

**ΑΤΕΙ ΗΡΑΚΛΕΙΟΥ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**ΣΧΕΔΙΑΣΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ ΠΛΑΤΦΟΡΜΑΣ
ΗΛΕΚΤΡΟΝΙΚΟΥ ΕΜΠΟΡΙΟΥ**

Σπουδάστρια:

Πυλαρινού Νίκη Α.Μ:707

Επιβλέπων Καθηγητής:

Ιωάννης Κοπανάκης, Επίκουρος καθηγητής

Ηράκλειο Απρίλιος 2008



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΕΙΣΑΓΩΓΗ.....	3
ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ ΣΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ.....	4
1.1 Τι είναι το "ηλεκτρονικό εμπόριο".....	4
1.3 Ηλεκτρονικό εμπόριο και CMS.....	6
1.2 Δομή.....	7
ΚΕΦΑΛΑΙΟ 2.ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ ΣΥΣΤΗΜΑΤΟΣ	
ΔΙΑΧΕΙΡΗΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ(CMS).....	8
2.1 Συστήματα Διαχείρισης Περιεχομένου CMS.....	8
2.2 Τεχνολογίες ανάπτυξης συστημάτων διαχείρισης περιεχομένου.....	9
2.3 Τεχνολογίες βάσεων δεδομένων.....	12
ΚΕΦΑΛΑΙΟ 3. ΑΝΑΛΥΣΗ ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ	
ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΗΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ(CMS).....	14
3.1 Σχεδιασμός βάσης δεδομένων.....	14
3.2 Ανάπτυξη συστήματος.....	19
ΚΕΦΑΛΑΙΟ 4. ΑΞΙΟΛΟΓΙΣΗ.....	52
ΚΕΦΑΛΑΙΟ 5. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	53
ΚΕΦΑΛΑΙΟ 6. ΒΙΒΛΙΟΓΡΑΦΙΑ.....	58



ΕΙΣΑΓΩΓΗ

Στο 1^ο κεφάλαιο, θα κάνουμε μια εισαγωγή στο ηλεκτρονικό εμπόριο και θα εξηγήσουμε τι ακριβώς εννοούμε με τον όρο Η.Ε. Θα καταγράψουμε τα είδη στα οποία διαχωρίζεται το Η.Ε και θα αναφερθούμε γιατί ολοένα και περισσότερες επιχειρήσεις οι οποίες χρησιμοποιούν μια εφαρμογή ηλεκτρονικού εμπορίου στρέφονται στη χρήση CMS(Συστήματα διαχείρισης περιεχομένου) για την διαχείριση των εφαρμογών τους.

Στο 2^ο κεφάλαιο θα αναλύσουμε τι είναι ένα CMS και θα αναφέρουμε κάποιες από τις πιο δημοφιλείς τεχνολογίες που υπάρχουν για τη δημιουργία ενός CMS. Επίσης θα αναφερθούμε σε κάποιες τεχνολογίες για τη δημιουργία βάσεων δεδομένων οι οποίες συνδέονται με τέτοιου είδους εφαρμογές για την αποθήκευση των δεδομένων.

Στο 3^ο κεφάλαιο θα ασχοληθούμε με την ανάλυση, το σχεδιασμό και την ανάπτυξη του συστήματος που έχουμε δημιουργήσει χρησιμοποιώντας μια συγκεκριμένη οντότητα του συστήματος μας, την οντότητα Ανακοινώσεις. Θα γίνει μια παρουσίαση των σελίδων του συστήματος αλλά και των δυνατοτήτων που προσφέρει.

Στο 4^ο κεφάλαιο θα γίνει μια αξιολόγηση της εφαρμογής μας μετά από το τεστάρισα μου πραγματοποιήσαμε.

Στο 5^ο κεφάλαιο θα παρουσιάσουμε τα συμπεράσματα που προέκυψαν από την υλοποίηση αυτής της εφαρμογής και των εργαλείων που χρησιμοποιήσαμε. Θα καταγράψουμε τα πλεονεκτήματα που προσφέρει η τεχνολογία που χρησιμοποιήσαμε για τη βελτιστοποίηση της εφαρμογής μας και τέλος θα αναφερθούμε σε μελλοντικές εργασίες που θα μπορούσαν να γίνουν αλλά και πώς θα μπορούσε να βελτιωθεί η εφαρμογή.

Στο 6^ο κεφάλαιο αναφέρουμε την βιβλιογραφία που χρησιμοποιήσαμε για την εκπόνηση της παρούσας πτυχιακής εργασίας.

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ ΣΤΟ ΗΛΕΚΤΡΟΝΙΚΟ ΕΜΠΟΡΙΟ

1.1 Τι είναι το "ηλεκτρονικό εμπόριο";

Το ηλεκτρονικό εμπόριο αποτελεί αναπόσπαστο κομμάτι του παγκοσμίου εμπορίου στις ημέρες μας. Για πολλούς θεωρείται ίσως η δεύτερη μεγαλύτερη τεχνολογική εξέλιξη μετά τη βιομηχανική επανάσταση, καθώς εξοικονομεί χρόνο και χρήμα και μπορεί να μεταμορφώσει μια μικρή εταιρία ακόμα και σε κολοσσό. Ο όρος Ηλεκτρονικό Εμπόριο (ΗΕ) χρησιμοποιείται για να περιγράψει την χρήση τηλεπικοινωνιακών μέσων (κυρίως δικτύων) για κάθε είδους εμπορικές συναλλαγές ή επιχειρηματικές δραστηριότητες μεταξύ επιχειρήσεων και ιδιωτών. Με άλλα λόγια, κάθε "εμπορική" δραστηριότητα που πριν από μερικά χρόνια ήταν δυνατή, μόνο χάρη στην φυσική παρουσία και μεσολάβηση ανθρώπων ή υλικών μέσων (π.χ. εμπορική αλληλογραφία), σήμερα μπορεί να επιτευχθεί αυτόματα, ηλεκτρονικά και εξ' αποστάσεως. Βλέπουμε λοιπόν, ότι το ΗΕ δεν αποτελεί μία και μόνη τεχνολογία. Πρόκειται περισσότερο για ένα συνδυασμό τεχνολογιών ανταλλαγής δεδομένων (όπως το EDI (την Ηλεκτρονική Ανταλλαγή Δεδομένων (Electronic Data Interchange- EDI) και το email), πρόσβασης σε δεδομένα (shared databases, electronic bulletin boards - BBS, το World Wide Web κ.ά.) και αυτόματης συλλογής δεδομένων (bar coding, magnetic/optical character recognition).

Το ΗΕ προσπαθεί να αναπτύξει την εκτέλεση των επιχειρησιακών συναλλαγών μέσα από διάφορα δίκτυα. Αυτές οι αναπτύξεις αναφέρονται σε μεγαλύτερη απόδοση (καλύτερη ποιότητα, μεγαλύτερη ικανοποίηση πελατών και καλύτερη λήψη αποφάσεων), μεγαλύτερη οικονομική χρησιμότητα (χαμηλότερο κόστος) και πιο γρήγορες συναλλαγές (μεγάλη ταχύτητα, αλληλεπίδραση πραγματικού χρόνου). Το όραμα όσων ασχολούνται με το ΗΕ είναι η ομογενοποίηση (χάρη στην εφαρμογή νέων τεχνολογιών) όλων των οικονομικών λειτουργιών των επιχειρήσεων και οργανισμών με τέτοιο τρόπο που κάθε δραστηριότητα να μπορεί :

α) να εκτελείται σε ηλεκτρονική μορφή

β) να μεταφέρεται εύκολα από τον ένα συναλλασσόμενο στον άλλο (π.χ. ένα ψηφιακό τιμολόγιο από τον πωλητή στον αγοραστή)

γ) να είναι προσιτή σε κάθε μέλος της ηλεκτρονικής οικονομικής κοινότητας (π.χ. να μπορεί οποιοσδήποτε να βρει τον τιμοκατάλογο ή τα τεχνικά χαρακτηριστικά των



προϊόντων μου άμεσα και με δικές τους ενέργειες, χωρίς να χρειαστεί δική μου μεσολάβηση π.χ. να μου τα ζητήσει και να του τα στείλω).

Συνοψίζοντας, ένας πιθανός ορισμός του Η.Ε. είναι : "Το Η.Ε. είναι οποιαδήποτε μορφή επιχειρησιακής συναλλαγής, οι συντελεστές της οποίας αλληλεπιδρούν με ηλεκτρονική μορφή περισσότερο παρά με φυσικές συναλλαγές ή διαμέσου φυσικής επικοινωνίας".

Το ηλεκτρονικό εμπόριο μπορεί να οριστεί από τέσσερις διαφορετικές οπτικές γωνίες:

- ❖ **Επιχειρήσεις:** Ως εφαρμογή νέων τεχνολογιών προς την κατεύθυνση του αυτοματισμού των συναλλαγών και της ροής εργασιών.
- ❖ **Υπηρεσίες:** Ως μηχανισμός που έχει στόχο να ικανοποιήσει την κοινή επιθυμία προμηθευτών και πελατών για καλύτερη ποιότητα υπηρεσιών, μεγαλύτερη ταχύτητα εκτέλεσης συναλλαγών και μικρότερο κόστος.
- ❖ **Απόσταση:** Ως δυνατότητα αγοραπωλησίας προϊόντων και υπηρεσιών μέσω του Internet ανεξάρτητα από τη γεωγραφική απόσταση.
- ❖ **Επικοινωνία:** Ως δυνατότητα παροχής πληροφοριών, προϊόντων ή υπηρεσιών, και πληρωμών μέσα από δίκτυα ηλεκτρονικών υπολογιστών.

Το ηλεκτρονικό εμπόριο σε πρακτικό επίπεδο, μπορεί να πάρει πολλές μορφές:

Εσωτερικό εμπόριο: Στόχος είναι η αποτελεσματικότερη λειτουργία των δραστηριοτήτων μιας επιχείρησης, ώστε να μπορεί να προσφέρει καλύτερα προϊόντα και υπηρεσίες στους πελάτες της. Οι εφαρμογές του συνήθως εντάσσονται στη λειτουργία ενός τοπικού δικτύου (Intranet) και μπορούν να είναι: επικοινωνία μεταξύ ομάδων εργασίας, ηλεκτρονική δημοσίευση (άμεση διανομή πληροφοριών) κτλ.

Συναλλαγές μεταξύ επιχειρήσεων (Business-to-Business-B2B): Το ηλεκτρονικό εμπόριο επιτρέπει σε επιχειρήσεις να βελτιώσουν τη μεταξύ τους συνεργασία, απλοποιώντας τις διαδικασίες και το κόστος των προμηθειών, την ταχύτερη αποστολή των προμηθειών και τον αποτελεσματικότερο έλεγχο του επιπέδου αποθεμάτων. Επιπλέον καθιστά ευκολότερη την αρχειοθέτηση των σχετικών εγγράφων και ποιοτικότερη την εξυπηρέτηση πελατών. Η δυνατότητα ηλεκτρονικής σύνδεσης με προμηθευτές και διανομείς καθώς και η πραγματοποίηση ηλεκτρονικών πληρωμών βελτιώνουν ακόμη περισσότερο την αποτελεσματικότητα: οι ηλεκτρονικές πληρωμές



περιορίζουν το ανθρώπινο σφάλμα, αυξάνουν την ταχύτητα και μειώνουν το κόστος των συναλλαγών. Το ηλεκτρονικό εμπόριο προσφέρει τη δυνατότητα αυξημένης πληροφόρησης σχετικά με τα προσφερόμενα προϊόντα είτε από τους προμηθευτές είτε από ενδιάμεσους οργανισμούς που προσφέρουν υπηρεσίες ηλεκτρονικού εμπορίου.

Λιανικές πωλήσεις - Ηλεκτρονικό εμπόριο μεταξύ επιχείρησης και καταναλωτών (Business-to-Consumer - B2C): Πρόκειται για την πιο διαδεδομένη μορφή ηλεκτρονικού εμπορίου. Ο καταναλωτής έχει πρόσβαση σε μια τεράστια ποικιλία προϊόντων σε δικτυακούς κόμβους-καταστήματα, βλέπει, επιλέγει, αν επιθυμεί να αγοράσει είδη ένδυσης μπορεί ενίοτε και να τα δοκιμάζει (μέσω ειδικών προγραμμάτων), ανακαλύπτει προϊόντα τα οποία δεν θα μπορούσε να βρει εύκολα στη χώρα του, συγκρίνει τιμές και τέλος αγοράζει. Κι όλα αυτά χωρίς να βγει από το σπίτι του, κερδίζοντας πολύτιμο χρόνο και κόπο.

1.3 Ηλεκτρονικό εμπόριο και CMS

Σήμερα ολοένα και περισσότερες επιχειρήσεις στρέφονται στο ηλεκτρονικό εμπόριο με σκοπό να αυτοματοποιούν και να βελτιώνουν όχι μόνο τις διαδικασίες της επιχείρησης τους αλλά και τις υπηρεσίες που παρέχουν.

Πριν από μερικά χρόνια ο μόνος τρόπος για να μπορέσει μια εταιρία να εδραιώσει την παρουσία της στο διαδίκτυο μέσω μιας εφαρμογής ηλεκτρονικού εμπορίου ήταν ένας. Η συνεργασία με μία εταιρεία (εξωτερική συνεργασία) η οποία θα δημιουργούσε την εφαρμογή αλλά και θα την συντηρούσε ως προς την ανάπτυξη αλλά και το περιεχόμενο της. Σαν αποτέλεσμα και περιεχόμενο αλλά και μελλοντικές υπηρεσίες καθυστερούσαν αρκετά να βγουν στην παραγωγή (online) γιατί οποιαδήποτε αλλαγή στην εφαρμογή μπορούσε να πραγματοποιηθεί μόνο από την τρίτη εταιρεία.

Με την ανάπτυξη αλλά και την διάδοση ειδικά των γλωσσών αντικειμενοστραφούς προγραμματισμού αλλά και των μοντέλων προγραμματισμού (model view controller – 3 tier - frameworks) ένας νέος τρόπος ανάπτυξης εφαρμογών διαδικτύου άνθισε. Ο τρόπος αυτός είναι τα συστήματα CMS (Σύστημα Διαχείρισης Περιεχομένου).



1.3 Δομή

Στα επόμενα κεφάλαια θα δούμε τι είναι ένα CMS αλλά και κάποιες από τις τεχνολογίες που υπάρχουν για την ανάπτυξη συστημάτων διαχείρισης περιεχομένου και τεχνολογίες για την δημιουργία βάσεων δεδομένων. Έπειτα θα αναλύσουμε την πρόταση μας για την δημιουργία ενός συστημάτων διαχείρισης περιεχομένου και θα δούμε πως έχει σχεδιαστεί και τι δυνατότητες έχει. Θα αναφερθούμε στα προγράμματα που θα χρησιμοποιήσουμε, θα αξιολογήσουμε την εφαρμογή μας και θα καταγράψουμε τα συμπεράσματα μας.



ΚΕΦΑΛΑΙΟ 2. ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΠΤΥΞΗΣ CMS

2.1 CMS (Content Management System)

Το CMS, Σύστημα Διαχείρισης Περιεχομένου στα ελληνικά, είναι μια εφαρμογή με την οποία μπορεί κανείς να διαχειριστεί συγκεντρωμένες πληροφορίες ηλεκτρονικής μορφής.

Σχετικά με το Διαδίκτυο, όταν αναφερόμαστε σε ένα CMS, μιλάμε για ένα σύστημα διαχείρισης του περιεχομένου ενός διαδικτυακού τόπου (Web CMS). Αποτελείται από ένα πλήθος εργαλείων, με τα οποία δίνεται η δυνατότητα σε όποιον ανανεώνει μια ιστοσελίδα να περνάει καινούριο περιεχόμενο, να αλλάζει τα κείμενα, τις φωτογραφίες, τα αρχεία και οτιδήποτε άλλο, εύκολα και γρήγορα.

Οι εφαρμογές διαχείρισης περιεχομένου επιτρέπουν την αλλαγή του περιεχομένου χωρίς να είναι απαραίτητες ειδικές γνώσεις σχετικές με τη δημιουργία ιστοσελίδων ή γραφικών, καθώς συνήθως τα κείμενα γράφονται μέσω κάποιων online WYSIWYG ("What You See Is What You Get") html editors, ειδικών δηλαδή κειμενογράφων, παρόμοιων με το MS Word, που επιτρέπουν τη μορφοποίηση των κειμένων όποτε υπάρχει ανάγκη.

