

ΣΥΣΤΗΜΑ ΔΙΑΘΕΣΗΣ ΦΩΤΟΓΡΑΦΙΩΝ



**ΤΕΧΝΟΛΟΓΙΚΟ
ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΚΡΗΤΗΣ**

Εισηγητής: Παχουλάκης Ιωάννης
Όνομα σπουδαστή: Τζούμπας Αχιλλέας
ΑΜ: 1275

Ευχαριστίες

Θα Θέλα να ευχαριστήσω όλους όσους συμμετείχαν και συνέβαλαν στην ολοκλήρωση της παρούσας πτυχιακής εργασίας. Ειδικότερα τον εισηγητή μου, κ. Παχουλάκη Ιωάννη για την βοήθεια και την καθοδήγησή του. Τα μέλη της εξεταστικής επιτροπής που μου έκαναν την τιμή να αξιολογήσουν την προσπάθειά μου. Τέλος την οικογένεια μου, που όλα αυτά τα χρόνια ήταν δίπλα μου και με στήριζε στις προσπάθειες μου .

Σας ευχαριστώ,
Τζούμπας Αχιλλέας

ΠΕΡΙΕΧΟΜΕΝΑ

1. Κεφάλαιο – Το λογισμικό πακέτο.....	4
1.1 Συνοπτική περιγραφή του συστήματος.....	4
1.2 Απαιτήσεις συστήματος.....	5
2. Κεφάλαιο - Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης.....	6
2.1 Πλατφόρμα .NET.....	6
2.1.1 Τι είναι το .NET.....	6
2.1.2 Τι είναι η C#.....	8
2.1.3 Πλεονεκτήματα .NET.....	8
2.1.3 Web Forms και ASP.NET.....	9
2.2 Βάσεις δεδομένων και SQL Server 2005.....	10
3. Κεφάλαιο – Περιγραφή της Βάσης Δεδομένων.....	12
3.1 Πίνακες της Βάσης Δεδομένων.....	12
3.2 Διάγραμμα Βάσης Δεδομένων – Συσχετίσεις πινάκων.....	20
3.3 Έτοιμες Διεργασίες – Stored Procedure.....	21
3.4 Σύνδεση με Βάση Δεδομένων.....	22
4. Κεφάλαιο – Ανάλυση της εφαρμογής μας.....	24
4.1 Η φόρμα DigitalGallery.Master.....	24
4.2 Η φόρμα homePage.aspx.....	26
4.3 Η φόρμα viewImage.aspx.....	29
4.4 Η φόρμα upLoadImage.aspx.....	31
4.5 Η φόρμα viewCategory.aspx.....	32
4.6 Η φόρμα rememberPassword.aspx.....	33
4.7 Η φόρμα registration.aspx.....	34
4.8 Η φόρμα communication.aspx.....	35
4.9 Η φόρμα changePassword.aspx.....	35
5. Κεφάλαιο – Βιβλιογραφία – Πηγές.....	37

1. Κεφάλαιο – Το λογισμικό πακέτο

Συνοπτική περιγραφή του συστήματος

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εφαρμογής όπου ο κάθε εγγεγραμμένος χρήστης θα μπορεί να δημοσιεύει τις φωτογραφίες που επιθυμεί καθώς και να τις κατεβάσει. Ακόμη θα έχει την δυνατότητα και να τις ψηφίζει. Η εφαρμογή έχει αναπτυχθεί σε τεχνολογία ASP.NET με την βοήθεια του Visual Studio όπου είναι το κύριο εργαλείο για ανάπτυξη .NET εφαρμογών. Η βάση δεδομένων της εφαρμογής έγινε με τον SQL Server 2005 Express Edition. Η βάση δεδομένων περιλαμβάνει όλα τα συστατικά που είναι απαραίτητα για την υλοποίηση μια τέτοιας ηλεκτρονικής σελίδας.

Απαιτήσεις Συστήματος

Οι απαιτούμενες τεχνικές προδιαγραφές της πλατφόρμας εγκατάστασης είναι οι ακόλουθες:

- Για την πλατφόρμα .NET Framework απαιτείται ένα από τα παρακάτω λειτουργικά συστήματα:
 - Windows XP Home Edition with SP2
 - Windows XP Professional with SP2
 - Windows XP Starter Edition
 - Windows Vista Starter
 - Windows Vista Home Basic
 - Windows Vista Home Premium
 - Windows Vista Ultimate
 - Windows Vista Business

- Για την πλατφόρμα του SQL Server 2005 SQL Express Edition ένα από τα παρακάτω λειτουργικά συστήματα:
 - Microsoft Windows 2000 Server with Service Pack (SP) 4 η νεότερη
 - Windows 2000 Professional Edition with SP 4
 - Microsoft Windows 2000 Server
 - Windows XP με SP 2
 - Windows Server 2003 Enterprise Edition, Standard Edition, or Datacenter Edition με SP 1
 - Windows Small Business Server 2003 with SP

- Κοινές προδιαγραφές:
 - Μνήμη: 192MB τουλάχιστον (προτεινόμενη 512MB ή μεγαλύτερη)
 - Επεξεργαστής: Pentium 600 MHz ή γρηγορότερος
 - Web server: Microsoft Internet Information Services (IIS) 5.0+

2 Κεφάλαιο – Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης

2.1 Πλατφόρμα .NET

Η πλατφόρμα .NET που χρησιμοποιήθηκε είναι μια πλατφόρμα ανάπτυξης λογισμικού με πολλές δυνατότητες στις εφαρμογές δικτύου και μεγάλη ευκολία στη σύνδεση με διάφορες βάσεις δεδομένων. Οι εφαρμογές της είναι δυναμικές σελίδες που τρέχουν στον Web Server και τα αποτελέσμα τους (σε html) εμφανίζεται στον Browser του χρήστη.

2.1.1 Τι είναι το .NET

Το .NET είναι μια νέα πλατφόρμα ανάπτυξης εφαρμογών σε περιβάλλοντα Windows. Σύμφωνα με την Microsoft, έχει ως σκοπό την απλοποίηση της ανάπτυξης εφαρμογών «κρύβοντας» τις τεχνικές λεπτομέρειες υλοποίησης πολλών λειτουργιών, όπως διαχείριση μνήμης, επικοινωνία μέσω δικτύου, είσοδο/έξοδο από συσκευές και αφήνοντας το προγραμματιστή ελεύθερο να επικεντρωθεί στην «λογική» του προγράμματος.

