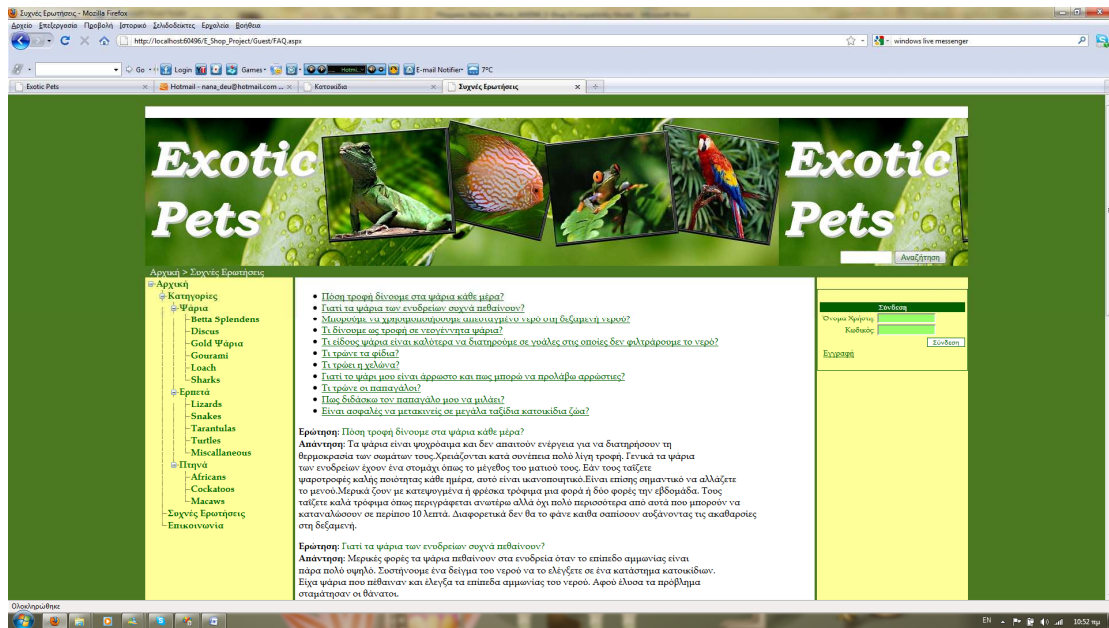


Αν ο χρήστης επιλέξει ένα συγκεκριμένο κατοικίδιο μέσω της εικόνας thumbnail τότε η πλοήγηση της εφαρμογής οδηγείται στην ιστοσελίδα ProductDetails.aspx. Αυτή η ιστοσελίδα είναι και αυτή μια δυναμική ιστοσελίδα. Δηλαδή δεν έχουμε μια διαφορετική ιστοσελίδα που να εμφανίζουμε την αναλυτική περιγραφή κάθε κατοικίδιου αλλά έχουμε μια και μοναδική για όλα τα κατοικίδια.

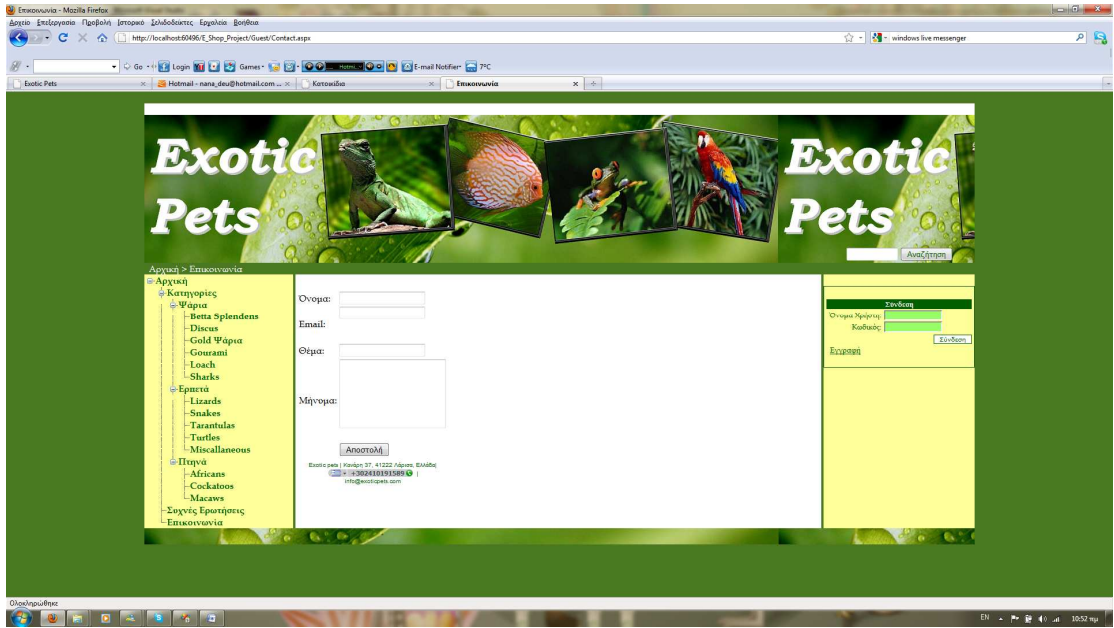
Πιο αναλυτικά κάθε κατοικίδιο (προϊόν) συνοδεύεται με ένα μοναδικό προσδιοριστικό στη βάση που ονομάζεται petID. Το petID χρησιμοποιείται ως το κλειδί για την αναζήτηση του κατοικίδιου στη βάση και την εύρεση όλων των ιδιαίτερων του χαρακτηριστικών όπως η αναλυτική περιγραφή του, η τιμή του, μια μεγαλύτερη φωτογραφία που μπορεί να δει ο χρήστης και άλλα. Η προγραμματιστική τεχνική που χρησιμοποιείται εδώ είναι ακριβώς η ίδια τεχνική που περιγράψαμε παραπάνω όσον αφορά την αναζήτηση και την εμφάνιση προϊόντων ανά κατηγορία επομένως δεν θα την περιγράψουμε ξανά εδώ. Ένα σχήμα (snapshot) για την συγκεκριμένη λειτουργία του ηλεκτρονικού καταστήματος που αφορά την παραπάνω συζήτησή μας φαίνεται στο παρακάτω σχήμα



Ο ανώνυμος χρήστης έχει δύο ακόμη δυνατότητες τις οποίες και περιγράφουμε παρακάτω. Η πρώτη δυνατότητα αφορά την εύρεση βασικών πληροφοριών που αφορούν την προστασία των κατοικίδιων, τον τρόπο περιποίησής τους και διάφορα μυστικά που επιτρέπουν την δυνατότητα στο κατοικίδιο να ζήσει όσο το δυνατό περισσότερο. Είναι το κομμάτι των συχνών ερωτήσεων ή αλλιώς Frequently Asked Questions (FAQ). Για να μπορέσει όμως να εμπλουτισθεί το κομμάτι των συχνών ερωτήσεων και απαντήσεων πρέπει να δίνεται η δυνατότητα σε χρήστες (και ιδιαίτερα και σε μη εγγεγραμμένους χρήστες) να ρωτούν διάφορες ερωτήσεις και να παίρνουν σύντομα εκτενείς και λεπτομερείς απαντήσεις από τον διαχειριστή του ηλεκτρονικού καταστήματος. Για αυτό το λόγο συμπεριλάβαμε σε αυτό το κομμάτι των μη εγγεγραμμένων χρηστών και την δεύτερη λειτουργία που προαναφέραμε και αφορά στην δυνατότητα που έχουν να επικοινωνήσουν με τον ιστότοπο μέσω e-mail και να στείλουν τις διάφορες ερωτήσεις τους. Όσον αφορά την πρώτη λειτουργικότητα αυτό φαίνεται στο παρακάτω σχήμα

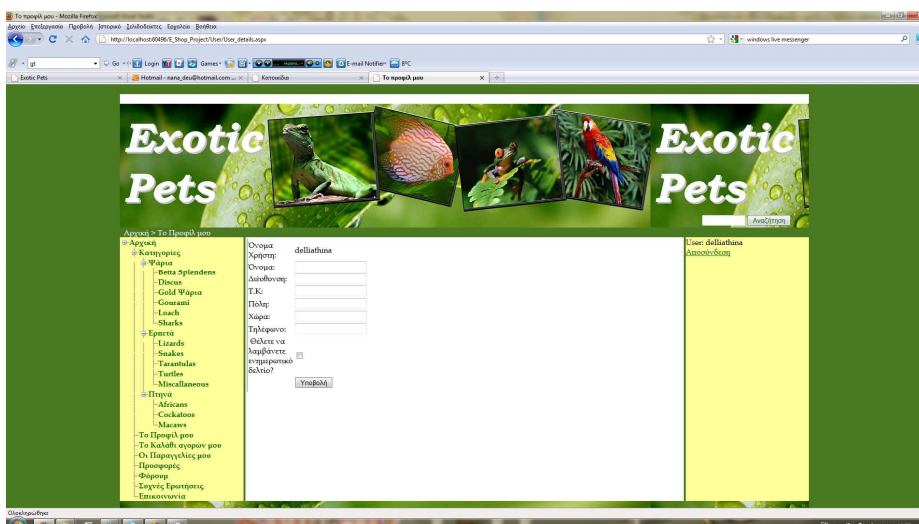
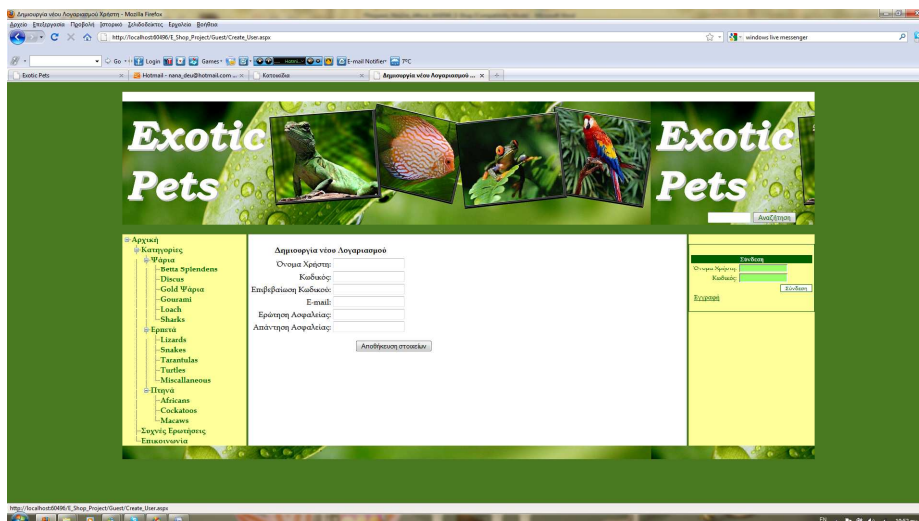


Η δυνατότητα αποστολής στο ηλεκτρονικό κατάστημα ενός ηλεκτρονικού μηνύματος προϋποθέτει την δημιουργία ενός λογαριασμού ηλεκτρονικού ταχυδρομείου. Για την παρούσα εργασία χρησιμοποιήσαμε τον προσωπικό μας λογαριασμό αλλά αυτό είναι πάρα πολύ εύκολο να αλλάξει στον κώδικα. Αυτό που έχει ιδιαίτερη σημασία να αναφέρει κανείς είναι ο τρόπος προγραμματισμού για την συγκεκριμένα λειτουργία και οποίος είναι σχετικά εύκολος διότι δίνεται η δυνατότητα να χρησιμοποιήσουμε τα ενσωματωμένα objects της γλώσσας C# για αυτό τον λόγο. Αρκεί και μόνο να παραμετροποιήσουμε το συγκεκριμένο αντικείμενο έτσι ώστε να επιδέχεται τον συγκεκριμένο smtp και imap server που αφορά τον εξυπηρετητή ηλεκτρονικού ταχυδρομείου που διαλέγουμε να χρησιμοποιήσουμε για την λήψη μηνυμάτων. Αυτή λοιπόν η δυνατότητα φαίνεται στο παρακάτω σχήμα.