Όπως προαναφέραμε οι περισσότερες επιχειρήσεις στρέφονται προς την χρήση ενός CMS για την εύκολη διαχείριση των ιστοσελίδων τους. Στο εμπόριο υπάρχουν πολλά CMS τα οποία έχουν υλοποιηθεί με διαφορετικές τεχνολογίες. Παρακάτω θα δούμε τα πιο δημοφιλή εργαλεία για την δημιουργία τέτοιων συστημάτων.



2.2 Τεχνολογίες ανάπτυξης Συστημάτων Διαχείρισης Περιεχομένου

Στην αγορά υπάρχουν και χρησιμοποιούνται πολλές τεχνολογίες για την ανάπτυξη συστημάτων διαχείρισης περιεχομένου. Οι πιο δημοφιλείς είναι η τεχνολογία PHP, η ASP.NET, JSP, η Perl, η Fusion Middleware της Oracle, η Ruby αλλά και άλλες λιγότερο δημοφιλείς.

Η τεχνολογία Asp.Net

Το Asp.Net είναι μια τεχνολογία προγραμματισμού για το internet της Microsoft, με μια πιο οργανωμένη, αντικειμενοστραφή προσέγγιση στη δημιουργία δυναμικών εφαρμογών για το web. Με την υπάρχουσα τεχνολογία των Active Server Page, ο κώδικας που εκτελείται στο διακομιστή ,συνδυάζεται και συνεργάζεται με την HTML από την πλευρά του πελάτη. Το Asp.Net είναι ένα προγραμματιστικό μοντέλο ανεξάρτητο από το πρόγραμμα περιήγησης καθώς οι εφαρμογές Asp.Net τρέχουν με όλες τις τελευταίες εκδόσεις των προγραμμάτων περιήγησης καθώς επίσης και με παλαιότερες εκδόσεις.

Ποιες γλώσσες προγραμματισμού μπορούν να χρησιμοποιηθούν με το Asp.Net

Το .Net Framework προσδιορίζει ότι οι εφαρμογές μπορούν να γραφούν σε οποιαδήποτε γλώσσα προγραμματισμού υποστηρίζει το Common Language Runtime(CLR). Το Common Language Runtime μεταφράζει οποιαδήποτε εφαρμογή υποστηρίζει το CLR σε Microsoft Intermediate Language (ενδιάμεση γλώσσα της Microsoft), ή Αυτός ο κώδικας ενδιάμεσου επιπέδου μεταγλωττίζεται στη συνέχεια για την πλατφόρμα στην οποία θα εκτελεστεί. Με αυτή την διαδικασία μπορούμε εύκολα να δημιουργούμε μια εφαρμογή σε μια γλώσσα και να χρησιμοποιείται σε οποιαδήποτε λειτουργικό σύστημα, χρησιμοποιώντας CLR.

Για την τελευταία έκδοση του .Net Framework η Microsoft παρουσίασε τέσσερις διαφορετικές γλώσσες που μπορούν να χρησιμοποιηθούν για κώδικα που υποστηρίζει το CLR.

Τι είναι το .Net Framework

Το .Net Framework είναι η υποδομή για την πλατφόρμα Microsoft .Net

Το .Net Framework είναι ένα περιβάλλον το οποίο μας δίνει τη δυνατότητα κατασκευής και εκτέλεσης Web εφαρμογών και Web services.



Η πρώτη server τεχνολογία ASP (Active Server Pages) ήταν μια ισχυρή και ευέλικτη γλώσσα προγραμματισμού, ωστόσο ήταν αρκετά αντικειμενοστραφής. Δεν ήταν ένα application framework αλλά ούτε και ένα enterprise development tool.

Το Microsoft .Net Framework δημιουργήθηκε για να λύσει αυτά τα προβλήματα και προσφέρει τα παρακάτω

- ❖ Ευκολότερο και ταχύτερο προγραμματισμό
- ❖ Μείωση του όγκου του κώδικα
- ❖ Πλουσιότερο server control ιεραρχικά με τα events
- ❖ Περισσότερες class libraries
- ❖ Μεγαλύτερη υποστήριξη για εργαλεία

Το .Net Framework περιέχει 3 κύρια τμήματα

Γλώσσες προγραμματισμού

- ❖ C# (C sharp)
- ❖ Visual Basic (VB.Net)
- ❖ J# (J sharp)

Τεχνολογίες Server και Client

- ❖ ASP.NET (Active Server Pages)
- ❖ Windows Forms (Windows desktop solutions)
- ❖ Compact Framework (PDA / Mobile solutions)

Development environments

- ❖ Visual Studio .NET (VS .NET)
- ❖ Visual Web Developer



Τεχνολογία JSP

Η μεγάλη ανάπτυξη της Java και ειδικά τον τελευταίο χρόνο που η Sun της έδωσε GPL άδεια και την έκανε open source, την οδήγησε ένα βήμα πέρα από την αρχικό της σκοπό (smart home). Τα δημοφιλή SDK της Java που ακολουθούσαν κάθε νέα της έκδοση έδωσαν τη δυνατότητα στους προγραμματιστές να εκμεταλλευτούν το πλήθος των έτοιμων βιβλιοθηκών και να οδηγήσουν την Java πέρα από τα στενά όρια της desktop εφαρμογής σε web applications και μετέπειτα σε cms. Η δυνατότητα της Java, να παράγει xml αλλά και html την έκανα μία ιδανική γλώσσα για την δημιουργία cms, και την μετέτρεψαν από Java Enterprise SDK σε JavaServer Pages (jsp), η γνωστή κατάληξη που βλέπουμε σε μεγάλα γνωστά sites όπως της sun, που είναι όχι μόνο το σπίτι της Java και του Solaris αλλά και πολλών άλλων project ελευθέρου λογισμικού όπως το open office.

Το δυνατό σημείο της jsp σε αντίθεση με άλλες scripting γλώσσες (αν και η jsp δε θα μπορούσε να χαρακτηριστεί απλά μία scripting γλώσσα) είναι η ανεξαρτησία της από οποιαδήποτε πλατφόρμα. Όπως και η java βασίζεται στην εικονική μηχανή για να παράγει τον τελικό binary code και όπως και η εικονική μηχανή είναι ανεξάρτητη πλατφόρμας (και στη συγκεκριμένη περίπτωση λειτουργικού συστήματος) έτσι και η jsp μπορεί να τρέξει όχι μόνο σε οποιαδήποτε λειτουργικό σύστημα αλλά και να συνδυαστεί χωρίς πρόβλημα η παραμετροποίηση κώδικα με άλλα πακέτα λογισμικού, όπως η mysql, oracle ακόμα και SQL server ή IBM DB2.

Τεχνολογία PHP

Η PHP είναι μια γλώσσα προγραμματισμού που σχεδιάστηκε για τη δημιουργία δυναμικών σελίδων στο διαδίκτυο και είναι επισήμως γνωστή ως: HyperText preprocessor. Είναι μια server-side (εκτελείτε στον διακομιστή) scripting γλώσσα που γράφεται συνήθως πλαισιωμένη από HTML, για μορφοποίηση των αποτελεσμάτων. Αντίθετα από μια συνηθισμένη HTML σελίδα η σελίδα PHP δεν στέλνεται άμεσα σε έναν πελάτη (client), αντ' αυτού πρώτα αναλύεται και μετά αποστέλλεται το παραγόμενο αποτέλεσμα. Τα στοιχεία HTML στον πηγαίο κώδικα μένουν ως έχουν, αλλά ο PHP κώδικας ερμηνεύεται και εκτελείται. Ο κώδικας PHP μπορεί να θέσει ερωτήματα σε βάσεις δεδομένων, να δημιουργήσει εικόνες, να διαβάσει και να γράψει αρχεία, να συνδεθεί με απομακρυσμένους υπολογιστές, κ.ο.κ. Σε γενικές γραμμές οι δυνατότητες που μας δίνει είναι απεριόριστες.

Αρχικά η ονομασία της ήταν PHP/FI από το Forms Interpreter η οποία δημιουργήθηκε το 1995 από τον Rasmus Lerdorf ως μια συλλογή από Perl scripts που τα χρησιμοποιούσε στην προσωπική του σελίδα. Δεν άργησε να τα εμπλουτίσει με λειτουργίες επεξεργασίας δεδομένων με SQL, αλλά τα σημαντικά βήματα που έφεραν και την μεγάλη αποδοχή της PHP ήταν αρχικά η μετατροπή τους σε C και μετέπειτα η δωρεάν παροχή του πηγαίου κώδικα μέσω της σελίδας του ώστε να επωφεληθούν όλοι από αυτό που είχε φτιάξει, αλλά και να τον βοηθήσουν στην περαιτέρω ανάπτυξή της.



2.3 Τεχνολογίες βάσεων δεδομένων

Σχεδόν όλες οι εφαρμογές που έχουν σχεδιαστεί για τον επιχειρηματικό κόσμο βασίζονται σε κάποιου είδους αποθήκευση δεδομένων. Είτε χρησιμοποιείται ένα λογιστικό φύλλο excel, είτε ένα αρχείο κειμένου, είτε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), όπως είναι ο SQL Server ή η Oracle.

Μια βάση δεδομένων (database) είναι μια συλλογή πληροφοριών που έχει οργανωθεί με τέτοιο τρόπο ώστε να είναι εύκολη η ανάκληση των επιθυμητών πληροφοριών από αυτήν. Είναι σαν μια βιβλιοθήκη αρχειοθέτησης στην οποία όλα τα αρχεία είναι οργανωμένα με τέτοιο τρόπο ώστε να είναι εύκολη η ανάκληση τους.

Υπάρχουν απλές βάσεις δεδομένων οι οποίες απευθύνονται σε μικρού μεγέθους εφαρμογές, που μπορούν να αξιοποιηθούν από έναν απλό χρήστη, μια μικρή επιχείρηση, γραφείο κ.λπ. και δεν απαιτούν τη χρήση πρόσθετου εξοπλισμού. Οι επαγγελματικές βάσεις δεδομένων απευθύνονται σε μεγάλες επιχειρήσεις, με εκατοντάδες ή χιλιάδες εργαζομένους. Οι ευρύτερα χρησιμοποιούμενες από αυτές είναι:

MS SQL Server

Είναι ένα σύστημα διαχείρισης βάσεων δεδομένων (relational database management system, RDBMS), που έχει αναπτυχθεί και προωθείται από την Microsoft. Το σύστημα αυτό είναι το σημαντικότερο τμήμα του Microsoft Back Office, ενός εμπορικού πακέτου εφαρμογών πελάτη/διακομιστή (client/server). Το MS SQL Server εκτελείται αποκλειστικά στα λειτουργικά συστήματα της Microsoft. Αυτή η ιδιαιτερότητα έχει και ένα πολύ σημαντικό μειονέκτημα: δεν μπορεί να εκμεταλλευτεί τις προχωρημένες ιδιότητες ενός λειτουργικού συστήματος όπως το UNIX, το οποίο, σε μερικές περιοχές, όπως οι εμπλουτισμένες παράλληλες αρχιτεκτονικές ή εταιρικές εφαρμογές, συνεχίζει να έχει ορισμένα πλεονεκτήματα σε σχέση με τα Windows. Τα σημαντικότερα στοιχεία του SQL Server είναι:

α) Είναι εύκολο στην χρήση

β) Μπορεί να χρησιμοποιηθεί από έναν φορητό υπολογιστή μέχρι σε συστήματα συμμετρικών πολυεπεξεργαστών (SMP).



γ) Παρέχει χαρακτηριστικά αποθηκών δεδομένων, που μέχρι τώρα διατίθενται μόνο στην Oracle και σε πιο ακριβά συστήματα DBMS.

Το SQL Server ήταν από την αρχή σχεδιασμένο σαν ένα DBMS πελάτη/διακομιστή. Η αρχιτεκτονική πελάτη/διακομιστή αναπτύχθηκε για να γίνεται διαχείριση ενός μεγάλου αριθμού υπολογιστών (PC, σταθμών εργασίας, SMP), που είναι συνδεδεμένοι σε ένα δίκτυο. Η λειτουργικότητα του SQL Server διαιρείται ανάμεσα σε πελάτες και διακομιστές. Ένας πελάτης παρέχει μια ή περισσότερες διασυνδέσεις χρήστη, που χρησιμοποιούνται για δημιουργία μιας αίτησης χρήστη προς ένα DBMS. Ο διακομιστής (δηλ. το DBMS) επεξεργάζεται αυτή την αίτηση και στέλνει το αποτέλεσμα πίσω στον πελάτη.

Oracle Database

Είναι όπως η MS SQL SERVER μια σχεσιακή βάση δεδομένων. Η Oracle μια από τις μεγαλύτερες εταιρίες έχει δέσει στην Oracle Database 10g η οποία είναι η τελευταία έκδοση του λογισμικού της για βάσεις δεδομένων κάποια χαρακτηριστικά που ξεχωρίζουν την Oracle Database 10g από τα ανταγωνιστικά προϊόντα. Η Oracle Database 10g παρέχει το clustering και το workload balancing, την υψηλή διαθεσιμότητα και την αυτοματοποιημένη διαχείριση που απαιτούνται για την εφαρμογή των υποδομών επιχειρηματικού Grid Computing. Η Oracle Database 10g, διαθέσιμη από το 2004, παρέχει ενισχυμένα χαρακτηριστικά που διευκολύνουν την υλοποίηση ενώ απλοποιούν τη διαχείριση της βάσης δεδομένων. Χαρακτηριστικά όπως το Automatic Storage Management, το οποίο απαλείφει, εικονικά, την ανάγκη για χειροκίνητη παραμετροποίηση και ρύθμιση του database storage, και οι εξελιγμένες XML δυνατότητες .

Υπάρχουν βέβαια και λύσεις από τον χώρο των open source εφαρμογών όπως είναι η βάση δεδομένων MySQL, η βάση δεδομένων SQLite και η PostgreSQL.



Κεφάλαιο 3. ΑΝΑΛΥΣΗ ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΗΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ

Σε αυτό το κεφάλαιο θα γίνει μια αναλυτική παρουσίαση των οντοτήτων του CMS που έχουμε σχεδιάσει αλλά και της βάσης δεδομένων. Θα δούμε αναλυτικά από τι αποτελείται μια συγκεκριμένη οντότητα, η οντότητα ανακοίνωση, πως έχει δημιουργηθεί ο πίνακας, η φόρμα αλλά και τι δυνατότητες έχει. Η ανάπτυξη της εφαρμογής θα γίνει με τεχνολογία ASP.NET και η βάση δεδομένων θα υλοποιηθεί σε MS SQL Server.

Το θέμα της παρούσας πτυχιακής εργασίας είναι η υλοποίηση ενός CMS για την διαχείριση του περιεχομένου ενός portal marketing. Με τη χρήση αυτού του συστήματος ο χρήστης θα μπορεί εύκολα και γρήγορα να εισάγει δεδομένα στη σελίδα, να διαγράφει δεδομένα, να αλλάζει το περιεχόμενο της σελίδας αλλά και να φορτώνει αρχεία ή εικόνες. Στόχος αυτού του portal είναι η διάχυση πληροφορίας μεταξύ των επαγγελματιών και ακαδημαϊκών στον ευρύτερο χώρο του marketing. Ο χρήστης θα έχει τη δυνατότητα να ενημερώνεται για διάφορα θέματα όπως θέσεις εργασίας, μεταπτυχιακά στην Ελλάδα και στο εξωτερικό αλλά και να ενημερώνεται για διάφορα θέματα marketing.

3.1 Σχεδιασμός βάσης δεδομένων

Το σύστημα μας αποτελείται από τις παρακάτω οντότητες :

Επαφές : Οντότητα που περιέχει τα στοιχεία των χρηστών.

Ανακοινώσεις : Οντότητα για την καταχώρηση ανακοινώσεων.

Πρόγραμμα Σπουδών : Οντότητα για την καταχώρηση προγραμμάτων σπουδών.

Εταιρία : Οντότητα που περιγράφει μια εταιρία.