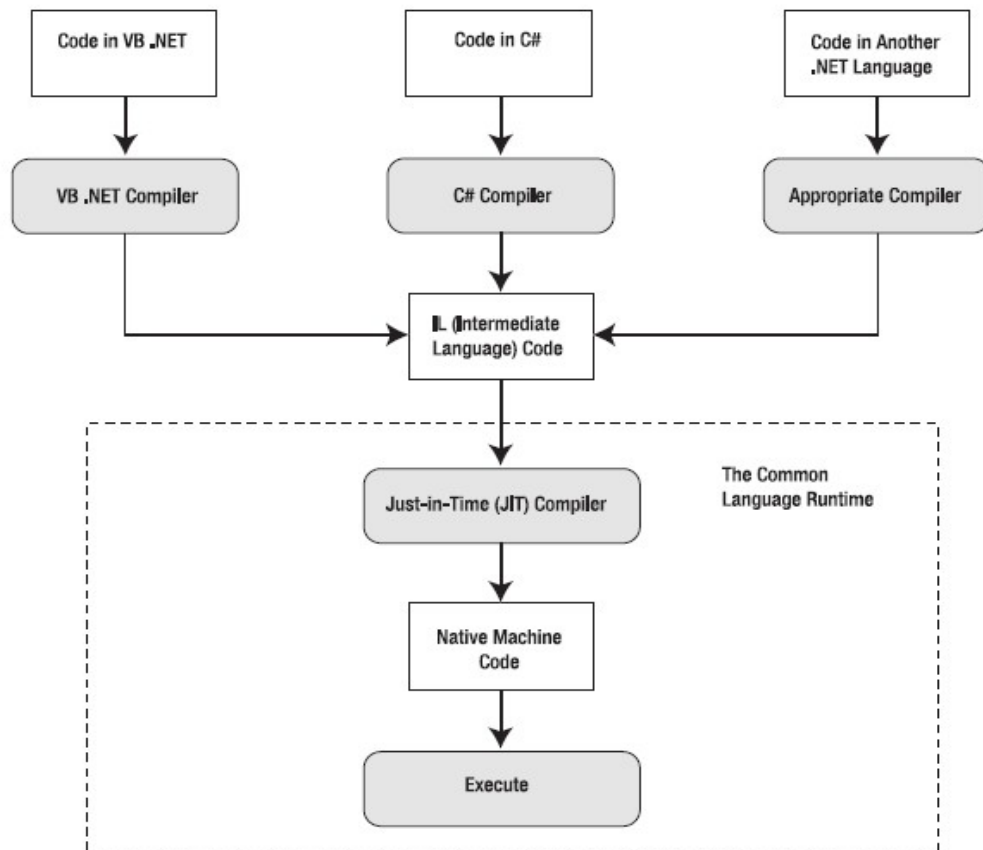
Το .NET χαρακτηρίζεται ως μία πλατφόρμα διαχείρισης με την έννοια ότι δημιουργεί ένα ελεγχόμενο και ασφαλές περιβάλλον μέσα στο οποίο μπορεί να τρέξει μια εφαρμογή. Η ασφάλεια έγκειται για παράδειγμα στον έλεγχο στην δέσμευση και προσπέλαση της μνήμης (δεν υπάρχουν pointers, δεν μπορείς να προσπελάσεις μια θέση μνήμης εκτός πίνακα), στο τύπο των μεταβλητών και δεδομένων (δεν μπορείς να θέσεις μια float τιμή σε μια ακέραια μεταβλητή) ή στην αυτόματη υλοποίηση δικλίδων ασφαλείας.

Το .NET υποστηρίζει πληθώρα γλωσσών προγραμματισμού οι οποίες είναι ειδικά σχεδιασμένες για αυτό, όπως C#, Visual Basic.NET, J++ και managed C++. Στην πραγματικότητα, το .NET καταλαβαίνει μόνο μια γλώσσα προγραμματισμού την Microsoft Intermediate Language (MSIL). Συνεπώς, οποιαδήποτε γλώσσα

προγραμματισμού μπορεί να μεταγλωττιστεί σε MSIL μπορεί να τρέξει στην πλατφόρμα .NET. Ο χρήστης μπορεί ακόμα να γράψει απευθείας ένα πρόγραμμα σε MSIL στο σημειωματάριο (Notepad) να το κάνει compile και να το τρέξει στο .NET!

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

Το CLR ενεργοποιεί έναν compiler το JIT (Just-in-Time) ο οποίος είναι υπεύθυνος για την μετατροπή του ενδιάμεσου κώδικα MSIL σε γλώσσα μηχανής. Εν κατακλείδι, το MSIL κατά την μεταγλώττιση ενός C# προγράμματος εξάγει και metadata. Τα metadata περιλαμβάνουν στοιχεία που χρησιμοποιεί η CLR για να φορτώσει συγκεκριμένες λειτουργίες, να τοποθετήσει αντικείμενα στην μνήμη, να ελέγξει την ασφάλεια και να πραγματοποιήσει μια σειρά από πρόσθετες λειτουργίες. Τα metadata συμπεριλαμβάνονται στο ίδιο αρχείο όπως το MSIL. Μια απεικόνιση για το πώς λειτουργεί η όλη διαδικασία είναι η παρακάτω:



2.1.2 Τι είναι η C#

Η C# είναι μια σχετικά νέα αντικειμενοστραφής γλώσσα προγραμματισμού η οποία δημιουργήθηκε από την Microsoft. Δανείζεται πολλά στοιχεία, και έχει παρόμοια σύνταξη, με την C++ και την Java, κάνοντας την εκμάθηση της σχετικά εύκολη. Είναι γλώσσα ειδικά σχεδιασμένη για να υποστηρίζει το .NET framework της ίδιας εταιρείας. Βασικό χαρακτηριστικό της είναι ότι δεν παράγει απευθείας κώδικα μηχανής όπως η C++, αλλά ένα ενδιάμεσο κώδικα που στοχεύει το .NET.

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

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

Όσο αφορά των κώδικα της γλώσσας αυτής, μας δίνει την δυνατότητα της κληρονομικότητας (inheritance), και της δημιουργίας κλάσεων (class). Ο προγραμματιστής έχει την δυνατότητα να κληρονομεί μια κλάση και έτσι να δανείζεται της ιδιότητες της. Υπάρχει η δυνατότητα να χρησιμοποιούμε έτοιμες μεθόδους, και εμείς απλώς να δίνουμε τις δικές μας παραμέτρους. Ακόμη μπορούμε να «πιάσουμε» ένα λάθος που θα προκύψει στον κώδικα μας και αυτό γίνεται με την εντολή try catch final.

Τέλος δίνει την δυνατότητα στον προγραμματιστή να χρησιμοποιεί έτοιμα πακέτα έτσι ώστε να κάνει το έργο του πιο εύκολο και πιο αποδοτικό.