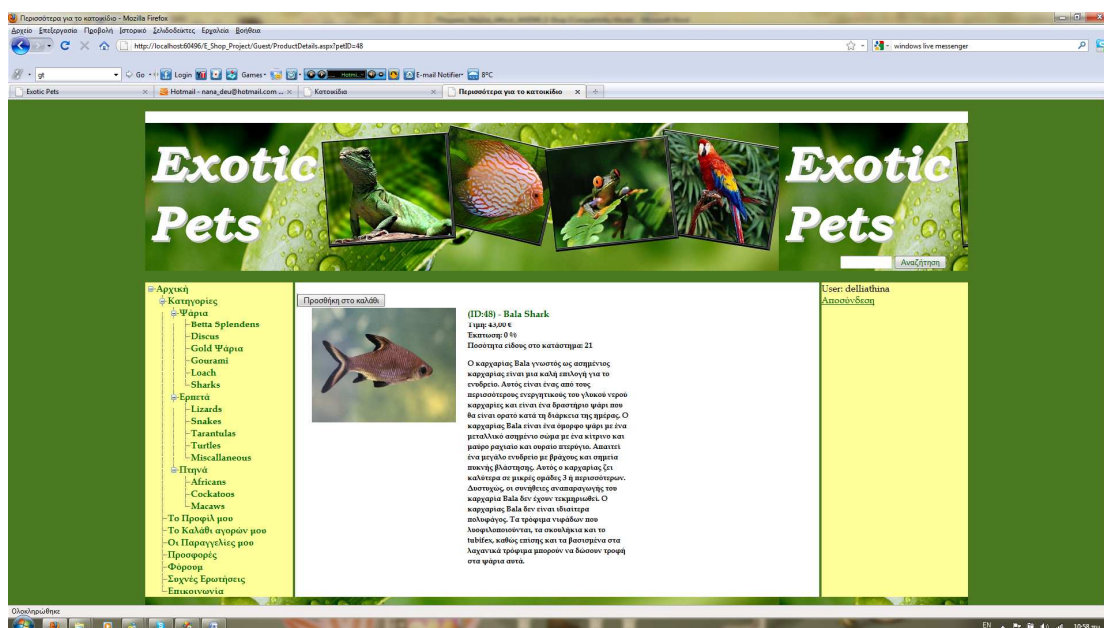


ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΟΜΑΔΑ ΤΩΝ ΕΓΓΕΓΡΑΜΜΕΝΩΝ ΧΡΗΣΤΩΝ

Για να γίνει κάποιος ανώνυμος χρήστης χρήστης-μέλος του site, πηγαίνει στο Login Control στην δεξιά περιοχή και πρέπει να πατήσει εγγραφή. Τότε μεταφέρεται στην σελίδα Create_User.aspx. Η σελίδα αυτή χρησιμοποιείται για την ένταξη των προσωπικών δεδομένων του χρήστη στη βάση δεδομένων. Παρατηρείστε ότι τα στοιχεία του χρήστη που κατ' αρχήν απαιτούνται είναι μόνο το user_name και το password καθώς επίσης και το e-mail του χρήστη. Το σύστημα κατόπιν οδηγεί τον χρήστη στη σελίδα user_details.aspx με βάση την οποία θα εντάξουμε στη βάση δεδομένων και τα υπόλοιπα προσωπικά στοιχεία όπως για παράδειγμα η διεύθυνση, τηλέφωνα επικοινωνίας, αριθμός πιστωτικής κάρτας κλπ. Στα δύο επόμενα σχήματα φαίνονται δύο απεικονίσεις των σελίδων στις οποίες αναφερθήκαμε παραπάνω.

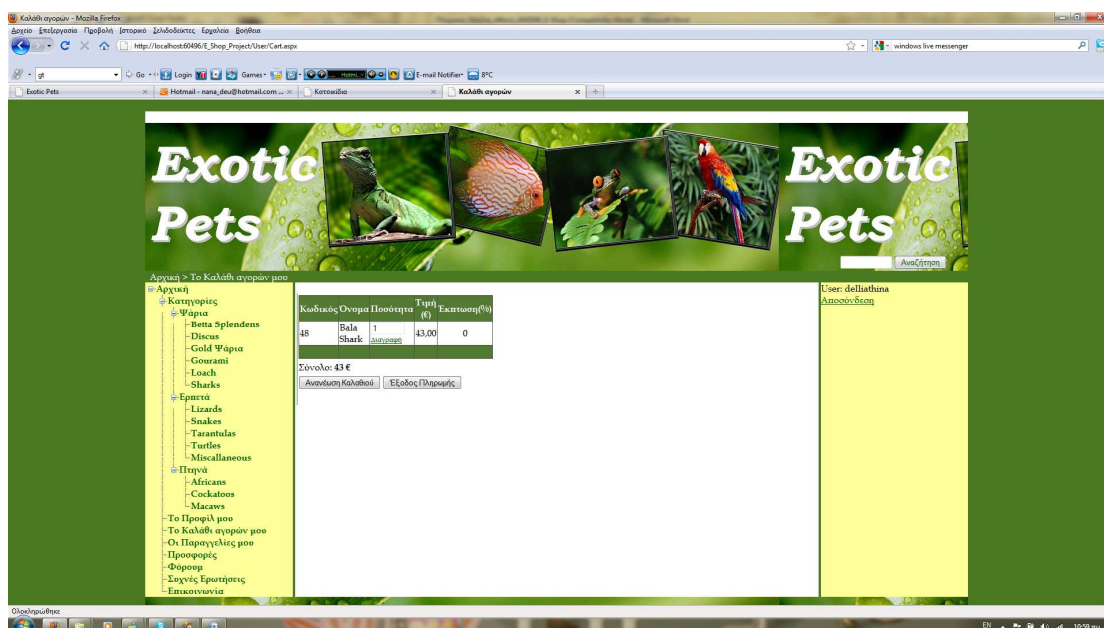


Από εκεί και μετά πλέον ως χρήστης-μέλος ο κάθε χρήστης έχει την δυνατότητα να αλλάξει το προφίλ του. Η αλλαγή προφίλ πρέπει να γίνεται διότι υπάρχουν περιπτώσεις όπου ο χρήστης μπορεί να αλλάξει τα προσωπικά του στοιχεία όπως για παράδειγμα η διεύθυνση ή το τηλέφωνο. Όμως ο βασικότερος λόγος για τον οποίο χρήστες επισκέπτονται ένα ηλεκτρονικό κατάστημα είναι η επιθυμία για αγορές. Για αυτό ακριβώς το λόγο εμφανίζει ή διαλέγει προϊόντα και μελετά τις λεπτομέρειες των προϊόντων χρησιμοποιώντας το μενού επιλογής και διαλέγοντας από τον κατάλογο των προϊόντων κάποιο συγκεκριμένο. Για αυτό το λόγο χρησιμοποιούμε την σελίδα Product_Details.aspx η οποία όπως αναφέραμε και στην προηγούμενη ενότητα δημιουργείται δυναμικά (είναι άδεια και γεμίζει με δεδομένα που αφορούν την επιλογή του χρήστη για την εμφάνιση κάποιου συγκεκριμένου προϊόντος). Ένα βασικό εργαλείο που προγραμματίσαμε στην σελίδα αυτή είναι η χρήση του καλαθιού αγορών μέσω του πατήματος του κουμπιού «προσθήκη στο καλάθι» όπως φαίνεται παρακάτω.



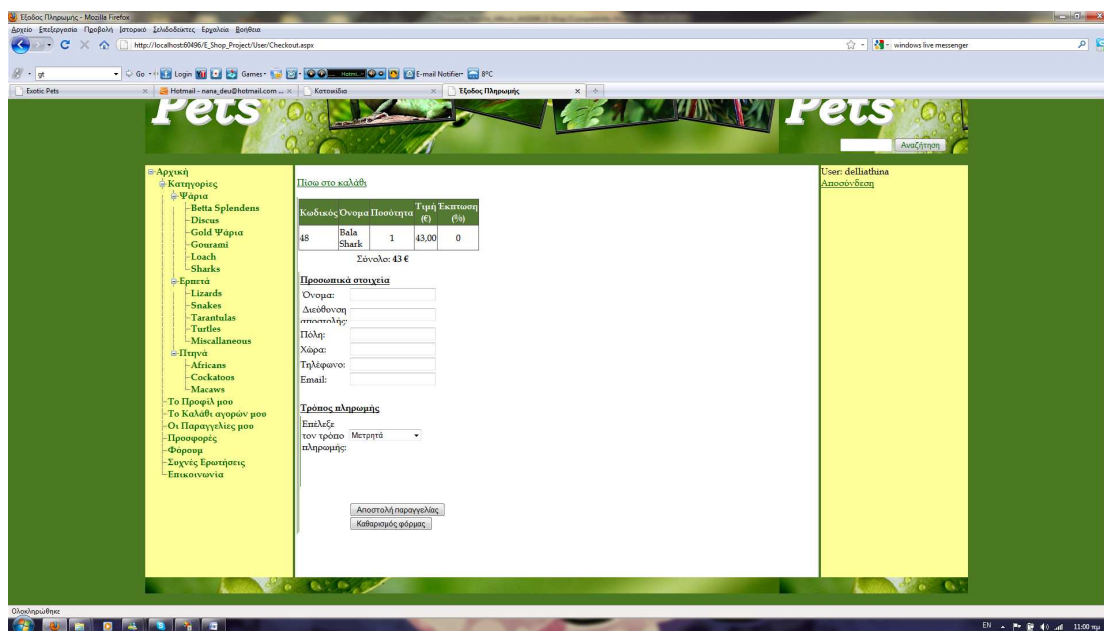
Κάθε φορά που ο χρήστης επιλέγει να προσθέσει κάτι στο καλάθι αγορών τότε έχουμε εμφάνιση του καλαθιού αγορών μέσω της φόρτωσης της σελίδας Cart.aspx, όπου εμφανίζονται τα περιεχόμενα του καλαθιού του συγκεκριμένου χρήστη. Το καλάθι αγορών δεν είναι τίποτε άλλο παρά μια οπτικοποίηση των επιλογών αγοράς του χρήστη. Απλώς αποθηκεύουμε τις επιλογές αγοράς του χρήστη και κάθε φορά που εμφανίζουμε το καλάθι αγορών πηγαίνουμε στη βάση για τον κάθε κωδικό

προϊόντος και αντλούμε δεδομένα όπως η τιμή του προϊόντος κλπ. Ο χρήστης έχει την δυνατότητα να διαγράψει από το καλάθι αγορών οποιαδήποτε στιγμή θέλει κάποια από τις επιλογές του. Απλώς για τον προγραμματιστή αυτό θα σήμαινε ότι θα πρέπει να πάμε στη βάση και να διαγράψουμε αυτό το στοιχείο συναλλαγής. Για αυτό το λόγο και όχι μόνο βέβαια (επίσης και για λόγους ασφαλείας των συναλλαγών) στην αρχή αρχικοποιούμε ένα session αντικείμενο το οποίο παρακολουθεί την συνεδρία συναλλαγών του χρήστη. Από εκεί λοιπόν πηγαίνουμε και διαγράφουμε δεδομένα ή εισάγουμε δεδομένα ανάλογα με την συναλλαγή που γίνεται. Όταν ο χρήστης αποφασίσει να προχωρήσει προς την διαδικασία πληρωμής, αντλούμε όλα τα δεδομένα από το session και προχωρούμε προς την διαδικασία επιβεβαίωσης της συναλλαγής και στην διαδικασία που πρέπει να ρωτήσουμε τον χρήστη για τον τρόπο πληρωμής και τα στοιχεία διανομής της παραγγελίας. Στο παρακάτω σχήμα δείχνουμε την περίπτωση της εμφάνισης του καλαθιού αγοράς όπου υπάρχει στο συγκεκριμένο παράδειγμα ένα προϊόν από προηγούμενη επιλογή του χρήστη προσθήκης του προϊόντος.