Term : Οντότητα που περιέχει διάφορους ορισμούς που σχετίζονται με το marketing.

Video : Οντότητα που περιέχει video



Χρήστης :Οντότητα που περιγράφει τους χρήστες του συστήματος

ProgSpudonCategory : Περιγράφει τον τύπο ενός προγράμματος σπουδών

AnakoinosiType : Περιγράφει τον τύπο μιας ανακοίνωσης

Σε κάθε οντότητα έχουμε τη δυνατότητα της καταχώρησης (insert), της ανανέωσης (update) και της διαγραφής (delete), αλλά και σε κάποιες από αυτές, τη δυνατότητα καταχώρησης αρχείων τύπου Excel, Word, Tiff, Jpeg, Gif, Text, Pdf, Bitmap, Zip και PowerPoint.

Οι διεργασίες που θα εκτελούνται είναι οι παρακάτω:

Εισαγωγή - Ανανέωση - Διαγραφή Επαφής

Εισαγωγή - Ανανέωση – Διαγραφή Ανακοίνωσης

Εισαγωγή - Ανανέωση – Διαγραφή Προγράμματος Σπουδών

Εισαγωγή - Ανανέωση – Διαγραφή Εταιρίας

Εισαγωγή - Ανανέωση – Διαγραφή Term

Εισαγωγή - Ανανέωση – Διαγραφή Video

Εισαγωγή - Ανανέωση – Διαγραφή Χρήστη

Σε αυτό το κεφάλαιο θα χρησιμοποιήσουμε μια συγκεκριμένη οντότητα της πλατφόρμας, την οντότητα Ανακοινώσεις, για να δούμε πώς λειτουργούν οι οντότητες αλλά και τον κώδικα που έχουμε χρησιμοποιήσει. Όμοια είναι σχεδιασμένες και οι υπόλοιπες οντότητες του συστήματος μας.

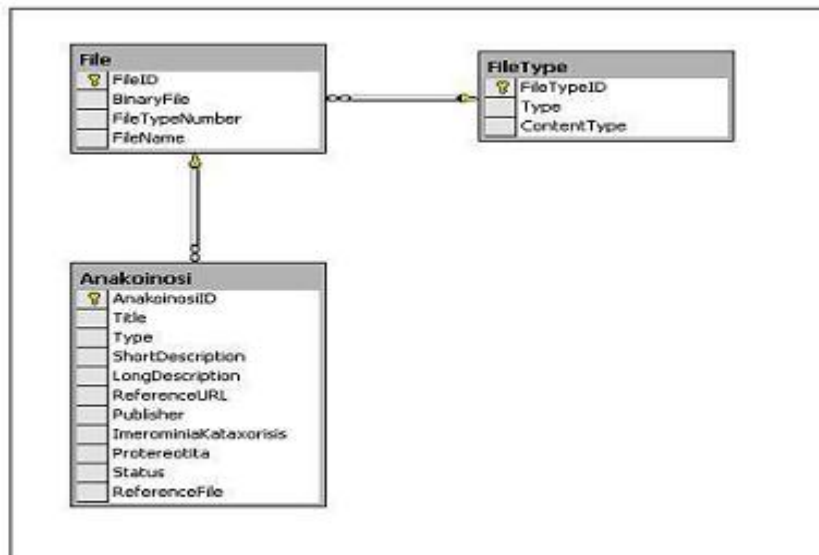
Η οντότητα ανακοίνωση, όπως και όλες οι οντότητες που έχουν τη δυνατότητα φόρτωσης αρχείων ,συνδέεται με την οντότητα File ως εξής:

Μια ανακοίνωση μπορεί να έχει μόνο ένα File, ενώ ένα File μπορεί να βρίσκεται σε μια ή περισσότερες ανακοινώσεις.

Η οντότητα File αποτελείται από τα εξής πεδία:

- Field: Είναι το αναγνωριστικό της οντότητας, είναι τύπου int, είναι το πρωτεύον κλειδί της οντότητας, είναι μοναδικό και αυξάνει αυτόματα κατά 1.
- BinaryFile: Εδώ αποθηκεύουμε τα bytes από τα οποία αποτελείται το αρχείο. Είναι τύπου image και χωράει μέχρι 2^{16} bytes.
- FileTypeNumber: Είναι foreign key όπου συνδέεται με την οντότητα filetype και μας δείχνει τι τύπου είναι το αρχείο που ανεβάσαμε στην βάση μας.
- FileName: Είναι το όνομα του αρχείου που ανεβάζουμε. Είναι αλφαριθμητικό (varchar) μέχρι 50 χαρακτήρες.

Η σύνδεση μεταξύ των 3 οντοτήτων είναι ως εξής.



Εικόνα 1. Το διάγραμμα της σύνδεσης μεταξύ τριών οντοτήτων



Οι οντότητες που υπάρχουν στην εφαρμογή μας είναι οι εξής:

Πίνακας Επαφές : ContactID, FirstName, LastName, Title, Department, SubDepartment, Sector, Company, AddrStreet, AddrProvince, AddrCity, AddrZip, AddrCountry, Email, URL, Telephone1, Telephone2, Mobile, Fax, MarketSector, Interests, Notes, Picture, Arxeio

Πίνακας Ανακοινώσεις : AnakoinosiID, Type, Title, AnakoinosiTypeID, ShortDescription, LongDescription, ReferenceURL, Publisher, ImerominiaKataxorisiss, Prottereotita, Status, ReferenceFile

Πίνακας Πρόγραμμα Σπουδών : ProgSpoudonID, Typos, ProgSpudonCategoryID, Titlos, Idrima, Diarkeia, Enarksi, AddrStreet, AddrProvince, AddCity, AddrTK, AddrCountry, ContactName, ContactTitle, Tel1, Tel2, Fax, email, url, Logo, ProgDesc, Category

Πίνακας Εταιρίες : EtairiaID, Eponimia, MarketSector, EtosLiturgias, Megethos, AddrStreet, AddrProvince, AddCity, AddrTK, AddrCountry, ContactName, ContactTitle, Tel1, Tel2, Fax, email, url, Logo, ProfileDoc, Priority, Status

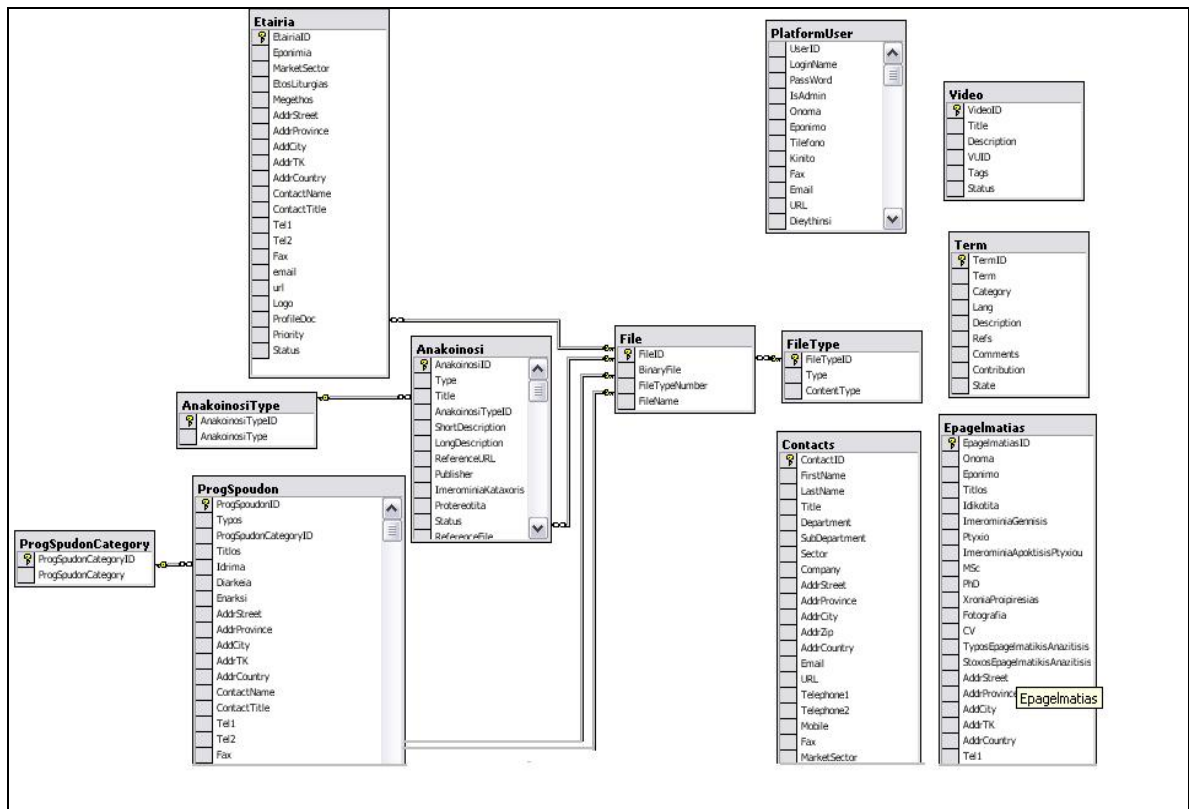
Πίνακας Term : TermID, Term, Category, Lang, Description, Refs, Comments, Contribution, State

Πίνακας Video : VideoID, Title, Description, VUID, Tags, Status

Πίνακας Χρήστης : UserID, LoginName, PassWord, IsAdmin, Onoma, Eponimo, Tilefono, Kinito, Fax, Email, URL, Dieythinsi, Taytotita, AFM, DOY, Eidikotita, Sxolia

Πίνακας ProgSpudonCategoryID, ProgSpudonCategory

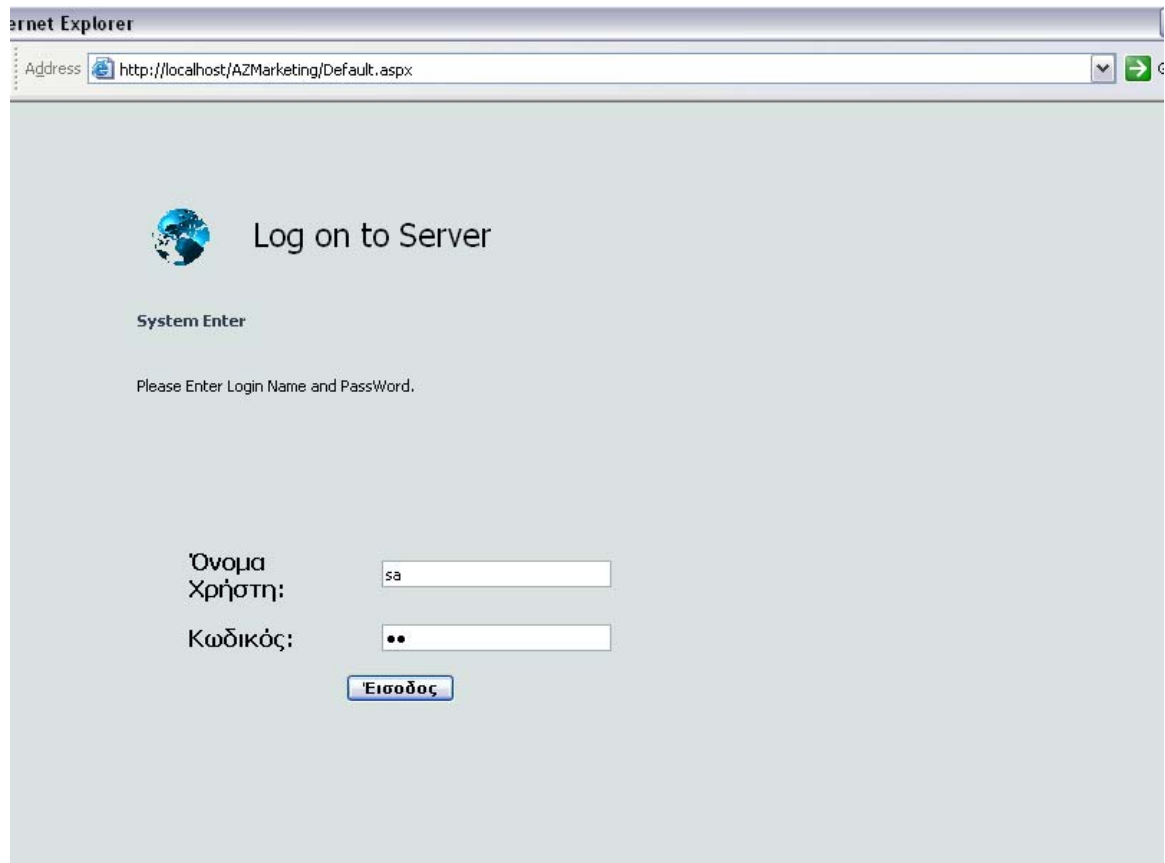
Πίνακας AnakoinosiTypeID, AnakoinosiType



Εικόνα 2. Το διάγραμμα της βάσης δεδομένων

3.2 Ανάπτυξη συστήματος

Παρακάτω περιγράφουμε τις οθόνες για την καταχώρηση, ανανέωση και διαγραφή ανακοινώσεων, την οθόνη που παρουσιάζει όλες τις εγγραφές του πίνακα, τον κώδικα που χρησιμοποιήσαμε αλλά και τις αποθηκευμένες διαδικασίες για τις παραπάνω διεργασίες.

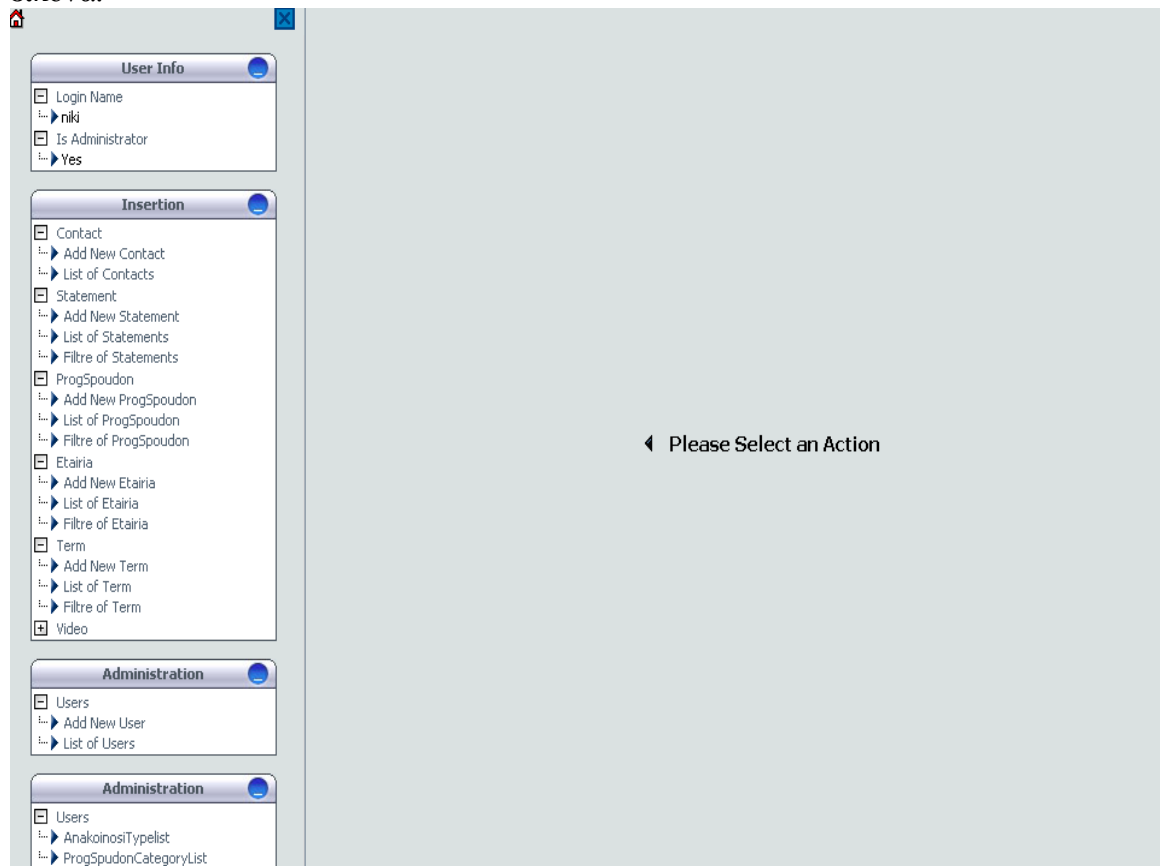


Εικόνα 3. Η οθόνη για την είσοδο στο σύστημα

Για κάθε χρήστη υπάρχει το userID το οποίο είναι πρωτεύον κλειδί στον πίνακα όπου καταχωρούνται οι χρήστες, και μοναδικό για κάθε χρήστη. Στην αρχική σελίδα απαιτείται ο χρήστης να πληκτρολογήσει το username και το password για να αποκτήσει πρόσβαση στο σύστημα. Δίνοντας ο χρήστης το username και το password γίνεται η σύνδεση με τη βάση και ελέγχεται αν ο κωδικός αντιστοιχεί σε κάποια από τις τιμές του πίνακα των χρηστών. Εάν είναι έγκυρες οι τιμές που πληκτρολογεί ο χρήστης τότε πηγαίνει στην αρχική σελίδα, ειδάλτως εμφανίζεται μήνυμα λάθους.