2.1.3 Πλεονεκτήματα .NET

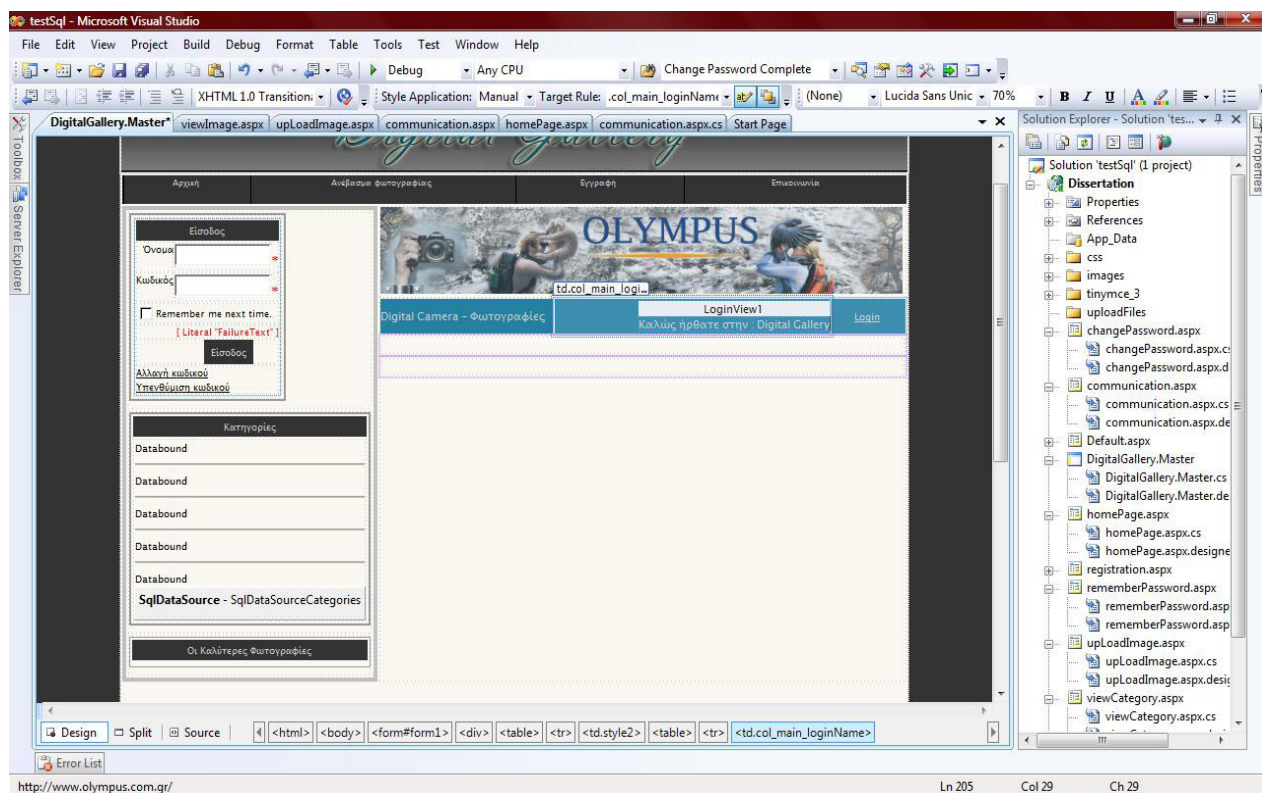
Τα πλεονεκτήματα που διαθέτει η .NET για την ανάπτυξη εφαρμογών είναι τα παρακάτω:

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

- Αυτοματοποιημένη διαχείριση μνήμης, ο χρήστης δεν χρειάζεται να ασχοληθεί με αποδέσμευση μνήμης και όσα αντικείμενα μένουν αχρησιμοποίητα οδηγούνται στον Garbage collector που διαθέτει η CLR

2.1.4 Web Forms και ASP .NET

Τα Web Forms (Φόρμες για web) μας δίνουν την δυνατότητα να φτιάξουμε εύκολα και γρήγορα μια εφαρμογή είτε windows application είναι αυτή, είτε web application. Εδώ το IDE που χρησιμοποιήσαμε είναι το Visual Studio 2008 το οποίο διαθέτει μια εργαλειοθήκη, όπου μπορούμε να την χρησιμοποιήσουμε στις κενές φόρμες που μας δίνει. Για να προγραμματίσουμε την εφαρμογή μας όπως εμείς επιθυμούμε θα πρέπει να γράψουμε κώδικα σε C# (ή οποιαδήποτε άλλη γλώσσα). Απαραίτητη προϋπόθεση είναι η ύπαρξη του αντίστοιχου διακομιστή της πλατφόρμας .NET (.NET Framework 3.5). Μια εικόνα για το περιβάλλον που διαθέτουμε για να προγραμματίσουμε την εφαρμογή μας είναι η παρακάτω:



Αφού επιλέξουμε τη εφαρμογή θα φτιάξουμε, δηλαδή web ή Windows application θα πρέπει να βρούμε το πού θα «φιλοξενηθεί» η εφαρμογή μας. Η φιλοξενία μπορεί να γίνει είτε από έναν απομακρυσμένο διακομιστή, είτε από έναν που βρίσκεται στο τοπικό δίκτυο. Στην περίπτωση μας χρησιμοποιούμε έναν τοπικό διακομιστή ο οποίος υπάρχει στο Visual Studio και λειτουργεί ως web server.

Ύστερα τα πράγματα είναι απλά, καθώς το μόνο που χρειάζεται, είναι να μεταφέρουμε τα έτοιμα controls που διαθέτει η εργαλειοθήκη στις φόρμες μας, και πίσω από αυτήν την σελίδα, να προγραμματίσουμε για το τι ακριβώς θέλουμε να κάνουν τα control αυτά σε γλώσσα C#.

2.2 Βάσεις Δεδομένων και SQL Server 2005

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

Αρχικά έπρεπε να επιλεγεί ποια Βάση Δεδομένων θα χρησιμοποιηθεί. Οι επιλογές ήταν πολλές όπως Oracle, mySQL, Access, SQL Server 2005 κ.α. Τα κριτήρια επιλογής της βάση ήταν να υποστηρίζει πολλούς χρήστες ταυτόχρονα, να υπάρχει η κατάλληλη τεχνογνωσία και να είναι τύπου server. Τελικά επιλέχθηκε ο SQL Server Express γιατί ήταν η Βάση Δεδομένων που πληρούσε τα περισσότερα κριτήρια.

Παρόλο που διαθέτει μέγιστο όριο δεδομένων, επιλέχτηκε αυτή καθώς τα 4GB είναι υπέρ αρκετά, για μια βάση όπου θα κρατεί δεδομένα μια εφαρμογής που αφορά πτυχιακή άσκηση .

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

Το Microsoft SQL Server είναι σχεδιασμένο για να υποστηρίζει βάσεις δεδομένων μεγάλου όγκου και κρίσιμης σημασίας σε πολλά διαφορετικά πεδία εφαρμογών, στα οποία

περιλαμβάνονται η επεξεργασία συναλλαγών εντός επικοινωνίας (online transaction processing, OLTP), οι αποθήκες δεδομένων (data warehousing) και το ηλεκτρονικό εμπόριο (e-commerce). Για την υποστήριξη αυτών των λειτουργιών, το SQL Server διαθέτει ορισμένα εργαλεία, στα οποία περιλαμβάνονται κάποια βοηθητικά προγράμματα γραμμής διαταγών όπως το bcp.exe, το οποίο αντιγράφει δεδομένα μεταξύ του SQL Server και ενός αρχείου του λειτουργικού συστήματος και τον Enterprise Manager, ένα προηγμένο παραστατικό εργαλείο για τη διαχείριση πολλών βάσεων δεδομένων και του ίδιου του SQL Server.

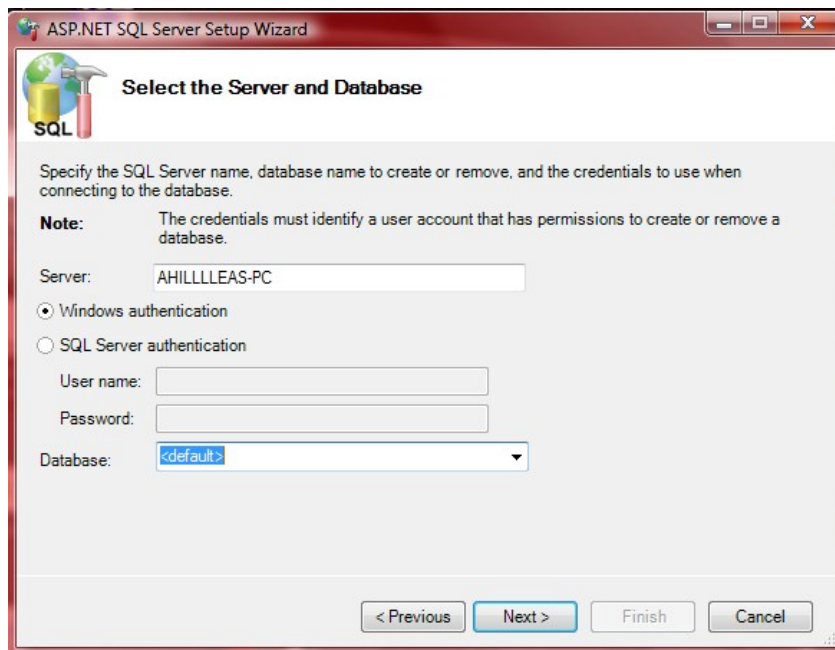
3. Κεφάλαιο – Περιγραφή της Βάσης Δεδομένων