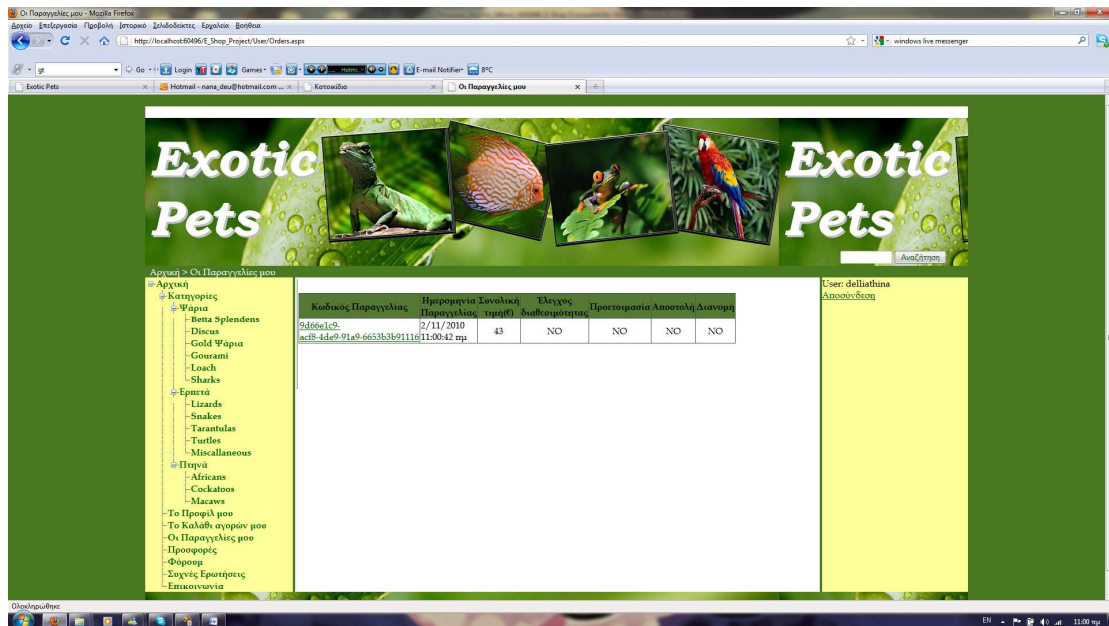


Πατώντας το κουμπί «έξοδος πληρωμής» ο χρήστης οδηγείται στη σελίδα Checkout.aspx. Υπάρχουν όπως είπαμε και παραπάνω, δύο κατηγορίες στοιχείων που πρέπει να πληκτρολογηθούν ή να επιλεγθούν στην παρούσα φάση. Η πρώτη κατηγορία δεδομένων αφορά τα προσωπικά στοιχεία του χρήστη όσον αφορά τον τόπο διανομής της παραγγελίας στον χώρο του. Το δεύτερο κομμάτι πληροφορίας που απαιτείται αφορά τον τρόπο πληρωμής. Εδώ ο χρήστης επιλέγει στο

συγκεκριμένο κατάστημα είτε μετρητά κατά την άφιξη της παραγγελίας είτε πληρωμή με πιστωτική κάρτα. Αξίζει να σημειώσουμε ότι στη δεύτερη περίπτωση ζητούμε από τον χρήστη να βάλει τον αριθμό πιστωτικής κάρτας αλλά επίσης κάνουμε και ένα βήμα παραπάνω: Χρησιμοποιώντας τον αλγόριθμο του Luhn, ελέγχουμε αν ο αριθμός της πιστωτικής κάρτας είναι όντως ο σωστός και αντιστοιχεί σε κάποιον λογαριασμό πραγματικά. Περαιτέρω έλεγχος δεν απαιτείται δεδομένου του γεγονότος ότι για τα ήδη υφιστάμενα ηλεκτρονικά καταστήματα αναλαμβάνουν εταιρίες που ονομάζονται gateways οι οποίες έχουν ως αντικείμενο να δέχονται το ποσό που αντιστοιχεί σε μια παραγγελία, τον αριθμό πιστωτικής κάρτας που αφορά την παραγγελία αυτή και αναλαμβάνουν την ευθύνη να εισπράξουν τα χρήματα από τους πελάτες και να καταθέσουν τα χρήματα αυτά στον λογαριασμό του ιδιοκτήτη του ηλεκτρονικού καταστήματος. Το επόμενο διάγραμμα αφορά λοιπόν την προηγούμενη συζήτησή μας για τις δυνατότητες τρόπων πληρωμής καθώς επίσης φαίνονται και τα πεδία που πρέπει να γεμίζει ο χρήστης όσον αφορά τα στοιχεία διανομής της παραγγελίας αυτής.

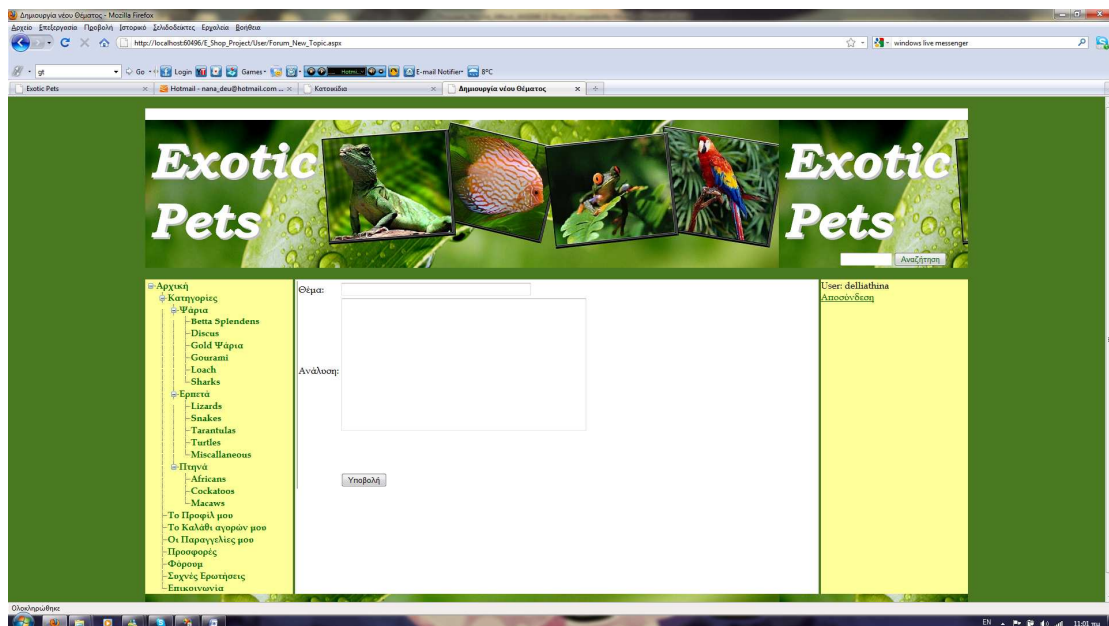
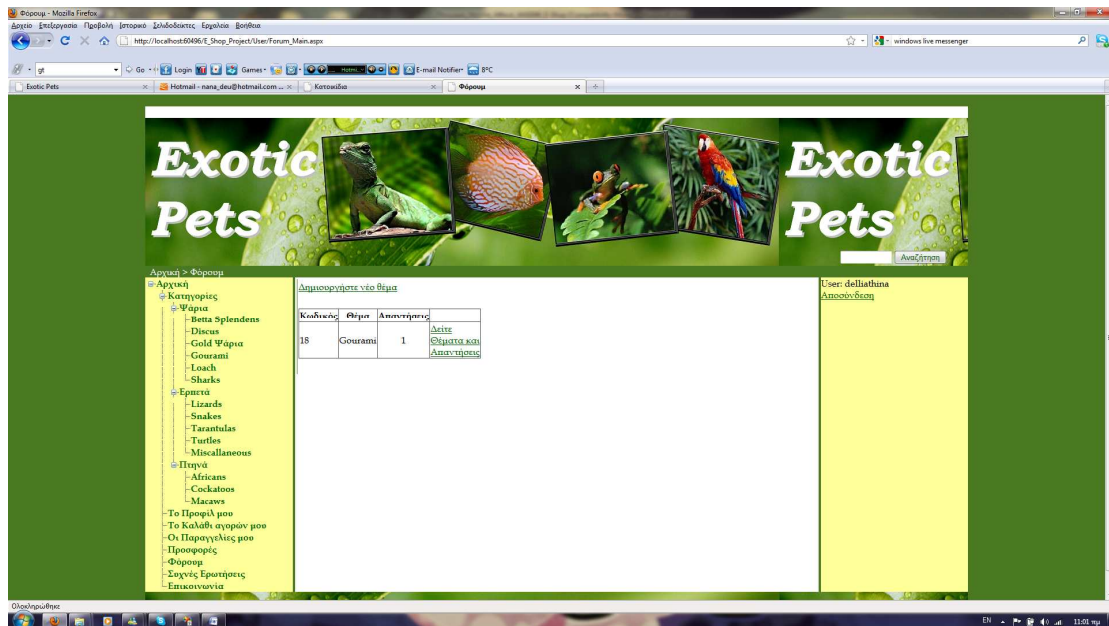


Με το link η παραγγελία μου που περιέχει η σελίδα order_complete πηγαίνουμε στην σελίδα orders.aspx με τις παραγγελίες του κάθε χρήστη και τους κωδικούς που έχουν πάρει από το σύστημα και τα διάφορα στάδια που θα περάσει η παραγγελία έως ότου παραδοθεί σε αυτόν (χρήστη-πελάτη).



Μια άλλη επιπρόσθετη λειτουργία του ηλεκτρονικού καταστήματος είναι η δυνατότητα που δίνεται στους εγγεγραμμένους χρήστες να βλέπουν κατά την είσοδο τους στο κατάστημα μια σελίδα προσφορών. Η σελίδα αυτή παραμετροποιείται και αλλάζει από τον διαχειριστή του καταστήματος για τον οποίο θα μιλήσουμε σε επόμενη ενότητα αυτού του κεφαλαίου.

Επίσης μια ακόμη λειτουργικότητα του καταστήματος έχει να κάνει με το κομμάτι ανάρτησης μηνυμάτων των μελών χρηστών με τη χρήση ενός απλού forum. Στο forum κάποιος χρήστης έχει τη δυνατότητα να δημιουργήσει μια νέα θεματική ενότητα ή να γράψει και να αναρτήσει ένα μήνυμα σε μια ήδη υπάρχουσα θεματική ενότητα η οποία έχει ήδη δημιουργηθεί είτε από αυτόν είτε από κάποιο άλλο χρήστη. Όλες οι θεματικές ενότητες καθώς και όλες οι αναρτήσεις μηνυμάτων από όλους τους χρήστες μπορούν να είναι ορατές από όλους ανεξάρτητα τους χρήστες, αρκεί όμως να είναι εγγεγραμμένοι. Τα παρακάτω δύο σχήματα δείχνουν την συγκεκριμένη σελίδα εμφάνισης των θεματικών ενότητων του forum και τη σελίδα ανάρτησης ενός νέου μηνύματος σε μια συγκεκριμένη θεματική ενότητα.

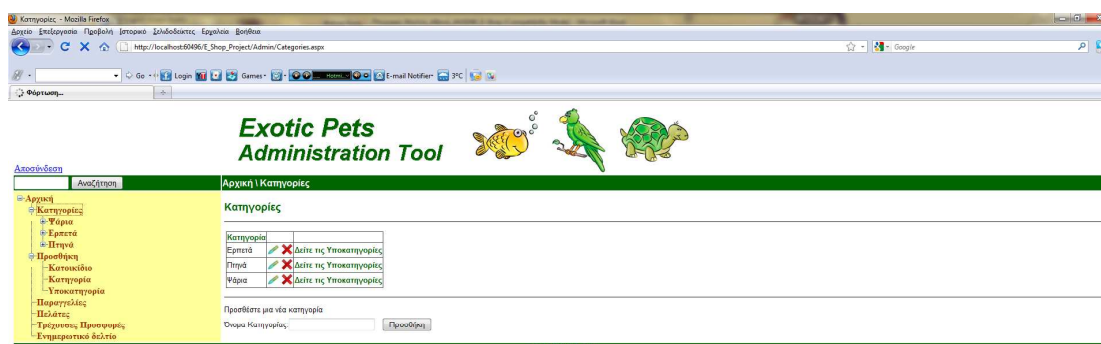


ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ ΓΙΑ ΤΗΝ ΟΜΑΔΑ ΤΩΝ ΔΙΑΧΕΙΡΙΣΤΩΝ

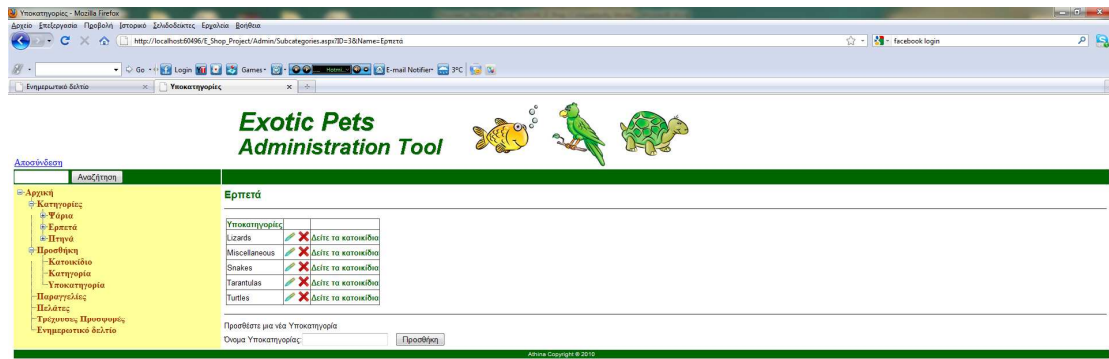
Η δυνατότητα σύνδεσης στον ιστότοπο με την ιδιότητα του διαχειριστή έχει δοθεί από τον αρχικό του σχεδιασμό με την χρήση του Web Site Configuration Tool. Με τη χρήση αυτού του εργαλείου κατηγοριοποιήσαμε τους χρήστες σε τρεις διαφορετικές κατηγορίες: τους Διαχειριστές, τους Εγγεγραμμένους Χρήστες και στους απλούς επισκέπτες. Μάλιστα με την χρήση αυτού του εργαλείου μας δίνεται η δυνατότητα να δημιουργήσουμε εκεί και λογαριασμούς για διαχειριστές και λογαριασμούς χρηστών χωρίς ακόμη να έχουμε σχεδιάσει τις σελίδες εγγραφών χρηστών. Για τους χρήστες

που κάνουν εγγραφή δημιουργήσαμε αργότερα ειδικές σελίδες. Για τους διαχειριστές όμως αφήσαμε την αρχική λειτουργία. Δηλαδή κάποιος ήδη διαχειριστής μπορεί να μπει στο Web Site Configuration Tool και να δημιουργήσει ένα ακόμη Administrator για το site.