Αν η εισαγωγή αντιστοιχεί σε εγγραφή στον πίνακα χρήστες της βάσης δεδομένων τότε ανοίγει η σελίδα για την επιλογή διεργασίας. Ακόμα ελέγχεται αν ο χρήστης είναι διαχειριστής ώστε να μπορέσουμε παρακάτω να του δώσουμε τα ανάλογα δικαιώματα, ή αν δεν είναι, αντίστοιχα να τον περιορίσουμε σε συγκεκριμένες διεργασίες τις οποίες θα μπορεί να εκτελέσει..

Αφού ο χρήστης εισάγει έγκυρο username και password τότε μεταφέρεται στην σελίδα επιλογής διεργασίας η οποία φαίνεται στην επόμενη εικόνα.



Εικόνα 4. Η οθόνη επιλογής διεργασίας

Η αρχική μας σελίδα χωρίζεται σε δυο frames, το αριστερό όπου εμφανίζονται οι οντότητες και τα υπομενού και το δεξιό frame, όπου εμφανίζονται τα αποτελέσματα των επιλογών μας.

Όπως μπορούμε να δούμε στο αριστερό frame υπάρχει η δυνατότητα να δούμε πληροφορίες για το χρήστη, υπάρχουν οι οντότητες, η δυνατότητα καταχώρησης νέου



χρήστη αλλά και η δυνατότητα καταχώρησης νέου τύπου ανακοινώσεων, οι οποίες θα μπορούν να επιλέγουν από το αντίστοιχο πεδίο της οντότητας ανακοινώσεων, και τέλος η κατηγορία προγράμματος σπουδών, την οποία θα μπορούμε να επιλέξουμε από το αντίστοιχο πεδίο της οντότητας προγράμματος σπουδών.

Θα επιλέξουμε την διεργασία καταχώρηση ανακοίνωσης. Επιλέγοντας την διεργασία στο δεξιό frame θα ανοίξει η σελίδα Statement.aspx WF_Activity=INSERT. Δίνοντας το query string WF_Activity κάνουμε την εφαρμογή δυναμική, αφού χρησιμοποιούμε την ίδια σελίδα σε πολλές διεργασίες τροποποιώντας την ανάλογα με την διεργασία.

Παρακάτω θα δούμε τον κώδικα που έχουμε χρησιμοποιήσει όταν φορτώνεται η σελίδα.

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

    If Not IsPostBack Then

        Dim ds As New DataSet
        Dim constr As String
        Dim Conn As SqlConnection
        constr =
ConfigurationSettings.AppSettings("ConnectionString")
        Conn = New SqlConnection(constr)
        Dim Command As SqlDataAdapter

        Dim WF_Activity As String
        Dim ID As Integer

        WF_Activity = Request.QueryString("WF_Activity")
        BindAnakoinosiTypeID()

        If (WF_Activity = "UPDATE" Or WF_Activity = "DELETE") Then
            ID = Request.QueryString("ID")
            Command = New SqlDataAdapter("select * from Anakoinosi
where AnakoinosiID=" & ID, Conn)

            Conn.Open()

            Command.Fill(ds, "Anakoinosi")

            ddAnakoinosiTypeID.SelectedValue =
ds.Tables("Anakoinosi").Rows(0).Item("Type")

            txtTitle.Text =
ds.Tables("Anakoinosi").Rows(0).Item("Title")
            txtShortDescription.Text =
ds.Tables("Anakoinosi").Rows(0).Item("ShortDescription")
            txtLongDescription.Text =
ds.Tables("Anakoinosi").Rows(0).Item("LongDescription")
            txtReferenceUrl.Text =
ds.Tables("Anakoinosi").Rows(0).Item("ReferenceUrl")
```



```
txtPublisher.Text =
ds.Tables("Anakoinosi").Rows(0).Item("Publisher")
txtImerominiaKataxorisiss.Text =
ds.Tables("Anakoinosi").Rows(0).Item("ImerominiaKataxorisiss")
ddProtereotita.SelectedValue =
ds.Tables("Anakoinosi").Rows(0).Item("Protereotita")
drpStatus.SelectedValue =
ds.Tables("Anakoinosi").Rows(0).Item("Status")

If ds.Tables("Anakoinosi").Rows(0).Item("ReferenceFile")
Is System.DBNull.Value Then
Else
hpReferenceFile.NavigateUrl =
String.Format("javascript:var
ret=window.open('gw/getFile.aspx?fileid={0}', 'wcGetFile',
'width=500,height=500,menubar=0,toolbar=0,status=0,location=0,resizable=1
,scrollbars=1');",
CType(ds.Tables("Anakoinosi").Rows(0).Item("ReferenceFile"), String))
End If

Conn.Close()

End If

If WF_Activity = "UPDATE" Then

btnKataxorisi.Visible = False
btndiagrafi.Visible = False
lblWrongDate.Visible = False

End If

If WF_Activity = "DELETE" Then

btnKataxorisi.Visible = False
btnAnaneosi.Visible = False
Attachment.Visible = False

ddAnakoinosiTypeID.Visible = False
txtTitle.Enabled = False
txtShortDescription.Enabled = False
txtLongDescription.Enabled = False
txtReferenceUrl.Enabled = False
Attachment.Visible = False
txtPublisher.Enabled = False
txtImerominiaKataxorisiss.Enabled = False
ddProtereotita.Enabled = False
drpStatus.Enabled = False
lblWrongDate.Visible = False

End If

If (WF_Activity = "INSERT") Then
hpReferenceFile.Visible = False
btnAnaneosi.Visible = False
btndiagrafi.Visible = False
```



```
hpReferenceFile.Visible = False  
lblWrongDate.Visible = False
```

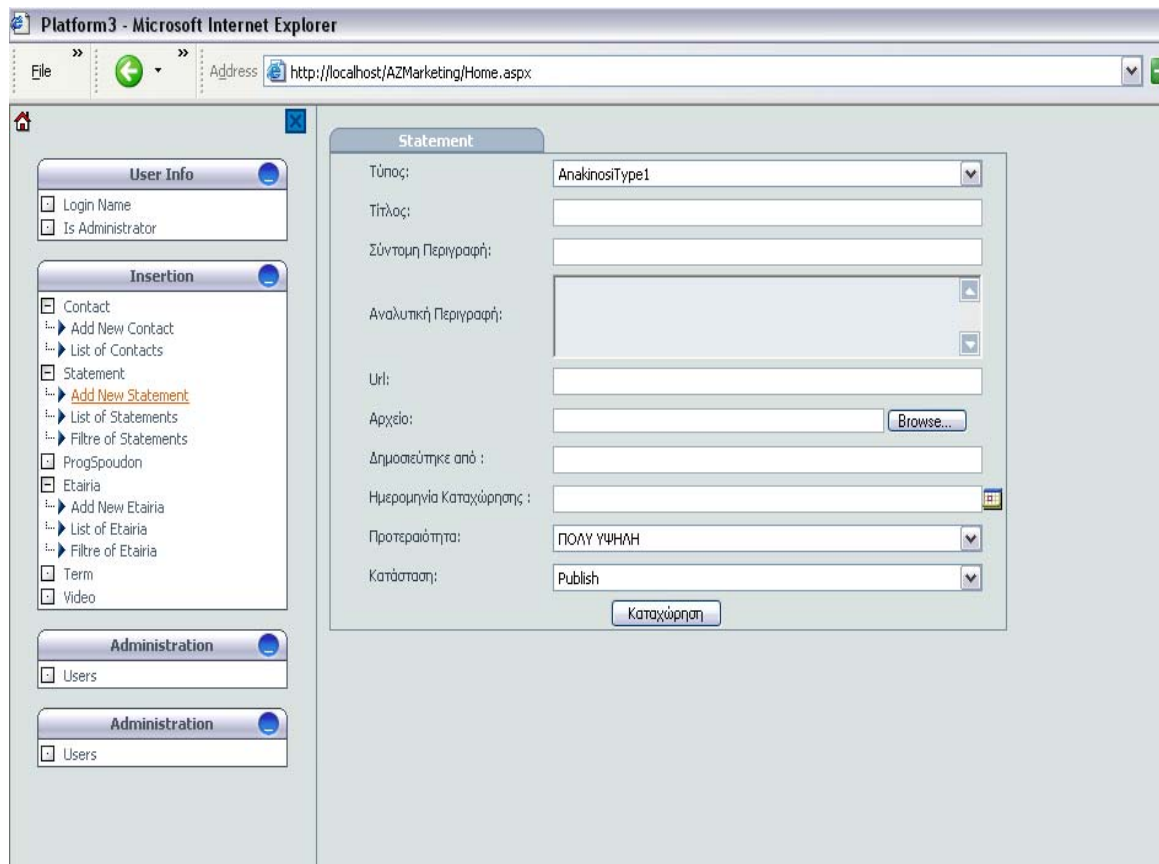
```
End If
```

```
End If
```

```
End Sub
```

Αρχικά γίνεται έλεγχος αν φορτώνεται η σελίδα για πρώτη φορά και αν αυτό ισχύει τότε δίνονται οι κατάλληλες τιμές στα διάφορα πεδία της σελίδας. Γίνεται η σύνδεση με τον SQL Server μέσω του Connection String το οποίο έχουμε αρχικοποιήσει στο αρχείο web.config. Όπως φαίνεται γίνεται έλεγχος του WF_Activity και ανάλογα με την τιμή του εμφανίζονται ή κρύβονται κάποια στοιχεία της σελίδας.

Στην παρακάτω σελίδα ο χρήστης μπορεί να καταχωρήσει τις τιμές που επιθυμεί και να φορτώσει κάποιο αρχείο και πατώντας το κουμπί της καταχώρησης θα δημιουργηθεί μια νέα εγγραφή στον πίνακα Ανακοινώσεις. Παρακάτω θα δούμε τον κώδικα που χρησιμοποιήσαμε για την καταχώρηση.



Εικόνα 5. Η οθόνη για την καταχώρηση Ανακοίνωσης



Ας δούμε τι συμβαίνει όταν ο χρήστης επιλέξει το κουμπί της καταχώρησης.

```
Private Sub btnKataxorisi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnKataxorisi.Click
```

```
    Dim constr As String
    Dim temp As Integer
    temp = 0
    Dim MyCommand As SqlCommand
    Dim Conn As SqlConnection

    If Page.IsValid Then

        constr =
ConfigurationSettings.AppSettings("ConnectionString")
        Conn = New SqlConnection(constr)
        MyCommand = New SqlCommand("StatementsInsert", Conn)
        MyCommand.CommandType = CommandType.StoredProcedure

        Dim Param01 As New SqlParameter("@ddAnakoinosiTypeID",
SqlDbType.Int)
        Dim Param02 As New SqlParameter("@Title", SqlDbType.NVarChar,
255)
        Dim Param03 As New SqlParameter("@ShortDescription",
SqlDbType.NVarChar, 500)
        Dim Param04 As New SqlParameter("@LongDescription",
SqlDbType.NVarChar, 2000)
        Dim Param05 As New SqlParameter("@ReferenceUrl",
SqlDbType.NVarChar, 255)
        Dim Param06 As New SqlParameter("@Publisher",
SqlDbType.NVarChar, 100)
        Dim Param07 As New SqlParameter("@ImerominiaKataxorisis",
SqlDbType.DateTime)
        Dim Param08 As New SqlParameter("@Status", SqlDbType.Int)
        Dim Param09 As New SqlParameter("@Prottereotita",
SqlDbType.Int)
        Dim Param10 As New SqlParameter("@ReferenceFile",
SqlDbType.Image)
        Dim Param11 As New
SqlParameter("@ReferenceFileFileTypeNumber", SqlDbType.Int)
        Dim Param12 As New SqlParameter("@ReferenceFileFileName",
SqlDbType.VarChar)

        Param01.Value = CInt(ddAnakoinosiTypeID.SelectedItem.Value)
        Param02.Value = txtTitle.Text
        Param03.Value = txtShortDescription.Text
        Param04.Value = txtLongDescription.Text
        Param05.Value = txtReferenceUrl.Text
        Param06.Value = txtPublisher.Text
        Try
            Param07.Value =
DateTime.Parse(txtImerominiaKataxorisis.Text, New
Globalization.CultureInfo("en-NZ", True))
```




```
        Catch m As Exception
            lblWrongDate.Visible = True
            lblWrongDate.Text = "Εισάγετε ημερομηνία της μορφής
DD/MM/YYYY!"
        Return
    End Try

    Param08.Value = CInt(drpStatus.SelectedValue)
    Param09.Value = CInt(ddProtereotita.SelectedItem.Value)

    If (Attachment.PostedFile.ContentLength <> 0) Then
        Param10.Value = upLoadFile(Attachment.PostedFile)

        Dim extension As String =
System.IO.Path.GetExtension(Attachment.PostedFile.FileName).ToUpper()
        extension = extension.Replace(".", "")
        Dim contenttype As String =
Attachment.PostedFile.ContentType
        Select Case extension
            Case "XLS"
                Param11.Value = CType(FileType.Excel, Integer)
            Case "DOC"
                Param11.Value = CType(FileType.Word, Integer)
            Case "TIF"
                Param11.Value = CType(FileType.Tiff, Integer)
            Case "JPG"
                Param11.Value = CType(FileType.Jpeg, Integer)
            Case "JPEG"
                Param11.Value = CType(FileType.Jpeg, Integer)
            Case "GIF"
                Param11.Value = CType(FileType.Gif, Integer)
            Case "TXT"
                Param11.Value = CType(FileType.Text, Integer)
            Case "PDF"
                Param11.Value = CType(FileType.Pdf, Integer)
            Case "BMP"
                Param11.Value = CType(FileType.Bitmap, Integer)
            Case "ZIP"
                Param11.Value = CType(FileType.Zip, Integer)
            Case "PPT"
                Param11.Value = CType(FileType.PowerPoint,
Integer)
            Case Else
                Param11.Value = CType(FileType.Default, Integer)
        End Select
        Param12.Value =
System.IO.Path.GetFileName(Attachment.PostedFile.FileName)
    Else
        Param10.Value = System.DBNull.Value
        Param11.Value = System.DBNull.Value
        Param12.Value = System.DBNull.Value
    End If

    With MyCommand.Parameters
        .Add(Param01)
        .Add(Param02)
        .Add(Param03)
```



```
.Add (Param04)
.Add (Param05)
.Add (Param06)
.Add (Param07)
.Add (Param08)
.Add (Param09)
.Add (Param10)
.Add (Param11)
.Add (Param12)

End With

MyCommand.Connection.Open ()
Dim sqlReader As SqlDataReader
sqlReader = MyCommand.ExecuteReader ()
Try
    If sqlReader.Read () Then
        temp = sqlReader (0).ToString ()
    End If
Catch m As Exception
    Response.Write (m.ToString ())
End Try
sqlReader.Close ()
MyCommand.Connection.Close ()
If temp <> 0 Then
    Response.Redirect ("success.aspx")
Else
    Response.Redirect ("failed.aspx")
End If
End If