3.1 Πίνακες της Βάσης Δεδομένων

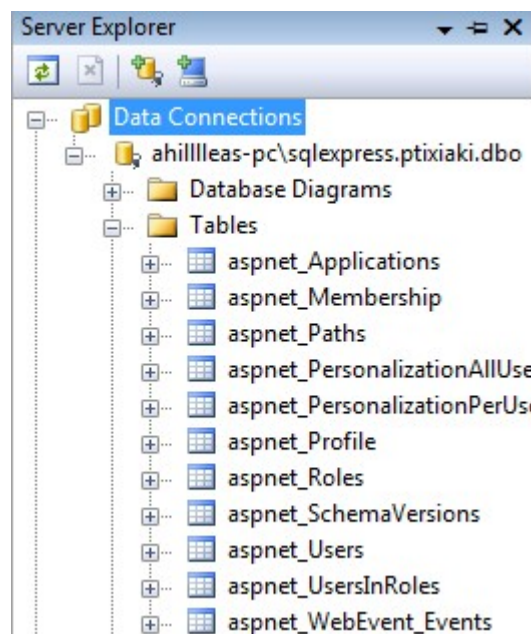
Αρχικά θα σας παρουσιάσω τους πίνακες που φτιάχτηκαν με τις Wizard που διαθέτει η Visual Studio.

Για να φτιάξουμε μερικούς πίνακες όπου θα ελέγχουμε την είσοδο των χρηστών αρκεί να ανοίξουμε το command prompt που διαθέτει το IDE μας και να δώσουμε την εντολή

Aspnet_regsql.exe . Έτσι θα μας εμφανιστεί το παρακάτω παράθυρο:



Όπου εμείς θα χρειαστεί απλά, να δώσουμε το όνομα της βάσης δεδομένων που θέλουμε να αποθηκευτούν οι πίνακες, και το user name και password εάν θέλουμε η βάση μας να έχει κωδικό. Έτσι θα δημιουργηθεί η παρακάτω βάση με τους παρακάτω πίνακες



Ας δούμε τώρα αναλυτικά τους πίνακες που μας δημιούργησε η wizard του IDE μας.

- Πίνακας aspnet_Application

	Column Name	Data Type	Allow Nulls
▶	ApplicationName	nvarchar(256)	<input type="checkbox"/>
	LoweredApplicationName	nvarchar(256)	<input type="checkbox"/>
🔑	ApplicationId	uniqueidentifier	<input type="checkbox"/>
	Description	nvarchar(256)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Σε αυτόν τον πίνακα κρατάμε το όνομα της εφαρμογής μας, αλλά και τον κωδικό που διαθέτει ο οποίος αποτελεί και το πρωτεύον κλειδί. Τέλος δεν επιτρέπονται τα κενά στα πεδία αυτά αλλά μόνο στην περιγραφή της εφαρμογής μας.

- Πίνακας aspnet_Membership

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

Εδώ κρατάμε όλες τις πληροφορίες που αφορούν τους χρήστες μας. Κρατάμε όλα τα στοιχεία για αυτούς με κάθε λεπτομέρεια. Ως πρωτεύον κλειδί ορίζεται ο μοναδικός κωδικός που δίνει το σύστημα σε κάθε user που δημιουργεί.

- Πίνακας aspnet_Paths

	Column Name	Data Type	Allow Nulls
▶	ApplicationId	uniqueidentifier	<input type="checkbox"/>
🔑	PathId	uniqueidentifier	<input type="checkbox"/>
	Path	nvarchar(256)	<input type="checkbox"/>
	LoweredPath	nvarchar(256)	<input type="checkbox"/>
			<input type="checkbox"/>

Σε αυτόν τον πίνακα κρατάει το σημείο στο οποίο βρίσκεται η εφαρμογή μας. Ως πρωτεύον κλειδί ορίζει ένα μοναδικό κωδικό για κάθε path.

- Πίνακας aspnet_PersonalizationAllUsers

	Column Name	Data Type	Allow Nulls
▶🔑	PathId	uniqueidentifier	<input type="checkbox"/>
	PageSettings	image	<input type="checkbox"/>
	LastUpdatedDate	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

Με πρωτεύον κλειδί το path id της εφαρμογής μας κρατάει το πότε έγινε το τελευταίο update στην εφαρμογή μας.

- Πίνακας aspnet_PersonalizationPerUser

	Column Name	Data Type	Allow Nulls
▶🔑	Id	uniqueidentifier	<input type="checkbox"/>
	PathId	uniqueidentifier	<input checked="" type="checkbox"/>
	UserId	uniqueidentifier	<input checked="" type="checkbox"/>
	PageSettings	image	<input type="checkbox"/>
	LastUpdatedDate	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

Αυτός ο πίνακας είναι όμοιος με τον παραπάνω η μόνη διαφορά είναι, ότι αυτός κρατάει και τον μοναδικό κωδικό του χρήστη και ότι διαθέτει ως πρωτεύον κλειδί ένα ξεχωριστό id.

- Πίνακας aspnet_Profile

	Column Name	Data Type	Allow Nulls
▶	UserId	uniqueidentifier	<input type="checkbox"/>
	PropertyNames	ntext	<input type="checkbox"/>
	PropertyValuesString	ntext	<input type="checkbox"/>
	PropertyValuesBinary	image	<input type="checkbox"/>
	LastUpdatedDate	datetime	<input type="checkbox"/>
			<input type="checkbox"/>

Σε αυτόν τον πίνακα κρατάμε το προφίλ του κάθε χρήστη. Υπάρχει διασύνδεση μεταξύ αυτού του πίνακα και του πίνακα aspnet_Users.

- Πίνακας aspnet_Roles

	Column Name	Data Type	Allow Nulls
▶	ApplicationId	uniqueidentifier	<input type="checkbox"/>
▶	RoleId	uniqueidentifier	<input type="checkbox"/>
	RoleName	nvarchar(256)	<input type="checkbox"/>
	LoweredRoleName	nvarchar(256)	<input type="checkbox"/>
	Description	nvarchar(256)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Το Visual Studio μας δίνει την δυνατότητα να έχουμε ρόλους, δηλαδή όσους χρήστες θέλουμε τους κατάσουμε σε κάποιο ρόλο και τους χειριζόμαστε όλους με τον ίδιο τρόπο. Ως πρωτεύον κλειδί ορίζεται ένας μοναδικός κωδικός σε κάθε ρόλο.

- Πίνακας aspnet_SchemaVersions

	Column Name	Data Type	Allow Nulls
▶	Feature	nvarchar(128)	<input type="checkbox"/>
▶	CompatibleSchemaVersion	nvarchar(128)	<input type="checkbox"/>
	IsCurrentVersion	bit	<input type="checkbox"/>
			<input type="checkbox"/>

Οπώς βλέπουμε στον πίνακα αυτό το κλειδί είναι σύνθετο. Υπάρχει και ένα πεδίο isCurrentVersion το οποίο παίρνει μόνο δύο καταστάσεις true και false.