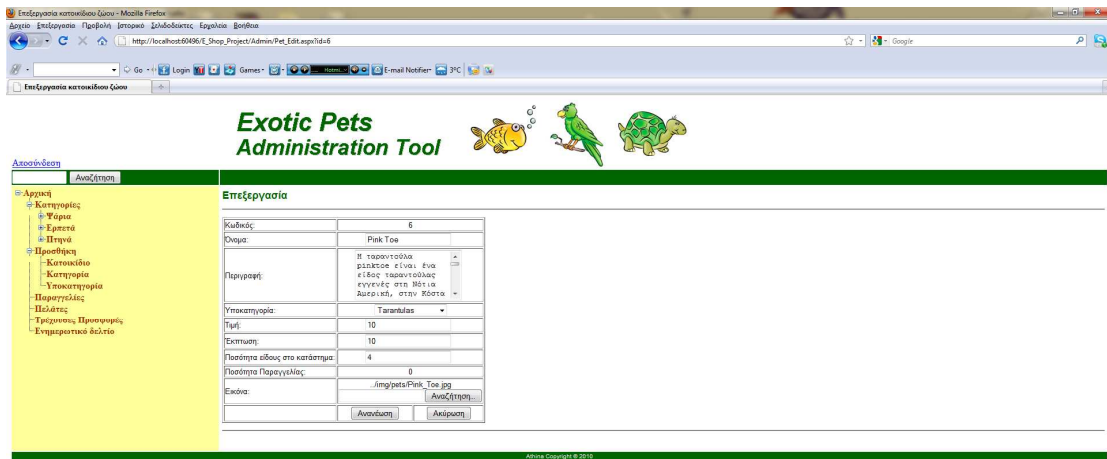
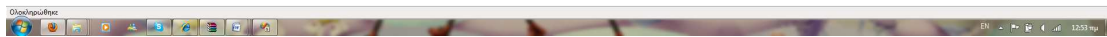
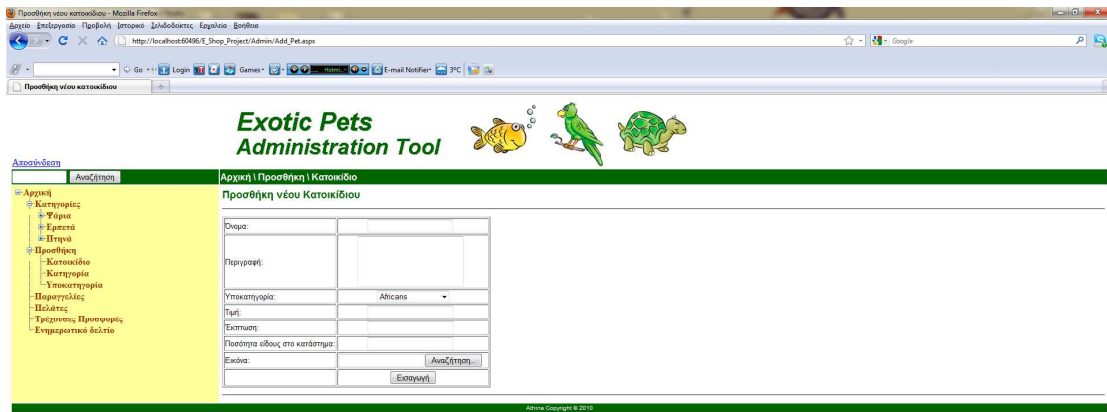
Όταν κάποιος διαχειριστής κάνει σύνδεση το σύστημα πάει σε μια διαφορετική αρχική σελίδα που ονομάζεται Admin/Categories.aspx και η οποία φαίνεται στο παρακάτω σχήμα. Σε αυτή την σελίδα ο διαχειριστής διαχειρίζεται (εισάγει, διαγράφει, επεξεργάζεται) τις κατηγορίες στην βάση δεδομένων του ηλεκτρονικού καταστήματος.



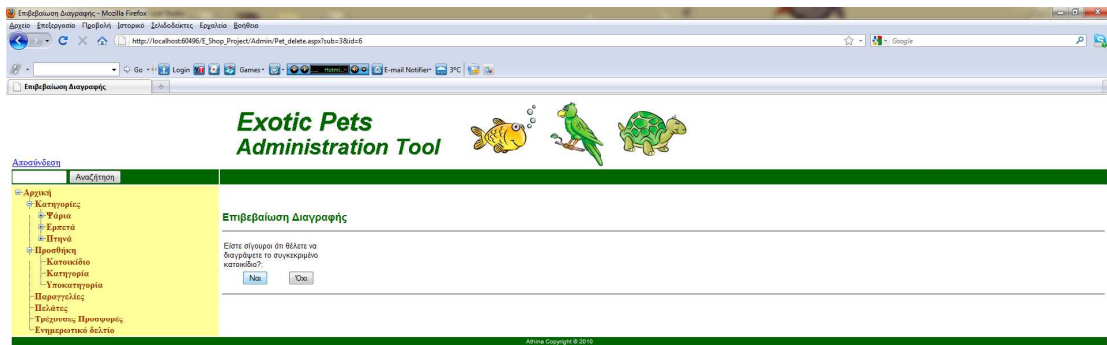
Για κάθε κατηγορία κατοικίδιων θυμίζουμε ότι έχουμε δημιουργήσει διάφορες υποκατηγορίες. Πατώντας στο link Δείτε τις υποκατηγορίες φορτώνει η σελίδα Admin/Subcategories.aspx στην οποία ο διαχειριστής μπορεί να εισάγει μια νέα υποκατηγορία, να διαγράψει μια υφιστάμενη ή ακόμη και να αλλάξει το όνομα μιας ήδη υπάρχουσας υποκατηγορίας προϊόντων. Αυτό απεικονίζεται στο παρακάτω σχήμα.



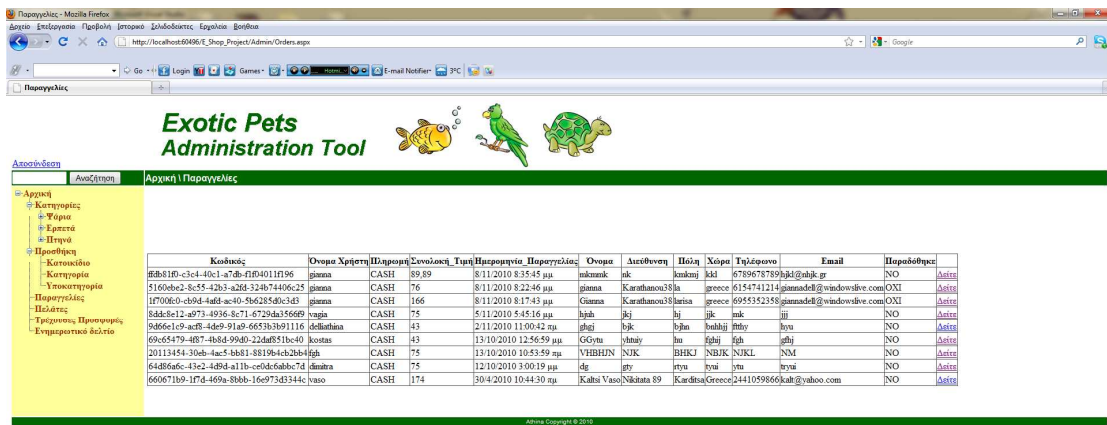
Για την εισαγωγή ενός νέου κατοικίδιου, δηλαδή ενός νέου προϊόντος, ο διαχειριστής μεταβαίνει στη σελίδα `Admin/Add_Pet.aspx`. Αυτή η μετάβαση γίνεται από την σελίδα των υποκατηγοριών που αφορούσαν την επεξεργασία τους. Σε αυτήν λοιπόν την σελίδα μπορεί ο διαχειριστής να επεξεργαστεί και να εισάγει όλες τις πληροφορίες που αφορούν ένα νέο κατοικίδιο συμπεριλαμβανομένων της εικόνας, της αναλυτικής περιγραφής και της αρχικής τιμής διάθεσης του νέου προϊόντος. Για τα ήδη εισαγμένα στη βάση δεδομένων κατοικίδια χρειαζόμαστε επίσης μια σελίδα επεξεργασίας για τους διαχειριστές ειδικά στις περιπτώσεις όπου χρειάζεται να γίνει αλλαγή της τιμής διάθεσής τους αλλά επίσης να αλλάξουν κάποιες φωτογραφίες που αφορούν το κατοικίδιο ή η περιγραφή του. Για αυτό το λόγο δημιουργήσαμε τη σελίδα `Admin/Pet_Edit.aspx` η οποία θα ανοίξει με παράμετρο το `pet_id` δηλαδή είναι δυναμική. Στη μέθοδο `Page_Load()` της σελίδας αυτής γνωρίζοντας το `pet_id` θα ανακτήσουμε αρχικά από τη βάση δεδομένων (πίνακα προϊόντων) τα στοιχεία που αφορούν το συγκεκριμένο κατοικίδιο και θα τα εμφανίσουμε σε κατάλληλα text boxes για την επεξεργασία τους. Όταν τελειώσει η επεξεργασία των στοιχείων ο διαχειριστής ενημερώνει το σύστημα με πάτημα ενός κουμπιού του οποίου ο κώδικας κάνει update στη βάση δεδομένων τα νέα στοιχεία για το συγκεκριμένο κατοικίδιο. Τα παρακάτω σχήματα δείχνουν αυτή τη διαδικασία.

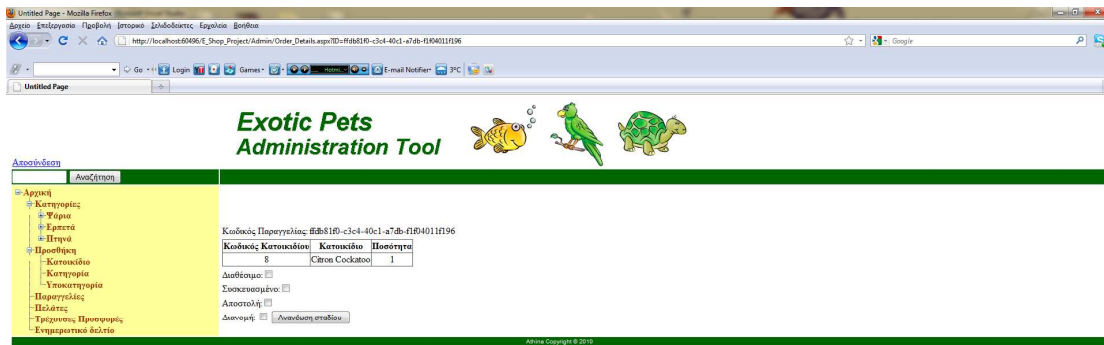


Μια άλλη επιλογή του διαχειριστή είναι η διαγραφή ενός προϊόντος από τη βάση δεδομένων για το λόγο του ότι δεν υποστηρίζεται πλέον η διάθεση του από το ηλεκτρονικό κατάστημα. Όπως φαίνεται στο παρακάτω σχήμα μπορεί ο διαχειριστής να διαγράψει από τη βάση δεδομένων το προϊόν. Πρέπει να σημειώσουμε ότι για τους χρήστες αυτή η αλλαγή σημαίνει ότι δεν θα φαίνεται το συγκεκριμένο προϊόν στο μενού επιλογών τους. Αυτό γίνεται όμως αυτόματα χωρίς περαιτέρω σχεδιασμό δεδομένου του γεγονότος ότι έχει σχεδιαστεί η ιστοσελίδα με τη χρήση του sitemap αρχείου και επομένως μια τέτοια αλλαγή σημαίνει αυτόματα και τη διαγραφή του προϊόντος από το αρχείο αυτό.