End Sub
```

Αρχικά γίνεται έλεγχος για το αν η σελίδα είναι έγκυρη δηλαδή αν έχουν συμπληρωθεί τα απαραίτητα πεδία. Αν είναι έγκυρη τότε περνάει τις τιμές των πεδίων σε παραμέτρους τις οποίες στέλνει στη βάση δεδομένων η οποία με την σειρά της εκτελεί μια αποθηκευμένη διαδικασία (stored procedure) τη StatementsInsert για την εισαγωγή μίας νέας εγγραφής ανακοίνωσης με τα στοιχεία της φόρμας. Η σύνδεση με τον SQL Server γίνεται μέσω του Connection String το οποίο έχουμε αρχικοποιήσει στο αρχείο web.config ώστε αν το αλλάξουμε σε αυτό το αρχείο να ενημερώνονται όλες οι σελίδες της εφαρμογής. Όπως μπορούμε να δούμε γίνεται και ο έλεγχος για την ημερομηνία που καταχωρεί ο χρήστης καθώς και τον τύπο του αρχείου που θα ανεβάσουμε μέσω της επέκτασης του αρχείου. Αν είναι επιτυχής η καταχώρηση τότε εμφανίζεται η σελίδα success.aspx αλλιώς αν δεν είναι επιτυχής η καταχώρηση εμφανίζεται η σελίδα failed.aspx



Παρακάτω θα δούμε την αποθηκευμένη διαδικασία που χρησιμοποιούμε για την καταχώρηση των ανακοινώσεων

```
ALTER PROCEDURE StatementsInsert

(

    @ddAnakoinosiTypeID int,@Title nvarchar(255),@ShortDescription
nvarchar(500),@LongDescription nvarchar(2000), @ReferenceURL
nvarchar(255),@ReferenceFile image,@Publisher
nvarchar(100),@ImerominiaKataxis datetime,@Protireotita int,@Status int,
@ReferenceFileFileNumber int,@ReferenceFileFileName varchar(100)

)

AS

if @ReferenceFile is not null

begin

insert into dbo.[File] (BinaryFile,FileTypeNumber,[FileName]) values
(@ReferenceFile,@ReferenceFileFileNumber,@ReferenceFileFileName)

declare @ReferenceFileFileId as int

select @ReferenceFileFileId=@@identity

end

if @ddAnakoinosiTypeID=0

begin

set @ddAnakoinosiTypeID = null

end

INSERT INTO dbo.Anakoinosi
```



```
(  
    Type,Title,ShortDescription,LongDescription,ReferenceURL,ReferenceFile,  
    Publisher,ImerominiaKataxoris,Protereotita,Status  
)  
  
VALUES  
  
(  
    @ddAnakoinosiTypeID,@Title,@ShortDescription,  
    @LongDescription,@ReferenceURL,@ReferenceFileFileId,@Publisher,@Imero  
    miniaKataxoris,@Protereotita,@Status  
)  
  
SELECT @@IDENTITY
```

Στην παρακάτω εικόνα βλέπουμε την σελίδα της ανανέωσης όπου ο χρήστης μπορεί να αλλάξει τις τιμές των πεδίων καθώς επίσης και να αλλάξει αρχεία τα οποία καταχωρούνται εκ νέου στη βάση δεδομένων.

Εικόνα 6. Η οθόνη για την ανανέωση Ανακοίνωσης

Ο κώδικας που χρησιμοποιήσαμε για την διεργασία της ανανέωσης είναι όμοιος με τον κώδικα της καταχώρησης με τη διαφορά ότι στις παραμέτρους που δίνουμε στον SQL Server δίνουμε και το ID της εγγραφής και εκτελείται η αποθηκευμένη διαδικασία StatementsUpdate.

```
Private Sub btnAnaneosi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnAnaneosi.Click
```

```
Dim constr As String  
Dim temp As Integer  
temp = 0  
Dim MyCommand As SqlCommand
```



```
Dim Conn As SqlConnection

If Page.IsValid Then

    constr =
ConfigurationSettings.AppSettings("ConnectionString")
    Conn = New SqlConnection(constr)
    MyCommand = New SqlCommand("StatementsUpdate", Conn)
    MyCommand.CommandType = CommandType.StoredProcedure

    Dim Param00 As New SqlParameter("@AnakoinosiID",
SqlDbType.Int)
    Dim Param01 As New SqlParameter("@ddAnakoinosiTypeID",
SqlDbType.Int)
    Dim Param02 As New SqlParameter("@Title", SqlDbType.NVarChar,
255)
    Dim Param03 As New SqlParameter("@ShortDescription",
SqlDbType.NVarChar, 500)
    Dim Param04 As New SqlParameter("@LongDescription",
SqlDbType.NVarChar, 2000)
    Dim Param05 As New SqlParameter("@ReferenceUrl",
SqlDbType.NVarChar, 255)
    Dim Param06 As New SqlParameter("@Publisher",
SqlDbType.NVarChar, 100)
    Dim Param07 As New SqlParameter("@ImerominiaKataxoris",
SqlDbType.DateTime)
    Dim Param08 As New SqlParameter("@Status", SqlDbType.Int)
    Dim Param09 As New SqlParameter("@Protereotita",
SqlDbType.Int)
    Dim Param10 As New SqlParameter("@ReferenceFile",
SqlDbType.Image)
    Dim Param11 As New
SqlParameter("@ReferenceFileFileTypeNumber", SqlDbType.Int)
    Dim Param12 As New SqlParameter("@ReferenceFileFileName",
SqlDbType.VarChar)

    Param00.Value = Request("ID")
    Param01.Value = CInt(ddAnakoinosiTypeID.SelectedItem.Value)
    Param02.Value = txtTitle.Text
    Param03.Value = txtShortDescription.Text
    Param04.Value = txtLongDescription.Text
    Param05.Value = txtReferenceUrl.Text
    Param06.Value = txtPublisher.Text
    Try
        Param07.Value =
DateTime.Parse(txtImerominiaKataxoris.Text, New
Globalization.CultureInfo("en-NZ", True))
    Catch m As Exception
        lblWrongDate.Visible = True
        lblWrongDate.Text = "Εισάγετε ημερομηνία της μορφής
DD/MM/YYYY!"
    Return
End Try
```



```
Param08.Value = CInt(drpStatus.SelectedValue)
Param09.Value = CInt(ddProtereotita.SelectedItem.Value)

If (Attachment.PostedFile.ContentLength <> 0) Then
    Param10.Value = upLoadFile(Attachment.PostedFile)

    Dim extension As String =
System.IO.Path.GetExtension(Attachment.PostedFile.FileName).ToUpper()
    extension = extension.Replace(".", "")
    Dim contenttype As String =
Attachment.PostedFile.ContentType
    Select Case extension
        Case "XLS"
            Param11.Value = CType(FileType.Excel, Integer)
        Case "DOC"
            Param11.Value = CType(FileType.Word, Integer)
        Case "TIF"
            Param11.Value = CType(FileType.Tiff, Integer)
        Case "JPG"
            Param11.Value = CType(FileType.Jpeg, Integer)
        Case "JPEG"
            Param11.Value = CType(FileType.Jpeg, Integer)
        Case "GIF"
            Param11.Value = CType(FileType.Gif, Integer)
        Case "TXT"
            Param11.Value = CType(FileType.Text, Integer)
        Case "PDF"
            Param11.Value = CType(FileType.Pdf, Integer)
        Case "BMP"
            Param11.Value = CType(FileType.Bitmap, Integer)
        Case "ZIP"
            Param11.Value = CType(FileType.Zip, Integer)
        Case "PPT"
            Param11.Value = CType(FileType.PowerPoint,
Integer)

        Case Else
            Param11.Value = CType(FileType.Default, Integer)
    End Select
    Param12.Value =
System.IO.Path.GetFileName(Attachment.PostedFile.FileName)
Else
    Param10.Value = System.DBNull.Value
    Param11.Value = System.DBNull.Value
    Param12.Value = System.DBNull.Value
End If

With MyCommand.Parameters
    .Add(Param00)
    .Add(Param01)
    .Add(Param02)
    .Add(Param03)
    .Add(Param04)
    .Add(Param05)
    .Add(Param06)
    .Add(Param07)
```



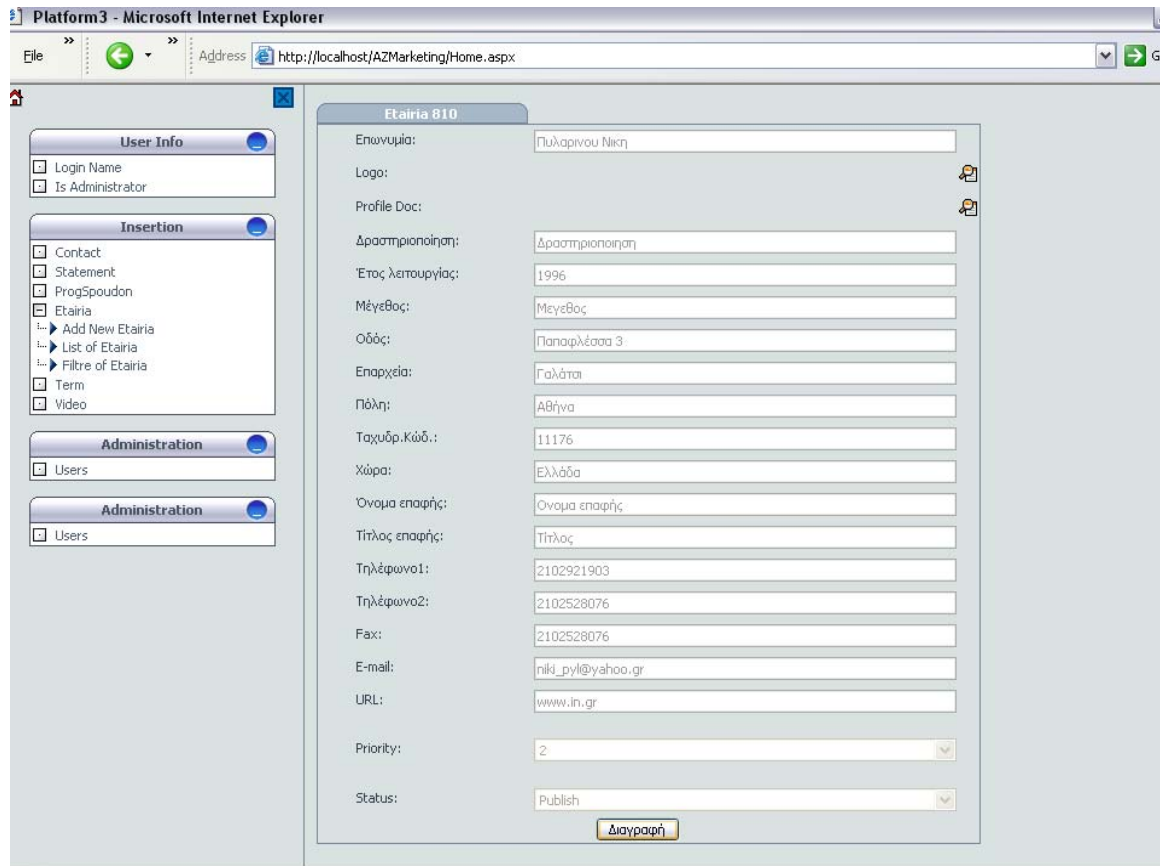
```
.Add (Param08)
.Add (Param09)
.Add (Param10)
.Add (Param11)
.Add (Param12)

End With

MyCommand.Connection.Open ()
Dim sqlReader As SqlDataReader
sqlReader = MyCommand.ExecuteReader ()
Try
    If sqlReader.Read () Then
        temp = sqlReader (0).ToString ()
    End If
Catch m As Exception
    Response.Write (m.ToString ())
End Try
sqlReader.Close ()
MyCommand.Connection.Close ()
If temp <> 0 Then
    Response.Redirect ("success.aspx")
Else
    Response.Redirect ("failed.aspx")
End If
End If

End Sub
```


Εδώ μπορούμε να δούμε τη σελίδα της διαγραφής, όπου ο χρήστης μπορεί να διαγράψει μια συγκεκριμένη εγγραφή από τη βάση δεδομένων.



Εικόνα 7. Η οθόνη για την διαγραφή Ανακοίνωσης

Στον παρακάτω κώδικα που αφορά στη διαγραφή των εγγραφών η μόνη παράμετρος που δίνουμε στον SQL Server είναι το ID της εγγραφής διότι χρειάζεται να δηλώσουμε μόνο το ID της εγγραφής που επιθυμούμε να διαγράψουμε και εκτελείται η αποθηκευμένη διαδικασία StatementsDelete

```
Private Sub btndiagrafi_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndiagrafi.Click
```



```
Dim constr As String
Dim temp As Integer
temp = 0
Dim MyCommand As SqlCommand
Dim Conn As SqlConnection

If Page.IsValid Then

    constr =
ConfigurationSettings.AppSettings("ConnectionString")
    Conn = New SqlConnection(constr)

    MyCommand = New SqlCommand("StatementsDelete", Conn)
    MyCommand.CommandType = CommandType.StoredProcedure

    Dim Param00 As New SqlParameter("@AnakoinosiID",
SqlDbType.Int)
    Param00.Value = Request("ID")

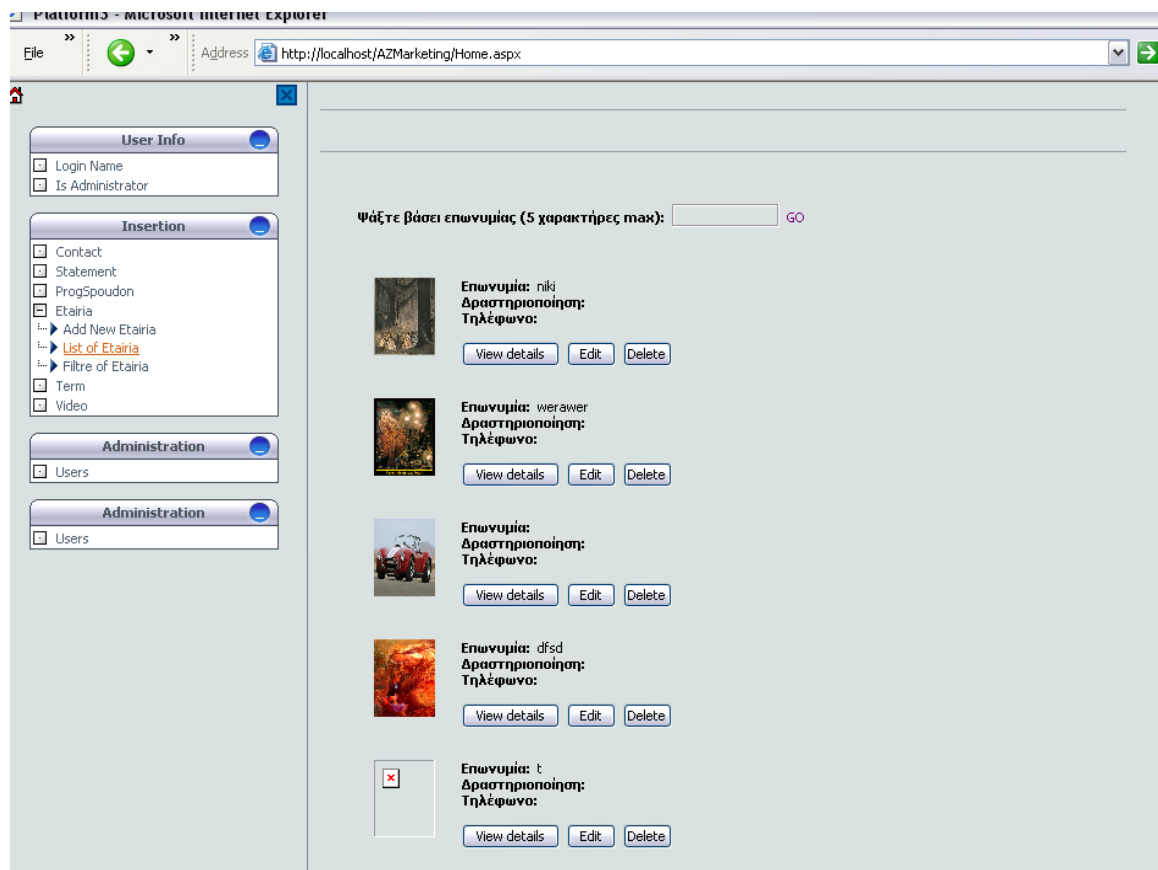
    With MyCommand.Parameters
        .Add(Param00)