- Πίνακας aspnet_Users

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

Σε αυτόν εδώ των πίνακα κρατούνται τα στοιχεία του χρήστη. Ως πρωτεύον κλειδί ορίζεται ένας μοναδικός κωδικός, που δημιουργείται με την εγγραφή ενός νέου χρήστη. Κρατάει ακόμα και πότε έγινε η τελευταία ενέργεια από των χρήστη στο πεδίο LastActivityDate αφού έχει οριστεί ως πεδίο ημερομηνίας.

- Πίνακας aspnet_UsersInRoles

	Column Name	Data Type	Allow Nulls
▶	UserId	uniqueidentifier	<input type="checkbox"/>
▶	RoleId	uniqueidentifier	<input type="checkbox"/>
			<input type="checkbox"/>

Τα πεδία σε αυτόν εδώ τον πίνακα είναι λίγα αλλά πολλή σημαντικά.Κρατάει για κάθε user τον κωδικό του αλλά και σε ποιο ρόλο ανήκει ο καθένας.

- Πίνακας aspnet_WebEvent_Events

Column Name	Data Type	Allow Nulls
EventId	char(32)	<input type="checkbox"/>
EventTimeUtc	datetime	<input type="checkbox"/>
EventTime	datetime	<input type="checkbox"/>
EventType	nvarchar(256)	<input type="checkbox"/>
EventSequence	decimal(19, 0)	<input type="checkbox"/>
EventOccurrence	decimal(19, 0)	<input type="checkbox"/>
EventCode	int	<input type="checkbox"/>
EventDetailCode	int	<input type="checkbox"/>
Message	nvarchar(1024)	<input checked="" type="checkbox"/>
ApplicationPath	nvarchar(256)	<input checked="" type="checkbox"/>
ApplicationVirtualPath	nvarchar(256)	<input checked="" type="checkbox"/>
MachineName	nvarchar(256)	<input type="checkbox"/>
RequestUrl	nvarchar(1024)	<input checked="" type="checkbox"/>
ExceptionType	nvarchar(256)	<input checked="" type="checkbox"/>
Details	ntext	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Εδώ κρατάμε τα event που γίνονται στην εφαρμογή μας με πλήρης λεπτομέρειες.Ως πρωτεύον κλειδί ορίζεται ένας μοναδικός κωδικός, που δημιουργείται κάθε φορά που έχουμε ένα event.

Έτσι σε αυτό το σημείο τελειώσαμε με τους πίνακες που φτιάχνει αυτόματα wizard του Visual Studio.Τώρα θα ακολουθήσουν οι πίνακες που αφορούν τις φωτογραφίες που διαθέτουμε.

- Πίνακας Categories

	Column Name	Data Type	Allow Nulls
▶ 🔑	Category_ID	int	<input type="checkbox"/>
	Category	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Σε αυτόν εδώ τον πίνακα κρατάμε τις κατηγορίες που διαθέτει η σελίδα μας. Ως πρωτεύον κλειδί υπάρχει το Category_ID όπου παίρνει ως ακέραιο έναν αύξοντα αριθμό με την προσθήκη μιας κατηγορίας. Για να κρατήσουμε το όνομα της κατηγορίας χρησιμοποιούμε το πεδίο Category όπου παίρνει ως μέγιστο 50 χαρακτήρες.

- Πίνακας Images

	Column Name	Data Type	Allow Nulls
▶ 🔑	ImageID	int	<input type="checkbox"/>
	UserID	uniqueidentifier	<input checked="" type="checkbox"/>
	Category_ID	int	<input checked="" type="checkbox"/>
	Title	nvarchar(50)	<input checked="" type="checkbox"/>
	Photografer	nvarchar(50)	<input checked="" type="checkbox"/>
	AddDate	datetime	<input checked="" type="checkbox"/>
	Camera	nvarchar(30)	<input checked="" type="checkbox"/>
	Diafragma	nvarchar(15)	<input checked="" type="checkbox"/>
	Speed	nvarchar(20)	<input checked="" type="checkbox"/>
	ISO	nvarchar(50)	<input checked="" type="checkbox"/>
	Focus	nvarchar(50)	<input checked="" type="checkbox"/>
	Lens	nvarchar(25)	<input checked="" type="checkbox"/>
	Image	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Σε αυτόν εδώ τον πίνακα κρατάμε τα πλήρη στοιχεία για μια φωτογραφία. Εχουμε δημιουργήσει μια σελίδα όπου οι συγκεκριμένοι χρήστες μπορούν να ανεβάσουν ότι φωτογραφία θέλουν, και μαζί με την φωτογραφία περνάμε και στοιχεία για τον τρόπο λήψης της φωτογραφίας. Ως πρωτεύον κλειδί ορίζεται το ImageID όπου παίρνει από μόνο του έναν αύξοντα αριθμό με την προσθήκη φωτογραφίας. Κρατάμε ακόμη και το UserID του χρήστη έτσι ώστε να ξέρουμε ποιος ανέβασε την συγκεκριμένη φωτογραφία, αλλά και

το Category_ID για να γνωρίζουμε σε ποιά κατηγορία ανήκει .Τέλος όλα τα υπόλοιπα πεδία έχουν οριστή ως χαρακτήρες εκτός από το πεδίο AddDate που παίρνει ημερομηνία.

- Πίνακας Votes

	Column Name	Data Type	Allow Nulls
▶	VoteID	int	<input type="checkbox"/>
	UserID	uniqueidentifier	<input checked="" type="checkbox"/>
	ImageID	nvarchar(50)	<input checked="" type="checkbox"/>
	Grade	int	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Στην σελίδα μας διαθέτουμε την δυνατότητα στον χρήστη να ψηφίζει την κάθε φωτογραφία που επιθυμεί και να δίνει τον βαθμό που θέλει. Έτσι οι ψηφοί αυτοί θα πρέπει να κρατηθούν σε αυτόν τον πίνακα. Για να γνωρίζουμε ποιος ψήφισε κρατάμε το UserID του χρήστη καθώς και το ImageID για να ξέρουμε ποια φωτογραφία ψήφισε. Τέλος διαθέτουμε ένα πεδίο που παίρνει ακέραιο αριθμό και αυξάνεται κάθε φορά με την προσθήκη νέας ψήφου αλλά και ένα πεδίο Grade που κρατάει τον βαθμό που δώσαμε στην φωτογραφία.