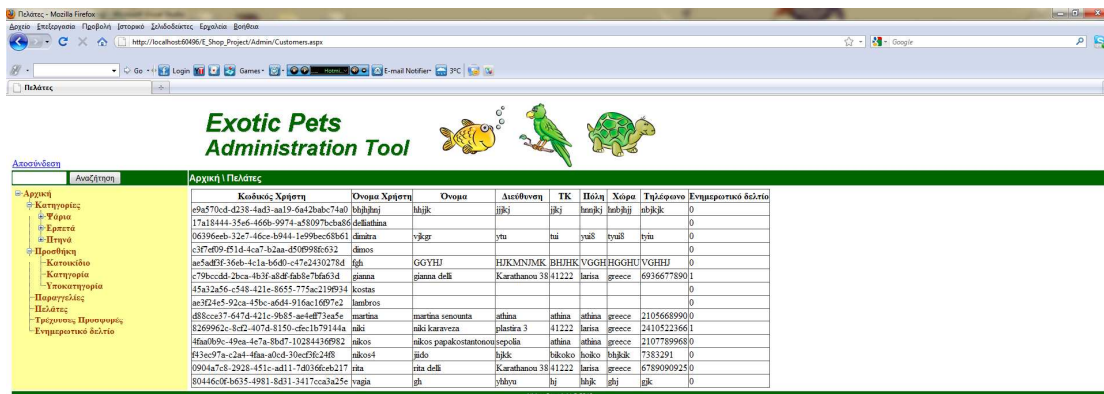


Για να έχει μια γενική εικόνα της κίνησης του ηλεκτρονικού καταστήματος ο διαχειριστής του δίνεται η δυνατότητα να έχει τον συνολικό πίνακα των παραγγελιών μέσω της σελίδας Admin/orders.aspx όπως φαίνεται στο παρακάτω σχήμα και εάν θέλει να μπορέσει να επεξεργαστεί κάποια από αυτές όπως φαίνεται στο μεθεπόμενο σχήμα της σελίδας Admin/Order_Details.aspx

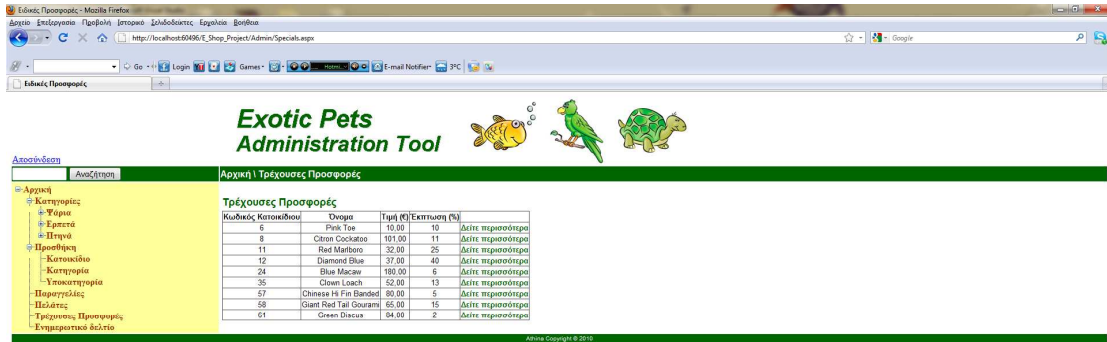




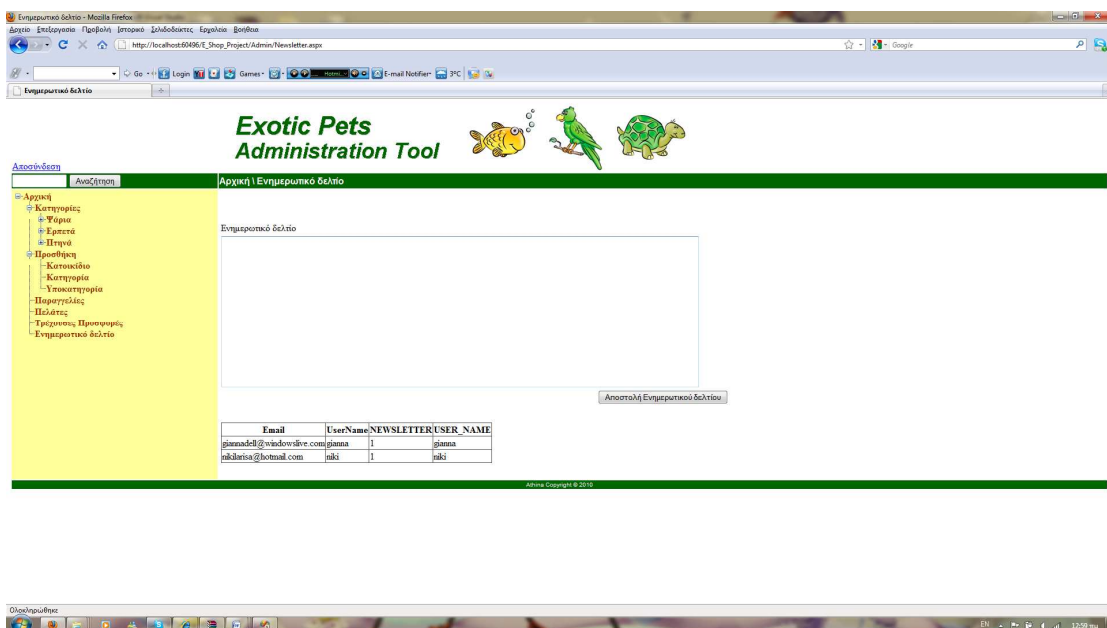
Μια επιπρόσθετη επιλογή του διαχειριστή είναι η δυνατότητα επίσης επισκόπησης όλων των εγγεγραμμένων μελών του καταστήματος η οποία γίνεται μέσω της σελίδας Admin/Customers.aspx



Καθώς επίσης και να διαχειριστεί (να επεξεργαστεί , να εισάγει , να διαγράψει) κάποιες προσφορές από την σελίδα των προσφορών όπως φαίνεται στο παρακάτω σχήμα που είναι μια απεικόνιση της σελίδας Admin/specials.aspx



Τέλος μια αυτοματοποίηση της ενημέρωσης του διαχειριστή με τα υπόλοιπα μέλη χρήστες του ηλεκτρονικού καταστήματος γίνεται μέσω της δυνατότητας του διαχειριστή να στέλνει ενημερωτικά δελτία μέσω ηλεκτρονικού ταχυδρομείου. Θα πρέπει να επισημάνουμε ότι κατά την διαδικασία εγγραφής ενός νέου μέλους ρωτάμε τον πελάτη αν θέλει να δέχεται τέτοια ενημερωτικά δελτία. Η διαμόρφωση του ενημερωτικού δελτίου γίνεται κεντρικά από τον διαχειριστή με τη χρήση της σελίδας Admin/Newsletter.aspx που φαίνεται στο παρακάτω σχήμα. Σε επίπεδο προγραμματισμού χρησιμοποιείται πάλι το mail αντικείμενο της C# μέσα σε κάποιο βρόχο και στέλνεται το ίδιο μήνυμα (δηλαδή το ίδιο σώμα μηνύματος) αλλά με την μόνη αλλαγή της ιδιότητας mail.To



ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΑΤΑΣΤΗΜΑΤΟΣ ΟΣΩΝ ΑΦΟΡΑ ΤΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

Οι παρακάτω πίνακες χρησιμοποιήθηκαν για την υλοποίηση και αποθήκευση πληροφοριών στο ηλεκτρονικό μας κατάστημα .

Πίνακας ΚΑΤΟΙΚΙΔΙΑ (Table Pets) (χρησιμοποιείται για την αποθήκευση απαραίτητων δεδομένων για κάθε κατοικίδιο το οποίο είναι διαθέσιμο στο κατάστημα μας)

	Column Name	Data Type	Allow Nulls
🔑	Pet_ID	int	<input type="checkbox"/>
	Name	varchar(80)	<input type="checkbox"/>
	Description	varchar(1000)	<input checked="" type="checkbox"/>
	SubCat_ID	int	<input type="checkbox"/>
	Price	decimal(10, 2)	<input checked="" type="checkbox"/>
	Discount	int	<input checked="" type="checkbox"/>
	Units_Stock	int	<input checked="" type="checkbox"/>
	Units_Ordered	int	<input checked="" type="checkbox"/>
	Image	varchar(150)	<input checked="" type="checkbox"/>

Πίνακας ΚΑΤΗΓΟΡΙΕΣ (Table Categories) αποθηκεύουμε τα ονόματα της κάθε κατηγορίας προϊόντων που περιέχονται στο ηλεκτρονικό μας κατάστημα

	Column Name	Data Type	Allow Nulls
🔑	Cat_ID	int	<input type="checkbox"/>
	Cat_Name	varchar(100)	<input type="checkbox"/>

Πίνακας ΥΠΟΚΑΤΗΓΟΡΙΕΣ (Table SubCategories) : περιέχει το όνομα της κάθε υποκατηγορίας των κατοικίδιων ενώ το πεδίο Sub_ID field υποδεικνύει το μοναδικό προσδιοριστικό για κάθε υποκατηγορία ενώ το Scat_ID δηλώνει το όνομα της κατηγορίας στην οποία ανήκει η κάθε υποκατηγορία

	Column Name	Data Type	Allow Nulls
🔑	Sub_ID	int	<input type="checkbox"/>
	SCat_ID	int	<input type="checkbox"/>
	SubName	varchar(50)	<input type="checkbox"/>

Ο πίνακας ΧΡΗΣΤΩΝ και ο πίνακας ΙΔΙΟΤΗΤΑΣ ΜΕΛΩΝ (Tables aspnet_Users και aspnet_Membeship)

Αυτοί οι δύο πίνακες περιέχουν τα δεδομένα που αφορούν τα προσωπικά δεδομένα του κάθε χρήστη ο οποίος εγγράφεται στο ηλεκτρονικό κατάστημα.

	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"/>

	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"/>
	LastPasswordChangedDa...	datetime	<input type="checkbox"/>
	LastLockoutDate	datetime	<input type="checkbox"/>
	FailedPasswordAttemptC...	int	<input type="checkbox"/>
	FailedPasswordAttempt...	datetime	<input type="checkbox"/>
	FailedPasswordAnswerAt...	int	<input type="checkbox"/>
	FailedPasswordAnswerAt...	datetime	<input type="checkbox"/>
	Comment	ntext	<input checked="" type="checkbox"/>


Ο πίνακας ΛΕΠΤΟΜΕΡΕΙΕΣ_ΧΡΗΣΤΩΝ (Table User_Details)

Αυτός ο πίνακας αποθηκεύει τα δεδομένα μετά την εγγραφή του χρήστη στο σύστημα.

	Column Name	Data Type	Allow Nulls
🔑	USER_NAME	nvarchar(256)	<input type="checkbox"/>
	NAME	varchar(100)	<input checked="" type="checkbox"/>
	ADDRESS	varchar(100)	<input checked="" type="checkbox"/>
	ZIP	varchar(15)	<input checked="" type="checkbox"/>
	CITY	varchar(50)	<input checked="" type="checkbox"/>
	COUNTRY	varchar(50)	<input checked="" type="checkbox"/>
	TELEPHONE	varchar(20)	<input checked="" type="checkbox"/>
	NEWSLETTER	varchar(1)	<input checked="" type="checkbox"/>


Ο πίνακας ΠΑΡΑΓΓΕΛΙΕΣ (Table Orders)

Στον πίνακα των παραγγελιών αποθηκεύουμε δεδομένα που έχουν σχέση με τις παραγγελίες χρηστών που έχουν όμως ολοκληρωθεί.