    End With

    MyCommand.Connection.Open()
    Dim sqlReader As SqlDataReader
    sqlReader = MyCommand.ExecuteReader()
    Try
        If sqlReader.Read() Then
            temp = sqlReader(0).ToString()
        End If
    Catch m As Exception
        Response.Write(m.ToString())
    End Try
    sqlReader.Close()
    MyCommand.Connection.Close()
    If temp <> 0 Then
        Response.Redirect("success.aspx")
    Else
        Response.Redirect("failed.aspx")
    End If

End If
```

Στην παρακάτω εικόνα βλέπουμε τη σελίδα των εγγραφών που υπάρχουν στον πίνακα των ανακοινώσεων. Για την προβολή και επεξεργασία των δεδομένων μας που επιστρέφονται από τη βάση δεδομένων έχουμε χρησιμοποιήσει 2 στοιχεία ελέγχου Web φόρμας, τα στοιχεία ελέγχου DataGrid και DataList. Εδώ βλέπουμε την αναπαράσταση ενός DataList. Σε αυτή τη σελίδα έχουμε τη δυνατότητα της αναζήτησης κάποια εγγραφής. Από εδώ μπορούμε να επιλέξουμε να δούμε την αναλυτική παρουσίαση της κάθε εγγραφής, να κάνουμε edit όπου μπορούμε να δούμε τις καταχωρήσεις του χρήστη στη συγκεκριμένη εγγραφή και να αλλάξουμε τα στοιχεία, και τέλος να διαγράψουμε την κάθε εγγραφή.



Εικόνα 8. Η οθόνη για την λίστα Ανακοίνωσης



Στην παρακάτω εικόνα βλέπουμε την αναλυτική παρουσίαση μιας συγκεκριμένης εγγραφής, όπου έχουμε τη δυνατότητα να δούμε τις καταχωρήσεις του χρήστη αλλά και να ανοίξουμε το αρχείο που έχει αποθηκεύσει ο χρήστης.

Platform3 - Microsoft Internet Explorer
Address: http://localhost/AZMarketing/Home.aspx

User Info
 Login Name
 Is Administrator

Insertion
 Contact
 Statement
 ProgSpoudon
 Etairia
 ▶ Add New Etairia
 ▶ List of Etairia
 ▶ Filtr of Etairia
 Term
 Video

Administration
 Users

Administration
 Users

Ψάξτε βάσει επωνυμίας (5 χαρακτήρες max): GO

Επωνυμία: Πυλαρινού Νίκη
Δραστηριοποίηση: Δραστηριοποίηση
Τηλέφωνο: 2102921903
View details Edit Delete

Επωνυμία:
Δραστηριοποίηση:
Τηλέφωνο:
View details Edit Delete

Επωνυμία:
Δραστηριοποίηση:
Τηλέφωνο:
View details Edit Delete

Επωνυμία:
Δραστηριοποίηση:
Τηλέφωνο:
View details Edit Delete

Επωνυμία:
Δραστηριοποίηση:
Τηλέφωνο:
View details Edit Delete

Πυλαρινού Νίκη
Δραστηριοποίηση: Δραστηριοποίηση
Έτος λειτουργίας: 1996
Μέγεθος: Μεγέθος
Οδός: Παπαφλέσσα 3
Επαρχία: Γαλάτσι
Πόλη: Αθήνα
Ταχ. κώδικας: 11176
Χώρα: Ελλάδα
Όνομα επαφής:
Τίτλος επαφής:
Τηλέφωνο1: 2102921903
Τηλέφωνο2: 2102528076
Fax: 2102528076
E-mail: niki_pyl@yahoo.gr
URL: www.in.gr
Profile Doc:
Priority: 2
Status: Publish

Εικόνα 9. Η οθόνη για την λίστα Ανακοίνωσης



Παρακάτω θα δούμε τον κώδικα που χρησιμοποιήσαμε αναλυτική παρουσίαση των εγγραφών μας.

Στην Page_Load καλούμε την search χωρίς να δώσουμε κάποια τιμή ώστε να μας επιστρέψει όλα τα δεδομένα. Χρησιμοποιούμε ένα session που κρατά την τιμή του pageIndex το οποίο κρατά την τιμή της σελίδας στην οποία βρισκόμαστε και του δίνουμε αρχική τιμή μηδέν. Πηγαίνοντας στην επόμενη σελίδα το pageIndex αυξάνει κατά ένα ενώ πηγαίνοντας στην προηγούμενη σελίδα αντίστοιχα μειώνεται κατά ένα. Στην υπορουτίνα MyList_ItemCommand έχουμε δηλώσει στο CommandName τις παραμέτρους της διεργασία που θα εκτελέσουμε όπως edit για να ανανεώσουμε τα στοιχεία μιας εγγραφής, delete για να διαγράψουμε την εγγραφή της οποίας το id έχουμε επιλέξει, view όπου θα μπορούμε να δούμε αναλυτικά την εγγραφή που έχουμε επιλέξει, καθώς και τη σελίδα που θα οδηγούμαστε μέσω του url στο οποίο δηλώνουμε την σελίδα που θα μεταβούμε και το query string για την διεργασία που θα εκτελεστεί. Επίσης βλέπουμε την υπορουτίνα Search Click με τη βοήθεια της οποίας μπορούμε να αναζητήσουμε μια εγγραφή από τη βάση μας δίνοντας σαν κριτήριο τον τίτλο της ανακοίνωσης. Τέλος υπάρχει και η υπορουτίνα ClearClick όπου πατώντας το κουμπί clear μας πηγαίνει στη σελίδα StatementsListDL δηλαδή στην πρώτη σελίδα της αναλυτικής λίστας των ανακοινώσεων.

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    txtSearch.Attributes.Add("onKeyPress", "javascript:if
(event.keyCode == 13) __doPostBack('" + btnSearch.UniqueID + "', '')")
    If Not IsPostBack Then
        Session("PageIndex") = 0
        SearchClick(sender, e)
    End If
End Sub

Public Sub Prev_Click(ByVal obj As Object, ByVal e As EventArgs)
    Session("PageIndex") = Session("PageIndex") - 1
    SearchClick(Nothing, Nothing)
End Sub

Public Sub Next_Click(ByVal obj As Object, ByVal e As EventArgs)
    Session("PageIndex") = Session("PageIndex") + 1
    SearchClick(Nothing, Nothing)
End Sub

Private Sub MyList_ItemCommand(ByVal source As Object, ByVal e As
DataListCommandEventArgs) Handles myList.ItemCommand

    Dim primaryKey As String
    Dim URL As String
```



```
primaryKey = CStr(e.CommandArgument)

If e.CommandName = "Edit" Then
    txtSearch.Text = ""
    URL = "StatementsDL.aspx?WF_Activity=UPDATE&ID=" + primaryKey
    Response.Redirect(URL)
End If

If e.CommandName = "Delete" Then
    txtSearch.Text = ""
    URL = "StatementsDL.aspx?WF_Activity=DELETE&ID=" + primaryKey
    Response.Redirect(URL)
End If

If e.CommandName = "View" Then
    txtSearch.Text = ""

    Dim Conn As String
    Dim MyCommand As String
    Dim MyDataSet As DataSet
    Dim MyAdapter As SqlDataAdapter

    Conn = ConfigurationSettings.AppSettings("ConnectionString")
    MyCommand = "SELECT * FROM Anakoinosi WHERE AnakoinosiID = "
+ primaryKey

    MyAdapter = New SqlDataAdapter(MyCommand, Conn)
    MyDataSet = New DataSet
    MyAdapter.Fill(MyDataSet)

    myDetailList.DataSource = MyDataSet
    myDetailList.DataBind()

    myList.SelectedIndex = e.Item.ItemIndex
End If

End Sub

Function FormatURL(ByVal strArgument1) As String

    Return ("gw/getFile.aspx?fileid=" & strArgument1)

End Function

Public Sub SearchClick(ByVal sender As Object, ByVal e As EventArgs)

    Dim constr As String
    Dim Conn As SqlConnection
    Dim MyCommand As String
    Dim MyDataSet As New DataSet
    Dim MyAdapter As SqlDataAdapter

    myDetailList.DataSource = Nothing
    myDetailList.DataBind()

    constr = ConfigurationSettings.AppSettings("ConnectionString")
```



```
If txtSearch.Text = "" Then

    btnSearch.Visible = True
    btnClear.Visible = False
    txtSearch.Enabled = True

    MyCommand = "SELECT * FROM Anakoinosi"
    Conn = New SqlConnection(constr)
    MyAdapter = New SqlDataAdapter(MyCommand, Conn)

    MyAdapter.Fill(MyDataSet, "Anakoinosi")

    pagedData.DataSource =
MyDataSet.Tables("Anakoinosi").DefaultView()
    pagedData.AllowPaging = True
    pagedData.PageSize = 6

    Try

        pagedData.CurrentPageIndex =
Int32.Parse(CStr(Session("PageIndex")))
    Catch ex As Exception
        Session("PageIndex") = 0
        pagedData.CurrentPageIndex = 0
    End Try

    btnPrev.Visible = (Not pagedData.IsFirstPage)
    btnNext.Visible = (Not pagedData.IsLastPage)

    myList.DataSource = pagedData
    myList.DataBind()

    btnSearch.Visible = False
    btnClear.Visible = True
    txtSearch.Enabled = False

    MyCommand = "SELECT * FROM Anakoinosi WHERE Title LIKE '%" &
txtSearch.Text & "%'"
    Conn = New SqlConnection(constr)
    MyAdapter = New SqlDataAdapter(MyCommand, Conn)

    MyAdapter.Fill(MyDataSet, "Anakoinosi")

    pagedData.DataSource =
MyDataSet.Tables("Anakoinosi").DefaultView()
    pagedData.AllowPaging = True
    pagedData.PageSize = 6

    Try

        pagedData.CurrentPageIndex =
Int32.Parse(CStr(Session("PageIndex")))
    Catch ex As Exception
        Session("PageIndex") = 0
        pagedData.CurrentPageIndex = 0
    End Try
```



```
        btnPrev.Visible = (Not pagedData.IsFirstPage)
        btnNext.Visible = (Not pagedData.IsLastPage)

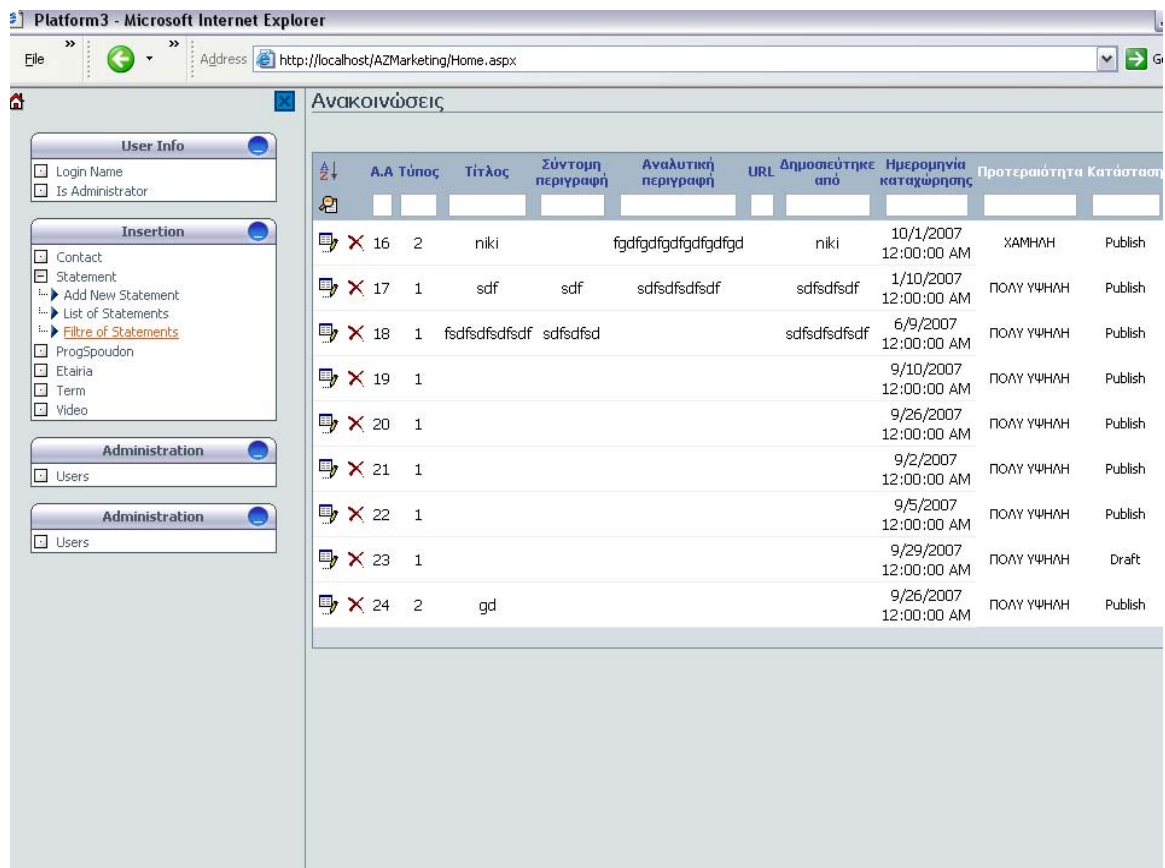
        myList.DataSource = pagedData
        myList.DataBind()

    End If

End Sub

Public Sub ClearClick(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    txtSearch.Text = ""
    Dim URL As String
    URL = "StatementsListDL.aspx?WF_Activity=UPDATE&ID=-1"
    Response.Redirect (URL)
End Sub

Function ViewFile(ByVal strArgument1) As String
    Try
        Return String.Format("var
ret=window.open('gw/getFile.aspx?fileid={0}', 'wcViewFile',
'width=500,height=500,menubar=0,toolbar=0,status=0,location=0,resizable=1
,scrollbars=1');", CStr(strArgument1))
    Catch m As Exception
        Return ""
    End Try
End Function
```

The screenshot shows a web browser window titled "Platform3 - Microsoft Internet Explorer" with the address "http://localhost/AZMarketing/Home.aspx". The page displays a section titled "Ανακοινώσεις" (Announcements) containing a DataGrid table. The table has columns for ID, Type, Title, Summary, Detailed Description, URL, Author, Registration Date, Priority, and Status. The table contains 9 rows of data, with the last row (ID 24) highlighted in blue.

ID	A.A	Τύπος	Τίτλος	Σύντομη περιγραφή	Αναλυτική περιγραφή	URL	Δημοσιεύτηκε από	Ημερομηνία καταχώρησης	Προτεραιότητα	Κατάσταση
16	2		niki		fgdfgdfgdfgdfgdfgd		niki	10/1/2007 12:00:00 AM	ΧΑΜΗΛΗ	Publish
17	1		sdf	sdf	sdfsdfsdfsdf		sdfsdfsdf	1/10/2007 12:00:00 AM	ΠΟΛΥ ΥΨΗΛΗ	Publish
18	1		fsd	fsd	fsd		fsd	6/9/2007 12:00:00 AM	ΠΟΛΥ ΥΨΗΛΗ	Publish
19	1							9/10/2007 12:00:00 AM	ΠΟΛΥ ΥΨΗΛΗ	Publish
20	1							9/26/2007 12:00:00 AM	ΠΟΛΥ ΥΨΗΛΗ	Publish
21	1							9/2/2007 12:00:00 AM	ΠΟΛΥ ΥΨΗΛΗ	Publish
22	1							9/5/2007 12:00:00 AM	ΠΟΛΥ ΥΨΗΛΗ	Publish
23	1							9/29/2007 12:00:00 AM	ΠΟΛΥ ΥΨΗΛΗ	Draft
24	2		gd					9/26/2007 12:00:00 AM	ΠΟΛΥ ΥΨΗΛΗ	Publish

Εικόνα 10. Η οθόνη για τη λίστα Ανακοινώσεων

Εδώ βλέπουμε την αναπαράσταση ενός DataGrid. Το στοιχείο ελέγχου DataGrid είναι ένα πλέγμα με πολλές στήλες δεδομένων που έχει πολλές επιπλέον λειτουργίες για τη διάταξη των περιεχομένων του, μαζί με λειτουργίες ταξινόμησης, επεξεργασίας, επιλογής και μετακίνησης στα δεδομένα. Στην παραπάνω οθόνη βλέπουμε όλες τις εγγραφές που υπάρχουν στον πίνακα ανακοινώσεων. Εδώ μπορούμε να ταξινομήσουμε τις εγγραφές των χρηστών αλφαβητικά, αλλά και να αναζητήσουμε κάποια συγκεκριμένη εγγραφή. Επίσης έχουμε τη δυνατότητα της ανανέωση κάποιας εγγραφής, αλλάζοντας τα δεδομένα που έχει καταχωρήσει ο χρήστης, και να διαγράψουμε κάποια εγγραφή. Μπορεί ακόμα ο χρήστης να αναζητήσει αποτελέσματα σχετικά με τα κριτήρια που θα εισάγει. Ακόμα μπορεί να δει τα αποτελέσματα ταξινομημένα κατά πεδίο επιλέγοντας αύξουσα ή φθίνουσα σειρά. Τέλος μπορούμε πατώντας τα κουμπιά στα αριστερά της λίστας να τροποποιήσουμε ή να διαγράψουμε μια εγγραφή.