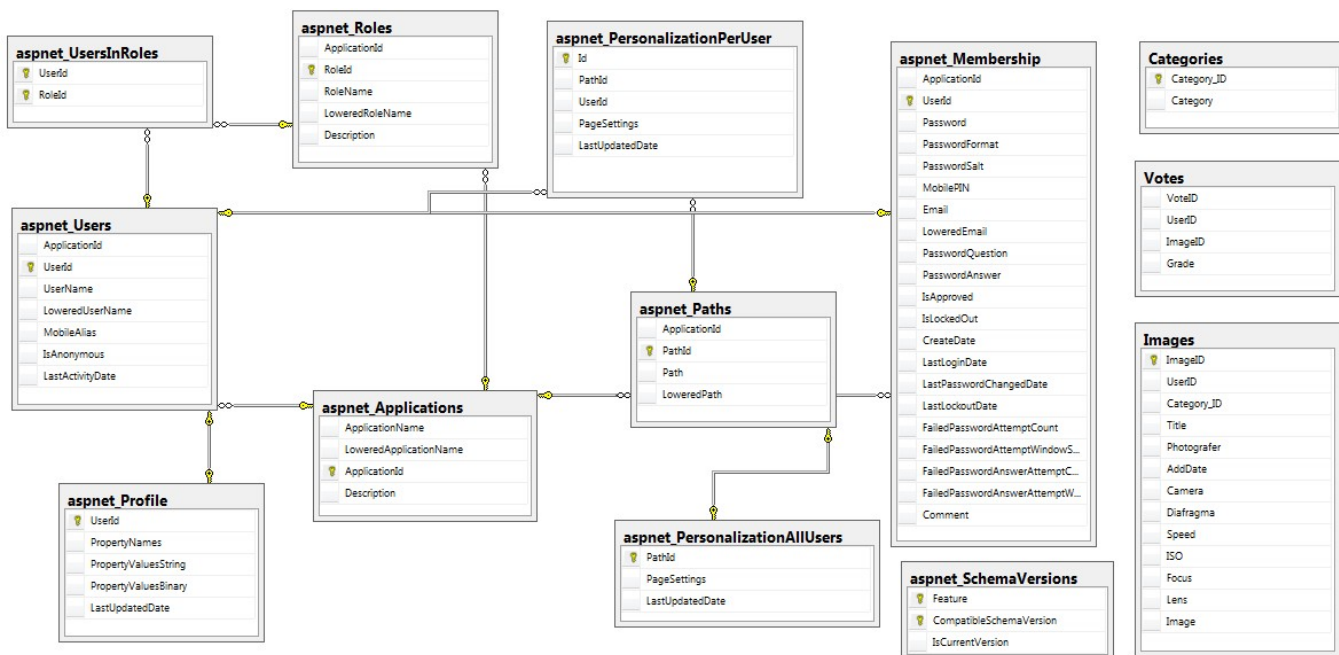
3.2 Διάγραμμα βάσης δεδομένων – Συσχετίσεις πινάκων

Το διάγραμμα της βάσης δεδομένων είναι μια γραφική αναπαράσταση των πινάκων και των μεταξύ τους σχέσεων. Η σχεδίασή του είναι ένας καλός τρόπος για να έχουμε μια εικόνα της δομής και των συσχετίσεων της βάσης δεδομένων. Εκτός αυτού είναι ένας βολικός τρόπος τεκμηρίωσης της βάσης, επειδή εκεί απεικονίζονται αυτόματα όποιες αλλαγές κάνουμε.

Στο παρακάτω σχήμα φαίνεται το διάγραμμα της βάσης δεδομένων. Μπορούμε να παρατηρήσουμε τις σχέσεις μεταξύ των πινάκων. Αρχικά παρατηρούμε ότι όλοι οι πίνακες που δημιουργήθηκαν με την Wizard του IDE συνδέονται μεταξύ τους. Ο aspnet_UsersInRole κρατάει τους χρήστες που ανήκουν σε ρόλους, έτσι είναι λογικό να συνδέεται με τους πίνακες aspnet_Roles και aspnet_Users με σχέση ένα προς πολλά.

Ο `aspnet_Roles` συνδέεται και με τον πίνακα `aspnet_Applications` έτσι ώστε να γνωρίζει σε ποια εφαρμογή ανήκουν οι ρόλοι.

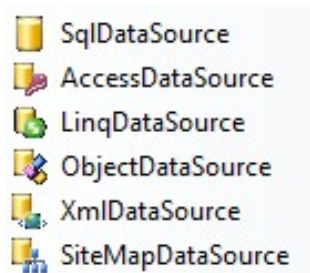
Ο `aspnet_Users` συνδέεται με σχέση 1 προς 1 με τον πίνακα `aspnet_Membership` όπου αυτός ο πίνακας κρατάει περισσότερες πληροφορίες για τους χρήστες μας. Ακόμη συνδέεται και με τον πίνακα `aspnet_Profile` αλλά και με τον `aspnet_Applications`. Τέλος οι πίνακες που φτιάχτηκαν για να κρατούν τις φωτογραφίες μας δεν συνδέονται μεταξύ τους καθώς δεν ήταν αναγκαίο να υπάρχει.



Σχήμα 3.2.1 Διάγραμμα βάσης δεδομένων

3.3 Έτοιμες Διεργασίες - Stored Procedure

Αν και είναι ποιο ορθό να χρησιμοποιούμε stored procedure για να παίρνουμε από την βάση μας τα επιθυμητά δεδομένα, εμείς δεν τις χρησιμοποιήσαμε αλλά χρησιμοποιήσαμε έτοιμα control που δίνει η Visual Studio για να κάνουν αυτή τη δουλειά. Τα control που χρησιμοποιούνται για αυτήν τη δουλειά απεικονίζονται παρακάτω:



Έτσι το μόνο που μένει σε εμάς να κάνουμε είναι να κάνουμε «drag and drop» έναν από τα εργαλεία αυτά που χρειαζόμαστε, και ύστερα να ορίσουμε στα έτοιμα παράθυρα που εμφανίζονται από πού θέλουμε να πάρουμε τα δεδομένα. Με την διαδικασία αυτή φτιάχνεται αυτόματα ο κώδικας που απαιτείται στην aspx σελίδα, όπως απεικονίζεται παρακάτω:

```
<asp:SqlDataSource ID="SqlDataSource1" runat="server" ConnectionString="<%= $ConnectionStrings:ListViewConnectionString %>"  
    SelectCommand="SELECT [Image], [Title], [Photografer], [ImageID] FROM [Images]">  
</asp:SqlDataSource>
```

Όπως βλέπουμε σε αυτή την εικόνα το control <asp:SqlDataSource> αρχίζει και τελειώνει με tags. Αρχικά δηλώνουμε το όνομα που θέλουμε να έχει το control αυτό, έτσι ώστε να μπορούμε όταν θέλουμε να το καλούμε από την σελίδα cs όπου εκεί γράφουμε των κώδικα για την κάθε σελίδα. Επειδή είναι control και δεν είναι γραμμένος ο κώδικας σε html θα πρέπει να τρέχει σε server, έτσι δίνουμε την εντολή runat=server.

Για να γνωρίζει από ποια βάση θα πάρει τα δεδομένα χρησιμοποιούμε το connectionString όπου δίνουμε το όνομα που διαθέτει το connectionString στο αρχείο Web.config. Έτσι το σύστημα πάει και βλέπει με ποια βάση συσχετίζεται το όνομα αυτό. Τέλος δίνουμε το selectCommand όπου γράφουμε σε γλώσσα sql τι θέλουμε να «τραβήξουμε» από την βάση μας.