	Column Name	Data Type	Allow Nulls
	ID	varchar(100)	<input type="checkbox"/>
	USERNAME	varchar(80)	<input type="checkbox"/>
	PAYMENT	varchar(20)	<input type="checkbox"/>
	TOTAL_PRICE	varchar(15)	<input type="checkbox"/>
	ORDER_DATE	datetime	<input type="checkbox"/>
	NAME	varchar(80)	<input type="checkbox"/>
	ADDRESS	varchar(50)	<input type="checkbox"/>
	CITY	varchar(30)	<input type="checkbox"/>
	COUNTRY	varchar(25)	<input type="checkbox"/>
	TELEPHONE	varchar(15)	<input type="checkbox"/>
	EMAIL	varchar(50)	<input type="checkbox"/>
	AVAILABILITY	varchar(3)	<input type="checkbox"/>
	PACKING	varchar(3)	<input type="checkbox"/>
	SENDING	varchar(3)	<input type="checkbox"/>
	DELIVERED	varchar(3)	<input type="checkbox"/>

Ο πίνακας ΛΕΠΤΟΜΕΡΕΙΕΣ_ΠΑΡΑΓΓΕΛΙΑΣ (Table Order_details)

Λεπτομέρειες όπως το id του κάθε κατοικιδίου (δηλαδή του κάθε προϊόντος) καθώς και η ποσότητα παραγγελίας για το κάθε προϊόν (κομμάτια προϊόντος) αποθηκεύονται σε αυτόν τον πίνακα .

	Column Name	Data Type	Allow Nulls
	ID	int	<input type="checkbox"/>
	ORDER_ID	varchar(100)	<input type="checkbox"/>
	PROD_ID	int	<input type="checkbox"/>
	QUANTITY	int	<input type="checkbox"/>

ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ ΤΟΥ WEBSITE

PET CLASS (PET.CS)

```
using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

/// <summary>
/// Περιγραφή του κατοικίδιου
/// </summary>

public class Pet
{
    //ιδιωτικά δεδομένα
    private int Pet_ID;
    private String Name;
    private String Description;
    private int SubCat_ID;
    private double Price;
    private int Discount;
    private int Units_Stock;
    private int Units_Ordered;
    private String Image;
    // default κατασκευαστής
    public Pet()
    {
        Pet_ID = 0;
        Name = "";
        Description = "";
        SubCat_ID = 0;
        Price = 0;
        Discount = 0;
        Units_Stock = 0;
        Units_Ordered = 0;
        Image = "";
    }
    //getters και setters
    public Pet(int _Pet_ID, String _Name, String _Description, int
_SubCat_ID, double _Price, int _Discount, int _Units_Stock, int
_Units_Ordered, String _Image)
    {
        Pet_ID = _Pet_ID;
        Name = _Name;
        Description = _Description;
        SubCat_ID = _SubCat_ID;
        Price = _Price;
        Discount = _Discount;
```

```

        Units_Stock = _Units_Stock;
        Units_Ordered = _Units_Ordered;
        Image = _Image;
    }

    public int petID
    {
        get
        {
            return Pet_ID;
        }
        set
        {
            Pet_ID = value;
        }
    }

    public String name
    {
        get
        {
            return Name;
        }
        set
        {
            Name = value;
        }
    }

    public String description
    {
        get
        {
            return Description;
        }
        set
        {
            Description = value;
        }
    }

    public int subCat_ID
    {
        get
        {
            return SubCat_ID;
        }
        set
        {
            SubCat_ID = value;
        }
    }

    public double price
    {
        get
        {
            return Price;
        }
        set
        {

```

```

        Price = value;
    }
}

public int discount
{
    get
    {
        return Discount;
    }
    set
    {
        Discount = value;
    }
}

public int units_Stock
{
    get
    {
        return Units_Stock;
    }
    set
    {
        Units_Stock = value;
    }
}

public int units_Ordered
{
    get
    {
        return Units_Ordered;
    }
    set
    {
        Units_Ordered = value;
    }
}

public String image
{
    get
    {
        return Image;
    }
    set
    {
        Image = value;
    }
}
}

```

CATEGORY CLASS (CATEGORY.CS)

```

using System;
using System.Data;
using System.Configuration;
using System.Linq;

```

```

using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

/// <summary>
/// Περιγραφή του classCategories
/// </summary>
public class Category
{
    // ιδιωτικές ιδιότητες για τα αντικείμενα αυτής της κλάσης
    private int category_ID;
    private String category_Name;

    public Category() //default κατασκευαστής
    {
        category_ID = 0;
        category_Name = "";
    }
    // υπερφορτωμένος κατασκευαστής
    public Category(int cat_ID, String cat_Name)
    {
        category_ID = cat_ID;
        category_Name = cat_Name;
    }
    // getters και setters για τις ιδιότητες
    public int CatID// για το ID της νέας κατηγορίας
    {
        get
        {
            return category_ID;
        }
        set
        {
            category_ID = value;
        }
    }

    public String CatName// για το όνομα της κατηγορίας
    {
        get
        {
            return category_Name;
        }
        set
        {
            category_Name = value;
        }
    }
}

```


USER CLASS (USER_CLASS.CS)

```
using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

/// <summary>
//Περιγραφή του χρήστη-μέλους
/// <summary>

public class User_class
{

    //ιδιωτικά δεδομένα
    private String Username;
    private String Name;
    private String Address;
    private String Zip;
    private String City;
    private String Country;
    private String Telephone;
    private String Newsletter;

    //default κατασκευαστής
    public User_class()
    {
        Username="";
        Name = "";
        Address="";
        Zip = "";
        City="";
        Country="";
        Telephone = "";
        Newsletter="0";
    }
    //υπερφορτωμένος κατασκευαστής
    public User_class(String _Username, String _Name, String
_Address, String _zip, String _city, String _country, String
_telephone, String _newsletter)
    {
        Username = _Username;
        Name = _Name;
        Address = _Address;
        Zip = _zip;
        City = _city;
        Country = _country;
        Telephone = _telephone;
        Newsletter = _newsletter;
    }
}
```

```

//getters και setters
public String username
{
    get
    {
        return Username;
    }
    set
    {
        Username = value;
    }
}
public String name
{
    get
    {
        return Name;
    }
    set
    {
        Name = value;
    }
}
public String address
{
    get
    {
        return Address;
    }
    set
    {
        Address = value;
    }
}
public String zip
{
    get
    {
        return Zip;
    }
    set
    {
        Zip = value;
    }
}
public String city
{
    get
    {
        return City;
    }
    set
    {
        City = value;
    }
}
public String country
{
    get
    {
        return Country;
    }
}

```

```

        }
        set
        {
            Country = value;
        }
    }
    public String telephone
    {
        get
        {
            return Telephone;
        }
        set
        {
            Telephone = value;
        }
    }
    public String newsletter
    {
        get
        {
            return Newsletter;
        }
        set
        {
            Newsletter = value;
        }
    }
}
}

```

CART ITEM CLASS (CART_ITEM.CS)

```

using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

///Εδώ είναι όλος ο κώδικας της κλάσης
/// cart_item που αφορά κάθε προϊόν που μπαίνει στο
/// καλάθι αγορών

public class Cart_Item
{
    // όλες οι ιδιωτικές ιδιότητες για τα αντικείμενα αυτής της κλάσης
    private int intPetID;
    private string strPetName;
    private double decUnitPrice;
    private int intQuantity;
    private int intDiscount;
    // getters και setters για κάθε μια από τις ιδιότητες

```

```

// πρώτα για το PetID το ID για το κατοικίδιο
public int PetID
{
    get
    {
        return intPetID;
    }
    set
    {
        intPetID = value;
    }
}
// μετά για το PetName, δηλαδή το όνομα του κατοικίδιου
public string PetName
{
    get
    {
        return strPetName;
    }
    set
    {
        strPetName = value;
    }
}
// μετά για το UnitPrice, δηλαδή η τιμή πώλησης
public double UnitPrice
{
    get
    {
        return decUnitPrice;
    }
    set
    {
        decUnitPrice=value;
    }
}
// μετά για το Quantity, δηλαδή η ποσότητα των κατοικίδιων
// για το συγκεκριμένο όμως κατοικίδιο
// που μπορεί να αγοράσει ο χρήστης σε μια παραγγελία
public int Quantity
{
    get
    {
        return intQuantity;
    }
    set
    {
        intQuantity=value;
    }
}
// μετά για την έκπτωση που μπορεί να δώσει ο διαχειριστής
public int Discount
{
    get
    {
        return intDiscount;
    }
    set
    {
        intDiscount = value;
    }
}

```

```
}  
}
```

CREDIT CARD VALIDATION CLASS (CREDITCARDVALIDATION.CS)

```
using System;  
using System.Data;  
using System.Configuration;  
using System.Linq;  
using System.Web;  
using System.Web.Security;  
using System.Web.UI;  
using System.Web.UI.HtmlControls;  
using System.Web.UI.WebControls;  
using System.Web.UI.WebControls.WebParts;  
using System.Xml.Linq;  
using System.Text;  
  
/// <summary>  
/// Ο αλγόριθμος του Luhn για την εγκυρότητα του αριθμού πιστωτικής  
/// βρέθηκε από το Internet και πιο συγκεκριμένα  
/// από το site www.codeproject.com  
/// Απλώς είναι copy paste  
/// </summary>  
public class CreditCardValidation  
{  
    public bool CardValidation(string cardNumber)  
    {  
        const string allowed = "0123456789";  
        int i;  
  
        StringBuilder cleanNumber = new StringBuilder();  
  
        for (i = 0; i < cardNumber.Length; i++)  
        {  
            if (allowed.IndexOf(cardNumber.Substring(i, 1)) >= 0)  
                cleanNumber.Append(cardNumber.Substring(i, 1));  
        }  
  
        if (cleanNumber.Length < 13 || cleanNumber.Length > 16)  
            return false;  
  
        for (i = cleanNumber.Length + 1; i <= 16; i++)  
            cleanNumber.Insert(0, "0");  
  
        int multiplier, digit, sum, total = 0;  
        string number = cleanNumber.ToString();  
  
        for (i = 1; i <= 16; i++)  
        {  
            multiplier = 1 + (i % 2);  
            digit = int.Parse(number.Substring(i - 1, 1));  
            sum = digit * multiplier;  
            if (sum > 9)  
                sum -= 9;  
            total += sum;  
        }  
    }  
}
```

```

        return (total % 10 == 0);
    }
}

```

ORDER STATUS CLASS (ORDER_STATUS.CS)

```

using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

/// <summary>
/// Περιγραφή της τάξης Order_Status
/// Η κλάση χρησιμοποιείται από τον διαχειριστή
/// για την διαχείριση των παραγγελιών και
/// για τον έλεγχο της φάσης στην οποία βρίσκεται κάθε
/// παραγγελία
/// Περιέχει μόνο ιδιότητες αλλά επίσης κατασκευαστές καθώς και
/// getters και setters
/// </summary>
public class Order_Status
{
    // ιδιωτικές ιδιότητες για τα αντικείμενα της κλάσης
    private String Availability;
    private String Packing;
    private String Sending;
    private String Delivered;
    //κατασκευαστής default
    public Order_Status()
    {
        Availability="OXI";
        Packing = "OXI";
        Sending="OXI";
        Delivered = "OXI";
    }
    //υπερφορτωμένος κατασκευαστής
    public Order_Status(String _Availability, String _Packing, String
    _Sending, String _Delivered)
    {
        Availability = _Availability;
        Packing = _Packing;
        Sending = _Sending;
        Delivered = _Delivered;
    }

    //getters και setters για όλες τις ιδιότητες
    public String availability
    {
        get
        {
            return Availability;

```

```

        }
        set
        {
            Availability = value;
        }
    }
    public String packing
    {
        get
        {
            return Packing;
        }
        set
        {
            Packing = value;
        }
    }
    public String sending
    {
        get
        {
            return Sending;
        }
        set
        {
            Sending = value;
        }
    }
    public String delivered
    {
        get
        {
            return Delivered;
        }
        set
        {
            Delivered = value;
        }
    }
}
}
DATA ACCESS CLASS (DATA_ACCESS.CS)

```

```

using System;
using System.Data;
using System.Configuration;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;
using System.Collections;