Ας δούμε τον κώδικα που χρησιμοποιήσαμε για το φίλτρο των ανακοινώσεων.

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

    If Not Page.IsPostBack Then
        Session.Item("empstrAnakoinosiID") = Nothing
        Session.Item("empstrTitle") = Nothing
        Session.Item("empstrType") = Nothing
        Session.Item("empstrShortDescription") = Nothing
        Session.Item("empstrLongDescripton") = Nothing
        Session.Item("empstrReferenceURL") = Nothing
        Session.Item("empstrPublisher") = Nothing
        Session.Item("empstrImerominiaKataxoris") = Nothing
        Session.Item("empstrProttereotita") = Nothing
        Session.Item("empstrStatus") = Nothing

        Session("empmycom") = Nothing
        Session("empfilAnakoinosi") = Nothing
        BindGrid()
    End If

End Sub
```

Εδώ βλέπουμε τον κώδικα που ασχολείται με το paging:

```
Sub PageIndexChanged(ByVal sender As Object, ByVal e As EventArgs)

    Dim strCmdName As String = CType(sender, LinkButton).CommandName.ToUpper
    Dim strCmdArg As String = CType(sender, LinkButton).CommandArgument.ToUpper
    Dim dsfilter As New DataSet
    Dim mycom As String = CType(Session.Item("empmycom"), String)

    Select Case strCmdName
        Case "NEXT"
            Datagrid1.CurrentPageIndex += 1
        Case "PREV"
            Datagrid1.CurrentPageIndex -= 1
        Case "PAGE"
            Datagrid1.CurrentPageIndex = CType(strCmdArg, Integer) - 1
    End Select

    BindGrid()
End Sub
```

Ο κώδικας που ασχολείται με την ταξινόμηση είναι ο εξής:

```
Public Sub SortCommand(ByVal sender As Object, _
    ByVal e As System.Web.UI.WebControls.DataGridSortCommandEventArgs)

    viewstate.Add("sortfield", e.SortExpression)
```



```
If viewstate("sortdirection") Is Nothing Then
    viewstate.Add("sortdirection", "ASC")
Else
    If viewstate("sortdirection") = "ASC" Then
        viewstate("sortdirection") = "DESC"
    Else
        viewstate("sortdirection") = "ASC"
    End If
End If

BindGrid()

End Sub
```

Ο παραπάνω κώδικας ελέγχει με βάση ποια στήλη θα γίνει η ταξινόμηση και αν θα είναι αύξουσα ή φθίνουσα και ανάλογα ταξινομεί τα αποτελέσματα.

```
Sub BindGrid()
    If Session("empfilEtairia") Is Nothing Then
        ds = New DataSet
        Dim sCommand As String

        If viewstate("sortfield") Is Nothing Then
            sCommand = "select * from " & tablename
        Else
            sCommand = "select * from " & tablename & " order by " &
viewstate("sortfield") & " " & viewstate("sortdirection") & ""
        End If

        MyDataAdapter = New SqlDataAdapter(sCommand, MyConnection)
        MyDataAdapter.Fill(ds, tablename)
        Datagrid1.DataSource = ds
        Datagrid1.DataMember = tablename
        Datagrid1.DataBind()

    Else

        Dim mycom As String = CType(Session.Item("empmycom"), String)
        dsfilter = New DataSet
        Dim sCommand As String

        If viewstate("sortfield") Is Nothing Then
            sCommand = mycom
        Else
            sCommand = mycom + "order by " & viewstate("sortfield") &
" " & viewstate("sortdirection") & ""

        End If

        MyDataAdapter = New SqlDataAdapter(sCommand, MyConnection)
        MyDataAdapter.Fill(dsfilter, tablename)
        Datagrid1.DataSource = dsfilter
        Datagrid1.DataMember = tablename
        Datagrid1.DataBind()
    End If
End Sub
```



End Sub

Το παραπάνω κομμάτι κώδικα ελέγχει αν έχουμε επιλέξει κάποιο κριτήριο αναζήτησης και αν αυτό ισχύει τότε επιστρέφει απο τη βάση τις εγγραφές που ταιριάζουν στο κριτήριο. Αν δεν υπάρχει κάποιο πεδίο αναζήτησης συμπληρωμένο τότε επιστρέφει από τη βάση όλες τις εγγραφές. Παράλληλα ελέγχει αν ο χρήστης έχει επιλέξει να ταξινομήσει τα αποτελέσματα ως προς κάποια στήλη, και αν ναι, τότε τα ταξινομεί ανάλογα.

Ας δούμε και το κομμάτι του κώδικα που εκτελείται όταν ο χρήστης πατήσει το κουμπί για την αναζήτηση μεταξύ των αποτελεσμάτων

```
Private Function Create textbox(ByVal width) As TextBox
    Dim text1 As TextBox
    Dim n As Integer
    text1 = New TextBox
    text1.Width = width
    Return text1
End Function

Public Sub lista_ItemCreated(ByVal sender As Object, ByVal e As
System.Web.UI.WebControls.DataGridItemEventArgs) Handles
Datagrid1.ItemCreated

    If e.Item.ItemType = ListItemType.Footer Then

        Dim td As TableCell
        Dim u As Unit
        Dim text1 As TextBox
        filterbar = New DataGridItem(0, 0, ListItemType.Header)
        Dim col As String = CType(Session.Item("pedio"), String)
        Datagrid1.Controls(0).Controls.AddAt(2, filterbar)
        td = New TableCell
        filterbar.Controls.Add(td)
        go = New LinkButton
        go.Width =
Datagrid1.Columns(0).ItemStyle.Width.Percentage(100)
        go.Text = "<img src=images/Search.gif border=0>"
        AddHandler go.Click, (AddressOf okclick)
        td.Controls.Add(go)
        td = New TableCell
        filterbar.Controls.Add(td)

        For n As Integer = 2 To Datagrid1.Columns.Count - 1 Step 1

            u = Datagrid1.Columns(n).ItemStyle.Width.Percentage(90)
            td = New TableCell
            filterbar.Controls.Add(td)
            text1 = Createtextbox(u)
            td.Controls.Add(text1)
            If n = 2 Then
```



```
                text1.Text =
CType(Session.Item("empstrAnakoinosiID"), String)
            End If
            If n = 3 Then
                text1.Text =
CType(Session.Item("empstrTitle"),String)
            End If
            If n = 4 Then
                text1.Text = CType(Session.Item("empstrType"),String)
            End If
            If n = 5 Then
                text1.Text =
CType(Session.Item("empstrShortDescription"), String)
            End If
            If n = 6 Then
                text1.Text =
CType(Session.Item("empstrLongDescription"), String)
            End If
            If n = 7 Then
                text1.Text =
CType(Session.Item("empstrReferenceURL"), String)
            End If
            If n = 8 Then
                text1.Text =
CType(Session.Item("empstrPublisher"),String)
            End If
            If n = 9 Then
                text1.Text =
CType(Session.Item("empstrImeronomiaKataxoris"), String)
            End If
            If n = 10 Then
                text1.Text =
CType(Session.Item("empstrProtereotita"), String)
            End If
            If n = 11 Then
                text1.Text =
CType(Session.Item("empstrStatus"),String)
            End If
            If n = 12 Then
                text1.Text =
CType(Session.Item("empstrReferenceFile"), String)
            End If

            Next

        End If

        If e.Item.ItemType = ListItemType.Header Then

            Dim td As TableCell
            Dim glyph As Label
            Dim sortstring As String
            sortstring = viewstate("sortfield")

            For i As Integer = 2 To Datagrid1.Columns.Count - 1 Step 1

                If Datagrid1.Columns(i).SortExpression = sortstring Then
```



```
        If viewstate("sortdirection") = "DESC" Then
            td = e.Item.Controls(i)
            glyph = New Label
            glyph.Font.Size = FontUnit.XSmall
            glyph.Text = "<img src=images/orderdesc.gif>"
            td.Controls.Add(glyph)
        ElseIf viewstate("sortdirection") = "ASC" Then
            td = e.Item.Controls(i)
            glyph = New Label
            glyph.Text = "<img src=images/orderasc.gif>"
            td.Controls.Add(glyph)
        End If
    End If
Next
End If

If e.Item.ItemType = ListItemType.Pager Then
    Dim tc As System.Web.UI.WebControls.TableCell =
CType(e.Item.Controls(0), System.Web.UI.WebControls.TableCell)
    Dim btnPage As LinkButton = New LinkButton
    Dim counter As Integer
    Dim labelpage As Label
    Dim btnprev As LinkButton = New LinkButton
    Dim btnnext As LinkButton = New LinkButton
    Dim Labelnext As Label = New Label
    Dim Labelprev As Label = New Label
    Dim PreviousPageLinkText As String = "<img
src=images/previous.gif border=0 align=top>"
    Dim NextPageLinkText As String = "<img src=images/next.gif
border=0 align=top>"
    Dim MaxNumPageForwardLinks As Integer = 5
    Dim MaxNumPageBackLinks As Integer = 5
    Dim PageLinkSeparator As String = "&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;"

    tc.Controls.Clear()

    If Datagrid1.PageCount > 0 AndAlso Datagrid1.CurrentPageIndex
<> 0 Then

        btnprev = New LinkButton
        With btnprev
            .ID = "btnPrevious"
            .CommandName = "PREV"
            .Text = PreviousPageLinkText
            AddHandler .Click, AddressOf PageIndexChanged
        End With
        tc.Controls.Add(btnprev)
    Else

        Labelprev = New Label
        Labelprev.Text = ""
        tc.Controls.Add(Labelprev)
    End If

    tc.Controls.Add(New LiteralControl(PageLinkSeparator))

    If Datagrid1.PageCount > 1 Then
```



```
        Dim intCurrentPage As Integer =
Datagrid1.CurrentPageIndex + 1
        Dim intStartPage As Integer = Math.Max(intCurrentPage -
MaxNumPageBackLinks, 1)
        Dim intEndPage As Integer = Math.Min(intCurrentPage +
MaxNumPageForwardLinks, Datagrid1.PageCount)

        For counter = intStartPage To intEndPage
            If Datagrid1.CurrentPageIndex = counter - 1 Then

                labelpage = New Label
                labelpage.Text = counter.ToString()
                labelpage.Attributes.Add("style", "color: black;
font-weight: bold")
                tc.Controls.Add(labelpage)
            Else

                btnPage = New LinkButton
                With btnPage
                    .ID = "btnPage" & counter.ToString()
                    .CommandName = "PAGE"
                    .CommandArgument = counter.ToString()
                    .Text = counter.ToString()
                    AddHandler .Click, AddressOf PageIndexChanged
                End With
                tc.Controls.Add(btnPage)
            End If

            tc.Controls.Add(New
LiteralControl(PageLinkSeparator))
        Next
    End If

    If Datagrid1.PageCount > 0 AndAlso Datagrid1.CurrentPageIndex
<> Datagrid1.PageCount - 1 Then

        btnnext = New LinkButton
        With btnnext
            .ID = "btnNext"
            .CommandName = "NEXT"
            .Text = NextPageLinkText
            AddHandler .Click, AddressOf PageIndexChanged
        End With
        tc.Controls.Add(btnnext)
    Else

        Labelnext = New Label
        Labelnext.Text = ""
        Labelnext.Attributes.Add("style", "color: gray")
        tc.Controls.Add(Labelnext)
    End If
```



```
tc.Controls.Add(New LiteralControl(PageLinkSeparator))

End If

End Sub

Public Sub okclick(ByVal sender As Object, ByVal e As EventArgs)

    Dim filterAnakoinosiID As TextBox
    Dim filterTitle As TextBox
    Dim filterType As TextBox
    Dim filterShortDescription As TextBox
    Dim filterLongDescription As TextBox
    Dim filterReferenceURL As TextBox
    Dim filterPublisher As TextBox
    Dim filterImerominiaKataxorisiss As TextBox
    Dim filterProtereotita As TextBox
    Dim filterStatus As TextBox
    Dim filterReferenceFile As TextBox

    Dim myCommand As String
    dsfilter = New DataSet
    Datagrid1.CurrentPageIndex = 0
    filterbar = go.Parent.Parent
    viewstate("sortfield") = Nothing

    MyDataAdapter = New SqlDataAdapter(myCommand, MyConnection)

    filterAnakoinosiID = filterbar.Controls(2).Controls(0)
    filterTitle = filterbar.Controls(3).Controls(0)
    filterType = filterbar.Controls(4).Controls(0)
    filterShortDescription = filterbar.Controls(5).Controls(0)
    filterLongDescription = filterbar.Controls(6).Controls(0)
    filterReferenceURL = filterbar.Controls(7).Controls(0)
    filterPublisher = filterbar.Controls(8).Controls(0)
    filterImerominiaKataxorisiss = filterbar.Controls(9).Controls(0)
    filterProtereotita = filterbar.Controls(10).Controls(0)
    filterStatus = filterbar.Controls(11).Controls(0)
    filterReferenceFile = filterbar.Controls(12).Controls(0)

    sCommand = "select * from " & tablename & " where l=1"
    If ((filterAnakoinosiID.Text <> "")) Then
        Session("empstrAnakoinosiID") = filterAnakoinosiID.Text
        sCommand = sCommand + " and AnakoinosiID like '%" &
(CType(Session.Item("empstrAnakoinosiID"), String)) & "%'"
    Else
        Session("empstrAnakoinosiID") = Nothing
    End If

    If ((filterTitle.Text <> "")) Then
        Session("empstrTitle") = filterTitle.Text
```




```
        sCommand = sCommand + "and Title like '%" &
(CType(Session.Item("empstrTitle"), String)) & "%'"
    Else
        Session("empstrTitle") = Nothing
    End If

    If ((filterType.Text <> "")) Then
        Session("empstrType") = filterType.Text
        sCommand = sCommand + "and Type like '%" &
(CType(Session.Item("empstrType"), String)) & "%'"
    Else
        Session("empstrType") = Nothing
    End If

    If ((filterShortDescription.Text <> "")) Then
        Session("empstrShortDescription") =
filterShortDescription.Text
        sCommand = sCommand + "and ShortDescription like '%" &
(CType(Session.Item("empstrShortDescription"), String)) & "%'"
    Else
        Session("empstrShortDescription") = Nothing
    End If

    If ((filterLongDescription.Text <> "")) Then
        Session("empstrLongDescription") = filterLongDescription.Text
        sCommand = sCommand + "and LongDescription like '%" &
(CType(Session.Item("empstrLongDescription"), String)) & "%'"
    Else
        Session("empstrLongDescription") = Nothing
    End If

    If ((filterReferenceURL.Text <> "")) Then
        Session("empstrReferenceURL") = filterReferenceURL.Text
        sCommand = sCommand + "and ReferenceURL like '%" &
(CType(Session.Item("empstrReferenceURL"), String)) & "%'"
    Else
        Session("empstrReferenceURL") = Nothing
    End If

    If ((filterPublisher.Text <> "")) Then
        Session("empstrPublisher") = filterPublisher.Text
        sCommand = sCommand + "and Publisher like '%" &
(CType(Session.Item("empstrPublisher"), String)) & "%'"
    Else
        Session("empstrPublisher") = Nothing
    End If

    If ((filterImerominiaKataxorisiss.Text <> "")) Then
        Session("empstrImerominiaKataxorisiss") =
filterImerominiaKataxorisiss.Text
        sCommand = sCommand + "and ImerominiaKataxorisiss like '%" &
(CType(Session.Item("empstrImerominiaKataxorisiss"), String)) & "%'"
    Else
        Session("empstrImerominiaKataxorisiss") = Nothing
    End If

    If ((filterProtereotita.Text <> "")) Then
        Session("empstrProtereotita") = filterProtereotita.Text
```



```
        sCommand = sCommand + "and Protereotita like '%" &
(CType(Session.Item("empstrProtereotita"), String)) & "%'"
    Else
        Session("empstrProtereotita") = Nothing
    End If

    If ((filterStatus.Text <> "")) Then
        Session("empstrStatus") = filterStatus.Text
        sCommand = sCommand + "and Status like '%" &
(CType(Session.Item("empstrStatus"), String)) & "%'"
    Else
        Session("empstrStatus") = Nothing
    End If

    If ((filterReferenceFile.Text <> "")) Then
        Session("empstrReferenceFile") = filterReferenceFile.Text
        sCommand = sCommand + "and ReferenceFile like '%" &
(CType(Session.Item("empstrReferenceFile"), String)) & "%'"
    Else
        Session("empstrReferenceFile") = Nothing
    End If

MyDataAdapter = New SqlDataAdapter(sCommand, MyConnection)
MyDataAdapter.Fill(dsfilter, tablename)
Datagrid1.DataSource = dsfilter
Datagrid1.DataMember = tablename
Datagrid1.DataBind()

Session("empmycom") = sCommand
Session("empfilAnakoinosi") = "notnothing"

End Sub
```

Ο παραπάνω κώδικας ελέγχει αν ο χρήστης έχει πληκτρολογήσει σε κάποιο textbox αναζήτησης κάποια τιμή. Αν αυτό ισχύει τότε προσθέτει στο query που χρησιμοποιείται για το γέμισμα της λίστας την αντίστοιχη παράμετρο.