3.4 Σύνδεση με Βάση Δεδομένων

Εκτός από το να χρησιμοποιήσουμε τα έτοιμα control που διαθέτει το Visual Studio μπορούμε για να τραβήξουμε δεδομένα από την βάση μας να χρησιμοποιήσουμε και την σελίδα cs, όπου απαιτεί να γνωρίζεις την γλώσσα C#. Ένα παράδειγμα κώδικα είναι το παρακάτω:

```

protected void ApproveImage_Click(object sender, ImageClickEventArgs e) {

    string connectionString = WebConfigurationManager.ConnectionStrings["ListViewConnectionString"].ConnectionString;
    SqlConnection myConnection = new SqlConnection(connectionString);
    string selectSQL = "Insert into Categories(Category) values(@Category)";
    try {
        myConnection.Open();
        Label1.Text = myConnection.State.ToString();
        SqlCommand sqlCommand = new SqlCommand(selectSQL, myConnection);
        sqlCommand.Parameters.Add(new SqlParameter("@Category", SqlDbType.NVarChar, 50));
        sqlCommand.Parameters["@Category"].Value=TxtNewCategory.Text;
        sqlCommand.ExecuteNonQuery();
        Label1.Text="Η καταχώρηση έγινε";
    }
    catch (Exception err) {
        Label1.Text="Error on page";
    }
    finally {
        int count = DdlCategory.Items.Count;
        count= count - 1;
        myConnection.Dispose();
        DdlCategory.Items.Insert(count, TxtNewCategory.Text);
        ApproveImage.Visible = false;
        CancelImage.Visible = false;
        TxtNewCategory.Visible = false;
    }
}

```

Όπως βλέπουμε στην παραπάνω εικόνα δημιουργούμε αρχικά έναν χαρακτήρα και κρατάμε σε αυτών το connection string που αφορά την σύνδεση μας με την βάση, όπου το παίρνουμε από το αρχείο Web.config. Ύστερα φτιάχνουμε ένα object SqlConnection το οποίο αφορά την συνδεσή μας με την βάση και δίνουμε ως παράμετρο το connection string που πήραμε από το αρχείο Web.Config.

Αφού περάσουμε το select statement που θέλουμε και το κρατήσουμε σε ένα string, τότε ανοίγουμε ένα try catch finally. Αυτό μας βοηθάει στο να μπορούμε να πιάσουμε τα λάθη που θα προκύψουν κατά την διάρκεια εκτέλεσης της εφαρμογής έτσι ώστε να μπορούμε να εντοπίσουμε εύκολα το λάθος μέσα στον κώδικα .

Μέσα εκεί ανοίγουμε την σύνδεση μας με την βάση και στο αντικείμενο SqlCommand περνάμε το select που έχουμε, καθώς και το connection με την βάση .Έτσι ως παράμετρο στο SqlCommand περνάμε εμείς την κάθε κατηγορία που έχουμε από το dropdownList.

Μέσα στο catch εμείς δηλώνουμε το τι θέλουμε να μας εμφανίσει έτσι ώστε να αντιληφτούμε από πού προήλθε το λάθος μας. Τέλος μέσα στο finally κλείνουμε το connection που έχουμε με την βάση καθώς και σώσουμε τα δεδομένα σε αυτήν.

4. Κεφάλαιο – Ανάλυση της εφαρμογής μας

4.1 Η φόρμα Master Page



Η Visual Studio μας δίνει την δυνατότητα να φτιάχνουμε Master Pages σε ASP.NET. Αποτελεί ένα πολύ χρήσιμο εργαλείο καθώς μπορούμε να φτιάχνουμε μια σελίδα την οποία θα μπορούμε να εισάγουμε κάθε φορά και σε άλλες σελίδες. Έτσι μας βγάζει από τον κόπο να φτιάχνουμε τα ίδια πράγματα για τις διαφορετικές σελίδες μας.

Στην DigitalGallery.Master εμείς φτιάξαμε εκείνα τα οποία θα παραμείνουν σε όλο το site μας και έτσι θα χρειαστεί απλά να κληρονομούμαι αυτήν στις νέες σελίδες. Αρχικά έχουμε προσθέσει πάνω στο site μια εικόνα που αποτελεί και το logo της εταιρείας. Πιο κάτω υπάρχει ένα control το asp:Menu το οποίο μας δίνει την δυνατότητα να φτιάξουμε εύκολα και γρήγορα το menu μας. Ύστερα προσθέτουμε το control asp:Login που αφορά την είσοδο του κάθε χρήστη.

Το εργαλείο αυτό μας βοηθάει να ελέγχουμε την πρόσβαση των χρηστών στο site μας. Ο κάθε χρήστης για να μπορέσει να ενεργεί και αυτός στο site θα πρέπει πρώτα να κάνει εγγραφή.

Οι δυνατότητες που θα διαθέτει ο κάθε εγγεγραμμένος χρήστης είναι η παρακάτω:

- Προσθήκη μιας φωτογραφίας
- Κατέβασμα επιθυμητής φωτογραφίας.
- Δικαίωμα ψήφου σε κάθε φωτογραφία που επιθυμεί.

Έτσι αυτές οι δυνατότητες κάνουν τον εγγεγραμμένο χρήστη να διαφέρει από τον απλό χρήστη. Όμως μπορεί ένας χρήστης να μην διαθέτει όλα τα παραπάνω καθώς αυτό εξαρτάται από τον χειριστή της σελίδας. Δηλαδή στην εφαρμογή μας έχουμε προσθέσει ρόλους, και ο κάθε χρήστης ανήκει σε κάποιον ρόλο, όπου αυτοί θα καθορίζουν τις δυνατότητες που θα έχει ο χρήστης μας. Οι ρόλοι που διαθέτει το σύστημα μας είναι:

- `downloadImage`
- `uploadImage`

Με αυτήν τη δυνατότητα καταφέρνουμε να ξεχωρίσουμε και τους εγγεγραμμένους χρήστες.

Ακόμη στην εφαρμογή μας έχουμε εισάγει και κατηγορίες έτσι ώστε να μπορεί ο επισκέπτης να βλέπει συγκεκριμένες φωτογραφίες. Τις κατηγορίες αυτές η εφαρμογή μας της παίρνει από την βάση δεδομένων. Για να το πετύχουμε αυτό χρησιμοποιούμε το εργαλείο `asp:DataList`, στο οποίο ορίζουμε το τι θα φέρουμε και από πού. Έτσι αρκεί να χρησιμοποιήσουμε ένα νέο control το `asp:SqlDataSource` στο οποίο ορίζουμε την βάση που θέλουμε να πάρουμε τα δεδομένα, αλλά και το τι θέλουμε να πάρουμε. Άρα αρκεί να συνδέσουμε το `asp:SqlDataSource` με το `asp:DataList`. Ο εγγεγραμμένος χρήστης διαθέτει και την δυνατότητα να προσθέσει και μια κατηγορία που επιθυμεί ο ίδιος.

Για να φέρουμε τις καλύτερες φωτογραφίες χρησιμοποιήσαμε τα ίδια ακριβώς εργαλεία με πριν. Το μόνο που άλλαξε είναι ότι διαλέξαμε διαφορετικό πίνακα από την βάση μας. Εμείς θέλουμε να μας φέρει από τον πίνακα `Votes` τις πρώτες 9 φωτογραφίες που έχουν το μεγαλύτερο άθροισμα ψήφων και να τις κάνει `order by` σύμφωνα με το άθροισμα που έχουν.

Τέλος σε αυτήν τη φόρμα διαθέτουμε και ένα εργαλείο το `asp:LoginView` στο οποίο μπορούμε να εμφανίσουμε διάφορα μηνύματα, έτσι ώστε να ξεχωρίζουμε εύκολα τις καταστάσεις, δηλαδή πότε έχει συνδεθεί ο χρήστης ή όχι.

4.2 Η φόρμα homePage



Η σελίδα που βλέπουμε αποτελεί την πρώτη σελίδα που εμφανίζεται στον χρήστη εάν ζητήσει να δει το site μας. Παρατηρούμε ότι δανείζεται όλα τα στοιχεία που διαθέτει η MasterPage μας. Έτσι εδώ το μόνο που έχουμε προσθέσει είναι η προβολή τυχαίων εικόνων.