```

```

/// <summary>
/// Περιγραφή του Data_Access
/// </summary>

```

```

public class Data_Access
{
    public Data_Access()
    {
        //
        // TODO: Add constructor logic here
        //
    }

    //χρησιμοποιείται ώστε να επιστρέφει τις κατηγορίες των
    κατοικιδίων στον διαχειριστή.
    public ArrayList Show_Categories()
    {
        ArrayList list = new ArrayList();
        Category cat = new Category();
        // δηλωση κατάλληλων αντικειμένων για διαχείριση της
        βάσης
        SqlConnection SQLCon = new SqlConnection();
        // δημιουργία του QUERY
        string strSQL = "SELECT * FROM Categories";
        SqlDataReader rdr = null;
        //ανοίγουμε τη βάση για να διαβάσουμε δεδομένα

        try
        {
            // ανοίγουμε τον SQLEXPRESS server
            // πρώτα κάνουμε το connection με το connection
            string
                SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
            // ανοίγουμε τη βάση
            SQLCon.Open();
            // δημιουργούμε ένα command αντικείμενο
            // αυτό θα τρέξει το query
            SqlCommand cmd = new SqlCommand();
            // στο αντικείμενο αυτό βάζουμε το connection
            cmd.Connection = SQLCon;
            cmd.CommandText = strSQL;
            // τρέχουμε το query
            rdr = cmd.ExecuteReader();
            // διαβάζουμε αποτελέσματα και τα βάζουμε στο
            ArrayList
            // Το ArrayList θα το επιστρέψουμε πίσω από αυτή την
            μέθοδο

            while (rdr.Read())
            {
                cat = new Category();
                cat.CatID = int.Parse(rdr[0].ToString());
                cat.CatName = rdr[1].ToString();
                list.Add(cat);
            }
        }
        // τελικά κλείνουμε το διάβασμα
        finally
        {
            if (rdr != null)
            {
                rdr.Close();
            }
        }
    }
}

```



```

        // και κλείνουμε την σύνδεση με τη βάση
        if (SQLCon != null)
        {
            SQLCon.Close();
        }
    }
    // επιστρέφουμε το ArrayList
    return list;
}

//χρησιμοποιείται από τον διαχειριστή για να εισάγει μια νέα
κατηγορία κατοικίδιων.
public String Insert_Category(String Category)
{
    try
    {
        // δήλωση κατάλληλων αντικειμένων για διαχείριση της
        βάσης
        // ανοίγουμε τον SQLEXPRESS server
        // πρώτα κάνουμε το connection με το connection string
        SqlConnection SQLCon = new SqlConnection();
        string strSQL = "INSERT INTO Categories (Cat_Name)
VALUES ('" + Category + "')";
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();
        // δημιουργούμε ένα command αντικείμενο
        // αυτό θα τρέξει το query
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        //τρέχουμε το query
        cmd.ExecuteNonQuery();
        //κλείνουμε τη σύνδεση
        SQLCon.Close();
        return (" Η νέα κατηγορία δημιουργήθηκε");
    }
    catch (Exception e)
    {
        // πιάνουμε την εξαίρεση όταν δημιουργήσει πρόβλημα
        return ("Εμφανίστηκε πρόβλημα " + e.Message);
    }
}

//χρησιμοποιείται από τον διαχειριστή για να δημιουργήσει μια νέα
υποκατηγορία κατοικιδίων κάτω από την κατηγορία
public String Insert_SubCategory(int Cat_ID,String SubCategory)
{
    try
    {
        // η ίδια διαδικασία όπως παραπάνω
        // δημιουργία σύνδεσης
        SqlConnection SQLCon = new SqlConnection();
        // δημιουργία query
        string strSQL = "INSERT INTO SubCategories
(SCat_ID,SubName) VALUES (" + Cat_ID + ",'" + SubCategory + "')";
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
    }
}

```

```

        // άνοιγμα σύνδεσης
        SQLCon.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        //τρέξιμο query
        cmd.ExecuteNonQuery();
        // κλείσιμο σύνδεσης
        SQLCon.Close();
        return ("Η νέα υποκατηγορία δημιουργήθηκε");
    }
    catch (Exception e)
    { // εξαίρεση
        return ("Εμφανίστηκε πρόβλημα " + e.Message);
    }
}

//χρησιμοποιείται από τον διαχειριστή για να διαγράψει μια
κατηγορία και όλες τις υποκατηγορίες από την συγκεκριμένη κατηγορία
public String Delete(int id,int type)
{
    SqlConnection SQLCon = new SqlConnection();
    string strSQL = "";
    // έχουμε ένα switch γιατί έχουμε
    // διαφόρων ειδών delete queries
    switch (type)
    {
        case 1:
            strSQL = "DELETE FROM Categories WHERE Cat_ID =" +
id;

            break;
        case 2:
            strSQL = "DELETE FROM SubCategories WHERE SCat_ID ="
+ id;

            break;

        default:
            return "Παρακαλώ προσπαθήστε ξανά";
            break;
    }

    try
    {
        // ξανά η ίδια διαδικασία όπως πάνω
        // δημιουργία αντικειμένου
        // δημιουργία σύνδεσης
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        //τρέξιμο query
        cmd.ExecuteNonQuery();
        SQLCon.Close();
        return "Η διαγραφή έγινε με επιτυχία";
    }
    catch (Exception e)
    { // εξαίρεση
        return "Η διαγραφή απέτυχε " + e.Message;
    }
}

```

```

    }
}

//Η Pet_Details μέθοδος χρησιμοποιείται για να εμφανίζει στον
διαχειριστή περισσότερες λεπτομέρειες για ένα pet
//έτσι ώστε να έχει την δυνατότητα να τις αναπροσαρμόσει.
public Pet Pet_Details(int id)
{
    Pet mypet = new Pet();

    SqlConnection SQLCon = new SqlConnection();
    string strSQL = "SELECT * FROM Pets where Pet_ID="+id;
    SqlDataReader rdr = null;

    try
    {
        //ιδια διαδικασία όπως και παραπάνω
        // διαδικαστικά για τη σύνδεση και το τρέξιμο του query
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        rdr = cmd.ExecuteReader();
        // εδώ διαβάζω τα αποτελέσματα του SELECT
        while (rdr.Read())
        {
            // για κάθενα από τα κατοικίδια φτιάχνω ένα νέο
            // αντικείμενο τύπου Pet
            mypet = new Pet();

            mypet.petID = int.Parse(rdr[0].ToString());
            mypet.name = rdr[1].ToString();
            mypet.description = rdr[2].ToString();
            mypet.subCat_ID = int.Parse(rdr[3].ToString());
            mypet.price = double.Parse(rdr[4].ToString());
            mypet.discount = int.Parse(rdr[5].ToString());
            mypet.units_Stock = int.Parse(rdr[6].ToString());
            mypet.units_Ordered = int.Parse(rdr[7].ToString());
            mypet.image = rdr[8].ToString();
        }
    }

    finally
    {
        if (rdr != null)
        {
            rdr.Close();
        }
        if (SQLCon != null)
        {
            SQLCon.Close();
        }
    }
    return mypet;
}

//χρησιμοποιείται από τον διαχειριστή για να εισάγει ένα νέο pet

```

```

    public String Insert_Pet(String name,String desc,int
subCat_id,double price,int discount,int stock,String img)
    {
        try
        {
            SqlConnection SQLCon = new SqlConnection();
            string strSQL = "INSERT INTO Pets
(Name,Description,SubCat_ID,Price,Discount,Units_Stock,Units_Ordered,
Image) VALUES ('" + name + "','"+ desc + "','"+ subCat_id + "','"+
price + "','"+ discount + "','"+ stock + "','0','../img/pets/' + img +
"')";

            SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
            SQLCon.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = SQLCon;
            cmd.CommandText = strSQL;
            cmd.ExecuteNonQuery();

            SQLCon.Close();
            return ("Το νέο κατοικίδιο προστέθηκε");
        }
        catch (Exception e)
        {
            return ("Εμφανίστηκε πρόβλημα " + e.Message);
        }
    }

    //χρησιμοποιείται από τον διαχειριστή για να αναπροσαρμόσει τις
λεπτομέρειες
    //ενός pet
    public String Update_Pet(int id, String name, String desc, int
subCat_id, double price, int discount, int stock, String img)
    {
        SqlConnection SQLCon = new SqlConnection();
        string strSQL = "UPDATE Pets SET Name= '" + name + "','
Description='" + desc + "',' SubCat_ID='" + subCat_id + "','Price='" +
price + "','Discount='" + discount + "','Units_Stock='" + stock +
 "','Image='" + img + "' WHERE Pet_ID = " + id;

        try
        {
            SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
            SQLCon.Open();
            SqlCommand cmd = new SqlCommand();
            cmd.Connection = SQLCon;
            cmd.CommandText = strSQL;
            cmd.ExecuteNonQuery();
            SQLCon.Close();
            return "Η ανανέωση πραγματοποιήθηκε με επιτυχία";
        }
        catch (Exception e)
        {
            return "Η ανανέωση απέτυχε" + e.Message;
        }
    }

    //χρησιμοποιείται από τον διαχειριστή για να διαγράψει ένα

```

```

// κατοικίδιο
public String Delete_Pet(int id)
{
    SqlConnection SQLCon = new SqlConnection();
    string strSQL = "DELETE FROM Pets WHERE Pet_ID =" + id;

    try
    {
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        cmd.ExecuteNonQuery();

        return "Το κατοικίδιο διαγράφηκε!";
    }
    finally
    {
        if (SQLCon != null)
        {
            SQLCon.Close();
        }
    }
}

// Αυτή η μέθοδος εισάγει το όνομα χρήστη ενός νέου χρήστη μετά
από εγγραφή στον πίνακα User_Details
// Ο πίνακας User_Details περιέχει τα υπόλοιπα στοιχεία του
χρήστη όπως όνομα ,διεύθυνση,τηλέφωνο
public void Insert_Username(String username)
{
    try
    {
        SqlConnection SQLCon = new SqlConnection();
        string strSQL = "INSERT INTO User_Details (USER_NAME)
VALUES ('"+ username+ "')";
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        cmd.ExecuteNonQuery();

        SQLCon.Close();
    }
    catch (Exception e)
    {
    }
}

// Αυτή η μέθοδος παίρνει τις λεπτομέρειες από τον τρέχον χρήστη
και τις εμφανίζει στην ' Το προφίλ μου' σελίδα.
// στην οποία ο χρήστη μπορεί να επέμβει και να μεταβάλει αυτές
τις λεπτομέρειες

```

```

public User_class Find_User(String username)
{
    User_class myuser = new User_class();

    SqlConnection SQLCon = new SqlConnection();
    string strSQL = "SELECT * FROM User_Details where
USER_NAME='" + username+"'";
    SqlDataReader rdr = null;

    try
    {
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        rdr = cmd.ExecuteReader();

        while (rdr.Read())
        {
            myuser = new User_class();
            myuser.username = rdr[0].ToString();
            myuser.name = rdr[1].ToString();
            myuser.address = rdr[2].ToString();
            myuser.zip = rdr[3].ToString();
            myuser.city = rdr[4].ToString();
            myuser.country = rdr[5].ToString();
            myuser.telephone = rdr[6].ToString();
            myuser.newsletter = rdr[7].ToString();

        }
    }

    finally
    {
        if (rdr != null)
        {
            rdr.Close();
        }
        if (SQLCon != null)
        {
            SQLCon.Close();
        }
    }
    return myuser;
}

```

```

// Αυτή η μέθοδος χρησιμοποιείται για να αναπροσαρμόσει τις
λεπτομέρειες στο προφίλ των χρηστών
public String Update_User_Details(String username, String name,
String address, String zip, String city, String country, String
telephone, String newsletter)
{
    SqlConnection SQLCon = new SqlConnection();
    string strSQL = "UPDATE User_details SET NAME='"+name+"',
ADDRESS= '"+address+"', ZIP='"+zip+"',CITY='"+city+"',
COUNTRY='"+country+"', TELEPHONE='"+telephone+"',
NEWSLETTER='"+newsletter+"' WHERE USER_NAME = '"+username+"'";

```

```

try
{
    SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
    SQLCon.Open();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = SQLCon;
    cmd.CommandText = strSQL;
    cmd.ExecuteNonQuery();
    SQLCon.Close();
    return "Τα προσωπικά σας στοιχεία καταχωρήθηκαν ";
}
catch (Exception e)
{
    return "Τα προσωπικά σας στοιχεία δεν καταχωρήθηκαν " +
e.Message;
}
}