```
Protected Sub Datagrid1_ItemDataBound(ByVal sender As Object, ByVal e
As DataGridItemEventArgs)
    If ListItemType.Item = e.Item.ItemType Or
ListItemType.AlternatingItem = e.Item.ItemType Or ListItemType.EditItem =
e.Item.ItemType Then

        Dim dr As DataRowView = CType(e.Item.DataItem, DataRowView)
        If (dr("ReferenceFile") Is System.DBNull.Value) Then

            Else

                Dim lnkReferenceFile As LinkButton =
CType(e.Item.FindControl("lnkReferenceFile"), LinkButton)
```



```
        lnkReferenceFile.Attributes("onclick") =  
String.Format("javascript:var  
ret=window.open('gw/getFile.aspx?fileid={0}', 'wcCharterParty',  
'width=500,height=500,menubar=0,toolbar=0,status=0,location=0,resizable=1  
,scrollbars=1');", dr("ReferenceFile"))  
        End If  
  
    End If  
  
End Sub  
  
End Class
```



ΚΕΦΑΛΑΙΟ 4. ΑΞΙΟΛΟΓΗΣΗ

Επόμενο βήμα της διαδικασίας είναι το τεστάρισμα της εφαρμογής και η γενικότερη αξιολόγηση της. Αφού δοκιμάσαμε να εκτελέσουμε όλες τις διεργασίες που μας παρέχει η εφαρμογή μας όπως insert, delete, update, ταυτόχρονα σε διαφορετικούς σταθμούς εργασίας παρατηρήσαμε ότι η εφαρμογή μας προλαμβάνει τα λάθη από το χρήστη λόγω των validators που έχουμε χρησιμοποιήσει, παρέχει σταθερότητα, ασφάλεια και δεν ‘κрасάρει’. Τα αποτελέσματα στην βάση δεδομένων ήταν ακριβή και οι διεργασίες εκτελούνταν χωρίς κανένα σφάλμα ενώ δεν παρουσιάστηκαν run time errors ή debugging errors.



ΚΕΦΑΛΑΙΟ 5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Οι υψηλές επιδόσεις στις Web εφαρμογές είναι κάτι το οποίο πρέπει να απασχολεί τους προγραμματιστές καθώς οι ο δυναμικός χαρακτήρας των ASP.NET εφαρμογών μπορεί εύκολα να προκαλέσει προβλήματα στις επιδόσεις που μπορεί να περάσουν χωρίς να εντοπισθούν. Ωστόσο θα πρέπει να έχουμε στο μυαλό μας ότι μια Web εφαρμογή είναι τόσο αποτελεσματική όσο έχει σχεδιαστεί να είναι. Το ASP.NET προσφέρει πολλές νέες και εξαιρετικές λειτουργίες, αλλά η κατάλληλη χρήση αυτών των στοιχείων είναι κάτι απαραίτητο για τις συνολικές επιδόσεις.

Πλεονεκτήματα που παρέχει το σύστημα μας

Τι σύστημα διαχείρισης δεδομένων το οποίο έχουμε υλοποιήσει είναι εύχρηστο και παρέχει στο χρήστη την εύκολη και γρήγορη διαχείριση του περιεχομένου του. Δίνει τη δυνατότητα στο χρήστη να εισάγει περιεχόμενο στην σελίδα, να πραγματοποιήσει αλλαγές στο περιεχόμενο, να διαγράψει περιεχόμενο καθώς και να φορτώσει αρχεία διαφόρων τύπων. Επίσης συνεργάζεται με το u-tube ώστε ο χρήστης να μπορεί να ανεβάσει video στην εφαρμογή.

Έχει σχεδιαστεί σύμφωνα με την αρχιτεκτονική τριών επιπέδων (3-tier) η οποία διασφαλίζει την αξιόπιστη λειτουργία των εφαρμογών όταν αυτές χρησιμοποιούνται ταυτόχρονα από πολλούς χρήστες.

Επίσης έχουμε χρησιμοποιήσει προτυποποιημένο κώδικα, το οποίο σημαίνει ότι επαναχρησιμοποιούμε σταθερό και σωστά δομημένο κώδικα το οποίο εξασφαλίζει την ευκολία σε μελλοντικές επεκτάσεις του συστήματος μας.

Στην εφαρμογή μας έχουμε χρησιμοποιήσει για την ανάκτηση δεδομένων από τη βάση δεδομένων αποθηκευμένες διαδικασίες (store procedures) οι οποίες προσφέρουν καλύτερες επιδόσεις και λειτουργικότητα. Είναι προτιμότερο να χρησιμοποιούμε SQL διαδικασίες που βρίσκονται αποθηκευμένες στον διακομιστή της βάσης δεδομένων για λειτουργίες που περιέχουν πολύπλοκα ερωτήματα. Οι αποθηκευμένες διαδικασίες εκτελούνται γενικά καλύτερα από τα ειδικά ερωτήματα, ειδικά όταν τα ερωτήματα είναι μεταγλωττισμένα και έτοιμα για εκτέλεση στον διακομιστή.



Για την προβολή και επεξεργασία των δεδομένων μας που επιστρέφονται από τη βάση δεδομένων έχουμε χρησιμοποιήσει 2 στοιχεία ελέγχου Web φόρμας, τα στοιχεία ελέγχου DataGrid και DataList. Το στοιχείο ελέγχου DataGrid είναι ένα πλέγμα με πολλές στήλες δεδομένων που έχει πολλές επιπλέον λειτουργίες για τη διάταξη των περιεχομένων του, μαζί με λειτουργίες ταξινόμησης, επεξεργασίας, επιλογής και μετακίνησης στα δεδομένα.

Αναμφίβολα η χρήση των web φορμών του ASP.NET είναι ένα από τα καλύτερα στοιχεία του ASP.NET. Χωρίς τα στοιχεία ελέγχου επικύρωσης, όπως για παράδειγμα η δύσκολη διαδικασία της διαχείρισης της κατάστασης και της επικύρωσης θα έπαιρνε πολύ περισσότερο χρόνο. Ωστόσο θα πρέπει να γνωρίζουμε ότι αν και τα στοιχεία ελέγχου διακομιστή είναι πολύ καλύτερα από τα αντίστοιχα HTML στοιχεία όσον αφορά τις λειτουργίες και την παραγωγικότητα του προγραμματιστή, απαιτούν περισσότερους πόρους και δεν είναι τόσο γρήγορα όσο τα HTML στοιχεία ελέγχου πελάτη. Γι' αυτό δεν θα πρέπει να αντικαταστήσουμε ολοκληρωτικά τα HTML στοιχεία ελέγχου με ASP.NET στοιχεία ελέγχου προκειμένου να μετατρέψουμε όλο τον κώδικα σε ASP.NET παρά μόνο όπου χρειάζεται. Για παράδειγμα μπορούμε να έχουμε HTML στοιχεία ελέγχου αν απλώς θέλουμε να εμφανίζουμε κάποια δεδομένα στην οθόνη. Στις πλατφόρμες μας έχουμε χρησιμοποιήσει στοιχεία ελέγχου επικύρωσης HTML τα οποία όπως προαναφέραμε είναι πιο γρήγορα και καταναλώνουν λιγότερους πόρους του συστήματος μας. Έτσι με τη χρήση τους προλαμβάνονται τα λάθη από την πλευρά του χρήστη.

Τέλος, είναι χρήσιμο να χρησιμοποιούμε το Web.config για να προσδιορίσουμε τις global τιμές. Στην εφαρμογή μας έχουμε δηλώσει το Connection Sting στο Web.config για τη σύνδεση με τη βάση δεδομένων. Τα Configuration αρχεία μπορούν να φορτωθούν ή να αλλάξουν ενώ η εφαρμογή τρέχει και έτσι πλέον δεν χρειάζεται να γίνει επανεκκίνηση του server.

Έχουμε χρησιμοποιήσει την try – catch μέσα στην οποία μπορούν να εκτελούνται εντολές, είτε εμφανιστεί μια εξαίρεση είτε όχι. Έτσι οποιαδήποτε εξαίρεση και αν προκύψει αποφεύγουμε τη διακοπή της λειτουργίας της εφαρμογής μας η οποία τρέχει.



Πλεονεκτήματα ASP.NET

ASP.NET Controls

Το ASP.NET περιέχει ένα μεγάλο set από HTML ελέγχους. Σχεδόν όλα τα HTML elements σε μια σελίδα μπορούν να οριστούν σαν ASP.NET control objects και μπορούν να ελεγχθούν από κώδικα. Το ASP.NET περιέχει επίσης ένα νέο set από objects ελέγχου εισαγωγής, όπως list boxes και ελέγχους επικύρωσης. Ένα νέο data grid control υποστηρίζει ταξινόμηση(sorting), data paging, και όλα όσα θα μπορούσαμε να περιμένουμε από ένα dataset control.

Event Aware Controls

Όλα τα ASP.NET objects σε μία Web page μπορούν να εκθέσουν events τα οποία μπορούν να επεξεργαστούν με ASP.NET κώδικα. Τα events όπως load, κλικ και αλλαγή χειριζόμενα από κώδικα κάνουν την κωδικοποίηση απλούστερη και περισσότερο οργανωμένη.

ASP.NET Components

Τα ASP.NET components βασίζονται σε XML. Ένα component μπορεί εύκολα να χρησιμοποιηθεί από τον προγραμματιστή που το ανέπτυξε ή από κάποιον άλλο. Ένα component σχεδιασμένο για πρόσβαση σε μια βάση δεδομένων παραδείγματος χάριν, μπορεί να χρησιμοποιηθεί από πολλές διαφορετικές εφαρμογές, αν έχουν περίπου τις ίδιες ανάγκες πρόσβασης σε δεδομένα.

Authentication χρήστη

Το ASP.NET υποστηρίζει forms-based πιστοποίηση χρήστη, συμπεριλαμβάνοντας διαχείριση των cookies και αυτόματο redirecting σε unauthorized logins.

Λογαριασμοί χρηστών και ρόλοι

Το ASP.NET επιτρέπει για λογαριασμούς χρηστών και ρόλων, να δώσει σε κάθε χρήστη πρόσβαση σε διαφορετικούς κώδικες server και εκτελέσιμα.



Υψηλή κλιμάκωση

Η Server to server επικοινωνία έχει εμπλουτιστεί αρκετά κάνοντας εφικτή ,την κλιμάκωση μιας εφαρμογής πάνω σε διαφορετικούς servers. Ένα παράδειγμα είναι η ικανότητα εκτέλεσης XML παραμέτρων, XSL transformations και ακόμα resource session objects σε άλλους servers.

Μεταγλωττισμένος κώδικας

Το πρώτο αίτημα για μια ASP.NET σελίδα στο server θα μεταγλωττίσει τον ASP.NET κώδικα και θα κρατήσει ένα cached αντίγραφο στη μνήμη. Σαν αποτέλεσμα θα έχουμε υψηλότερη απόδοση.

Εύκολο Configuration

Το Configuration στο ASP.NET πραγματοποιείται με τη χρήση απλών text αρχείων.

Τα Configuration αρχεία μπορούν να φορτωθούν ή να αλλάξουν ενώ η εφαρμογή τρέχει. Πλέον δεν χρειάζεται να γίνει επανεκκίνηση του server και δεν υπάρχει πλέον metabase ή πολυπλοκότητα με τη registry .

Εύκολο Deployment

Δεν χρειάζεται πλέον επανεκκίνηση του server για την ανάπτυξη ή την αντικατάσταση του μεταγλωττισμένου κώδικα. Το ASP.NET απλά ανακατευθύνει όλα τα νέα αιτήματα στο νέο κώδικα.

Συμβατότητα

Το ASP.NET δεν είναι απόλυτα συμβατό με τις παλαιότερες εκδόσεις ASP, έτσι ο περισσότερος κώδικας από τις παλαιότερες εκδόσεις ASP θα χρειαστούν μερικές τροποποιήσεις για να τρέξουν σε ASP.NET. Για να ξεπεραστεί αυτό το πρόβλημα το ASP.NET χρησιμοποιεί την επέκταση αρχείου ".aspx". Αυτό κάνει τις ASP.NET εφαρμογές ικανές να τρέχουν με μαζί με τις standard ASP εφαρμογές στον ίδιο server.



Μελλοντική εργασία

Η μελλοντική εργασία που θα μπορούσε να γίνει είναι η επέκταση του συστήματος μας με επιπλέον οντότητες που θα μπορεί να διαχειριστεί αλλά και η αυτοματοποίηση κάποιων διαδικασιών.



ΚΕΦΑΛΑΙΟ 7. ΒΙΒΛΙΟΓΡΑΦΙΑ

- Μάθετε την ASP.NET , Joe Martin , Brett Tomson Εκδόσεις Γκιούρδας
- ASP.NET Web Developer 's Guide Mesbah Ahmed, Chris Garrett, Jeremy Faircloth, Chris Payne, DotThatCom.com, Wei Meng Lee Series Editor, Jonothon Ortiz Technical Editor
- Transact – SQL Lowell Mauer, et al.
- Beginning ASP.Net 1.1 E-Commerce Cristian Darie, Karli Watson, Εκδόσεις Apress
- Building Web Solutions with ASP.NET and ADO.NET, Dino Esposito, Εκδόσεις Microsoft Press
- <http://www.ebusinessforum.gr/>
- <http://www.go-online.gr/>
- <http://www.3wschool.com>
- <http://msdn.microsoft.com/>
- <http://www.infosoc.gr>



Summary

The subject of this project is to create a content management system (CMS) for the management of a marketing portal. With this CMS users can easily change the articles, the photos, can insert new content or delete content.

A **content management system** (CMS) is a program used to create a web application for the content of a Web site. CMSes are deployed primarily for interactive use by a potentially large number of contributors.

The content managed includes files such as, image media, audio, documents which combine the web content. The idea behind a CMS is to make these files available inter-office, as well as over the web. A CMS will often be used as an archive as well. Many companies use a CMS to store files in a non-proprietary form. Companies use a CMS to share files with ease, as most systems use server-based software, even further broadening file availability.

For the implementation of this system we have used ASP.NET and MS SQL Server.