```

//Αυτή η μέθοδος χρησιμοποιείται για να εισάγει ένα νέο θέμα στο φόρουμ στην βάση στον πίνακα FORUM_TOPIC

```

public String Topic_Insert(String Topic, String Detail, String
Username)
{
    try
    {
        SqlConnection SQLCon = new SqlConnection();
        string strSQL = "INSERT INTO FORUM_TOPIC
(TOPIC,DETAIL,USERNAME,DATE) VALUES ('" + Topic + "','" + Detail +
"', '" + Username + "',getdate())";
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        cmd.ExecuteNonQuery();

        SQLCon.Close();
        return ("Το νέο θέμα δημιουργήθηκε");
    }
    catch (Exception e)
    {
        return ("Εμφανίστηκε πρόβλημα " + e.Message);
    }
}

```

//αυτή η μέθοδος χρησιμοποιείται για να εισάγει μια νέα απάντηση σε ένα θέμα από το φόρουμ στην βάση στον πίνακα FORUM_ANSERS

```

public String Topic_Insert_Answer(int q_id, String Username,
String Answer)
{
    try
    {
//δημιουργώ το connection
        SqlConnection SQLCon = new SqlConnection();

```

```

//δημιουργώ το query
string strSQL = "INSERT INTO FORUM_ANSWERS
(Q_ID,USERNAME,ANSWER,DATE) VALUES (" + q_id + "','" + Username +
"',','" + Answer + "','getdate())";
//δημιουργώ το connection string
SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
//δημιουργώ το command
SQLCon.Open();
SqlCommand cmd = new SqlCommand();
cmd.Connection = SQLCon;
cmd.CommandText = strSQL;
//τρέχω το command
cmd.ExecuteNonQuery();

SQLCon.Close();
return ("Η απάντηση σας υποβλήθηκε");
}
catch (Exception e)
{
return ("Εμφανίστηκε πρόβλημα " + e.Message);
}
}

//αυτή η μέθοδος χρησιμοποιείται για να αναπροσαρμόσει τον αριθμό
των απαντήσεων για ένα συγκεκριμένο θέμα στο φόρουμ μετά από μια νέα
απάντηση
public void Update_Answers_Number(int id,int count)
{
//δημιουργώ το connection

SqlConnection SQLCon = new SqlConnection();
//δημιουργώ το query

string strSQL = "UPDATE FORUM_TOPIC SET REPLY=" + count + " WHERE ID
="+id;

try
{
//δημιουργώ το connection string

SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
SQLCon.Open();
//δημιουργώ το command

SqlCommand cmd = new SqlCommand();
cmd.Connection = SQLCon;
cmd.CommandText = strSQL;
cmd.ExecuteNonQuery();
//τρέχω το command

SQLCon.Close();

}
catch (Exception e)
{
}
}

```



```

//αυτή η μέθοδος εισάγει μια νέα παραγγελία στον πίνακα ORDERS
παίρνει τιμές από το καλάθι αγορών μετά από την αποδοχή για πληρωμή
public String New_Order(String id, String username,String
payment,String price,String name,String address, String city, String
country, String telephone,String email)
{
    try
    {
        SqlConnection SQLCon = new SqlConnection();
//δημιουργώ το string του query
        string strSQL = "INSERT INTO ORDERS
(ID,USERNAME,PAYMENT,TOTAL_PRICE,ORDER_DATE,NAME,ADDRESS,CITY,COUNTRY
,TELEPHONE,EMAIL) VALUES ('" + id + "','" + username + "','" +
payment + "','" + price + "','getdate(),'"+ name + "','" + address +
 "','" + city + "','" + country + "','" + telephone + "','" + email +
"')";
//το connection string
SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";

        SQLCon.Open();
//δημιουργώ το command
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
//τρέχω το command

        cmd.ExecuteNonQuery();

        SQLCon.Close();
        return ("Η Παραγγελία σας πραγματοποιήθηκε");
    }
    catch (Exception e)
    {
        return ("Εμφανίστηκε πρόβλημα " + e.Message);
    }
}

// αυτη η μέθοδος εισάγει τις λεπτομέρειες κάθε παραγγελίας( pet
id, quantity) στον πίνακα ORDER_DETAILS
public void Insert_Order_Details(String order_id,int prod_id,int
quantity)
{
    try
    {
//δημιουργώ το connection

        SqlConnection SQLCon = new SqlConnection();
//δημιουργώ το string του query

        string strSQL = "INSERT INTO ORDER_DETAILS
(ORDER_ID,PROD_ID,QUANTITY) VALUES ('" + order_id + "','" + prod_id +
"," + quantity + "')";
//το connection string

        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();

```

```

        //δημιουργώ το command

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
//τρέχω το command

        cmd.ExecuteNonQuery();

        SQLCon.Close();
    }
    catch (Exception e)
    {
    }
}

//Update_Pet_Quantities μέθοδο αναπροσαρμόζει τις υπομείναντες
ποσότητες pet στο κατάστημα μετά απο κάθε νέα παραγγελία
public void Update_Pet_Quantities(int id, int sold_quantity)
{
    SqlConnection SQLCon = new SqlConnection();
    string strSQL = "UPDATE Pets SET Units_Stock=Units_Stock - "
+ sold_quantity + " WHERE Pet_ID =" + id;

    try
    {

//δημιουργώ το connection string
        SQLCon.ConnectionString = @"Data Source=
        .\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF;
        Integrated Security = True; User Instance = True";
        SQLCon.Open();
//δημιουργώ το command
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
//τρέχω το command

        cmd.ExecuteNonQuery();
        SQLCon.Close();
    }
    catch (Exception e)
    {
    }
}

//Update_Delivery_Status μέθοδο χρησιμοποιείται για να
αναπροσαρμόσει την κατάσταση της παραγγελίας,
//έστι ώστε ο χρήστης να μπορεί
//να δει πως είναι η κατάσταση της παραγγελίας του οποιαδήποτε
στιγμή
public void Update_Delivery_Status(String id,Order_Status status)
{
    SqlConnection SQLCon = new SqlConnection();
    string strSQL = "UPDATE ORDERS SET
    AVAILABILITY='"+status.availability+"', PACKING='"+status.packing+"',
    SENDING='"+status.sending+"', DELIVERED='"+status.delivered+"' WHERE
    ID ='"+id+"'";

    try

```

```

    {
//δημιουργώ το connection

        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
        SQLCon.Open();
//δημιουργώ το command

        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
//τρέχω το command
        cmd.ExecuteNonQuery();
        SQLCon.Close();

    }
    catch (Exception e)
    {
    }
}

//αυτή η μέθοδος επιστρέφει την κατάσταση από μια συγκεκριμένη
παραγγελία και την εμφανίζει στην σελίδα 'Οι Παραγγελίες μου' για να
πληροφορήσει τον χρήστη για την παραγγελία του
public Order_Status Get_Order_Status(String id)
{
    Order_Status mystatus = new Order_Status();

    SqlConnection SQLCon = new SqlConnection();
    string strSQL = "SELECT
AVAILABILITY,PACKING,SENDING,DELIVERED FROM ORDERS where ID='" + id +
"'";
    SqlDataReader rdr = null;

    try
    {
//δημιουργώ το connection
        SQLCon.ConnectionString = @"Data Source=
.\SQLEXPRESS;AttachDbFilename=|DataDirectory|ASPNETDB.MDF; Integrated
Security = True; User Instance = True";
//το ανοίγω
        SQLCon.Open();
//δημιουργώ το command
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = SQLCon;
        cmd.CommandText = strSQL;
        rdr = cmd.ExecuteReader();
//εδώ θα διαβάσω τα αποτελέσματα
//για κάθENA από τις απαντήσεις του query
//δημιουργώ ένα αντικείμενο του τύπου Order_Status
        while (rdr.Read())
        {
            mystatus = new Order_Status();
            mystatus.availability=rdr[0].ToString();
            mystatus.packing= rdr[1].ToString();
            mystatus.sending = rdr[2].ToString();
            mystatus.delivered = rdr[3].ToString();
        }
    }
}
//στο τελείωμα κλείνω τη σύνδεση

```

```
finally
{
    if (rdr != null)
    {
        rdr.Close();
    }
    if (SQLCon != null)
    {
        SQLCon.Close();
    }
}
return mystatus;
}
}
```

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. C.Darie, K.Watson - Beginning ASP.NET 2.0 E-Commerce in C# 2005 - From Novice to Professional. 2008, ISBN (pbk): 1-59059-468-1, Springer-Verlag New York
2. Korper, S. & Ellis, J. (2000) The E-Commerce book: Building the E-Empire. London: Academic Press
3. Rayport, J.F. and Jaworski, B.J. (2003) Introduction to E-Commerce. London:McGraw-Hill
4. Electronic Commerce, Fourth Edition, Pete Loshin and John Vacca, Charles River Media © 2004, ISBN:1584500646
5. e-commerce Patterns for Building B2C Web Sites Using IBM WebSphere Commerce Suite V5.1 John Ganci Sanjoy Banik Fabrizio Boaglio Ashish Cowlagi Miroslav Holecý Siva Kumar © Copyright International Business Machines Corporation, 2008
6. E-commerce Web site design: strategies and models H. Joseph Wen Houn-Gee Chen Hsin-Ginn Hwang, Information Management & Computer Security 9/1 [2001]
7. Palmer, J.W. and Griffith, D.A. (1998), ``An emerging model of Web site design for marketing'', Communications of the ACM, Vol. 41 No. 3, pp. 44-51.
8. Slywotzky, A.J. (2000) , ``The future of commerce'', Harvard Business Review, January-February, p. 39.
9. Watson, R.T., Akselsen, S. and Pitt, L.F. (1998), ``Attractors: building mountains in the flat landscape of the World Wide Web'', California Management Review, Vol. 40 No. 2, pp. 36-56.
10. Kalakota, R. and Whinston, A.B. (1996), Frontiers of Electronic Commerce, Addison-Wesley, Reading, MA.
11. Economics of Electronic Commerce (Publisher: Macmillan Computer Publishing) Author(s): Soon-yong Choi; Andrew Whinston; Dale Stahl ISBN: 1578700140 Publication Date: 07/22/97
12. Strategies for e-business, Creating value from e-commerce, Prentice hall, 2006, ISBN 0 273 68840 5 Authors: Tawfik Jelassi Albrecht Enders
13. The economics of the Electronic commerce SY Choi, DO Stahl-1997 Macmillan Technical Publishing 201 West 103rd Street Indianapolis, IN 46290 USA
14. Η αλυσίδα Αξίας (value chain) http://en.wikipedia.org/wiki/Value_chain
15. Σημειώσεις του κ. Σταματιάδη από την σελίδα www.sitemaker.gr/fstamatiadis/assets/chapter_1.doc
16. e-Business & Εφοδιαστική Αλυσίδα Πρόκληση ή Απειλή; <http://www.plant-management.gr/index.php?id=108>
17. Thomas John Allen, και Michael S. Scott Morton Information technology and the corporation of the 1990s

<http://books.google.gr/books?id=Vfyn0XbwnUIC&dq=Scott-Morton+και+Allen+1994>

18. Ariba Buyer Solutions <http://www.ariba.com/>
19. www.elexportal.com
20. Paul Timmers, Business Models for Electronic Markets, Electronic Markets vol. 8, no 2, 1998,
<http://www.electronicmarkets.org/modules/pub/view.php/electronicmarkets-183>
21. E-commerce Activity Chains , <http://www.jstor.org/pss/4132321>
22. www.in.gr
23. Peet, J. (2000) 'Survey: E-COMMERCE'. The Economist. Vol.354, pp. 89-131
24. Σιώμοκος, Ι. & Τσιάμης, Ι.Σ. (2004) Στρατηγικό Ηλεκτρονικό Μάρκετινγκ. Αθήνα: Εκδόσεις Αθ. Σταμούλης
25. Shaw, M., Blanning, R., Strader, T., & Whinston, A. (2000) Handbook on Electronic Commerce. New York: Springer

Ευχαριστίες

Με αφορμή την ολοκλήρωση της πτυχιακής μου εργασίας θα ήθελα να ευχαριστήσω τον καθηγητή μου κύριο Κλεισαρχάκη Μιχάλη που με εμπιστέυτηκε και μου ανάθεσε το θέμα της εργασίας.

Επιπλέον, ευχαριστώ την οικογένεια μου για την συμπαράσταση και κατανόηση που μου έδειξαν κατά την διάρκεια εκπόνησης της εργασίας.