Για να το πετύχουμε αυτό και να εμφανίζουμε τις διάφορες εικόνες που διαθέτει η βάση δεδομένων χρησιμοποιούμε ένα εργαλείο το `asp:ListView` το οποίο μας εμφανίζει τις φωτογραφίες σε μια συγκεκριμένη διάταξη. Έτσι με την προσθήκη αυτού του εργαλείου στην σελίδα μας προστίθεται ο παρακάτω κώδικας:

```

<asp:ListView ID="ListViewHomePage" runat="server" GroupItemCount="4">
  <GroupSeparatorTemplate>
    <tr id="lineSeperator" runat="server">
      <td colspan="4">
        <hr />
      </td>
    </tr>
  </GroupSeparatorTemplate>
  <LayoutTemplate>
    <table id="Table1" runat="server">
      <tr id="Tr1" runat="server">
        <td id="Td1" runat="server">
          <table id="groupPlaceholderContainer" runat="server">
            <tr id="groupPlaceholder" runat="server">
            </tr>
          </table>
        </td>
      </tr>
      <tr id="Tr2" runat="server">
        <td id="Td2" runat="server">
        </td>
      </tr>
    </table>
  </LayoutTemplate>

```

Εδώ δεν παρουσιάζεται όλος ο κώδικας που αφορά το control ListView αλλά το πρώτο του τμήμα. Αρχικά δηλώνουμε το όνομα του εργαλείου που θέλουμε να έχει μέσα στον κώδικα και ύστερα το ανά πόσα θέλουμε να εμφανίζονται τα αντικείμενα μας. Έτσι μας εμφανίζει τις φωτογραφίες ανά τετράδες. Ύστερα προσθέτουμε διαχωριστή στις φωτογραφίες μας δηλαδή ανά κάθε γραμμή 4 φωτογραφιών να υπάρχει μία διαχωριστική γραμμή. Τέλος καθορίζουμε το πώς θα εμφανίζεται συνολικά το control αυτό.

Ακόμη ο κώδικας που απομένει για να ολοκληρωθεί η παρουσίαση του ListView είναι ο παρακάτω:

```

<ItemTemplate>
  <td class="listview" align="left">
    <asp:HyperLink ID="HomePagehbl" Font-Underline="false" ForeColor="Black"
    NavigateUrl='<%# "~/viewImage.aspx?imgid=" + Eval("ImageID") %>'
      runat="server">
      <img src='uploadFiles/<%#Eval("Image") %>' width="80"
      id="homePageImageView" height="80" alt="" /></asp:HyperLink>
    <br />
    <label class="registrationLabel"> Τίτλος:</label>
    <asp:Label runat="server" Text='<% #Eval("Title") %>' CssClass="registrationLabel"></asp:Label><br />
    <asp:Label CssClass="registrationLabel" Text="Φωτογράφος:" runat="server"></asp:Label>
    <asp:Label runat="server" Text='<%#Eval("Photografer") %>' CssClass="registrationLabel"></asp:Label><br />
    <asp:ImageButton runat="server" Visible="false" ImageUrl="~/images/download_button.jpg" ID="downloadIcon"
      CommandArgument='<%#Eval("Image") %>' OnCommand="downloadImageFunction" />
  </td>
</ItemTemplate>
</GroupTemplate>
</asp:ListView>

```

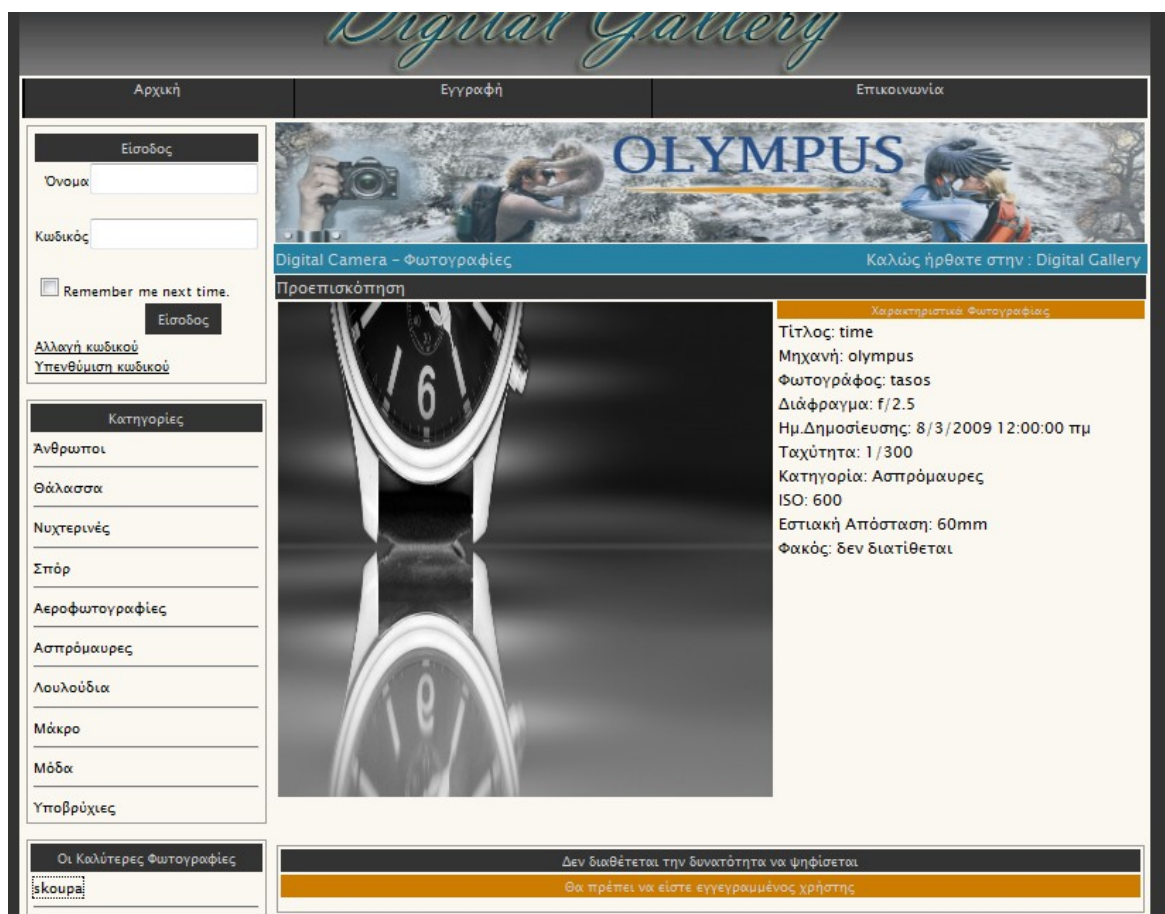
Μέσα στα άγκιστρα ItemTemplate θα πρέπει να δηλώσουμε το τι θέλουμε να εμφανιστεί ως αντικείμενο μέσα στο ListView. Ως αρχικό θα εμφανίσουμε την εικόνα που παίρνουμε από την βάση μας και βέβαια σε μικρότερη διάσταση από την πραγματική. Αυτή η εικόνα αποτελεί και link όπου μας οδηγεί σε μια καινούργια σελίδα την viewImage.aspx.

Ύστερα θέλουμε να εμφανίσουμε τον τίτλο που έχει δώσει στην φωτογραφία ο χρήστης που την ανέβασε. Έτσι αρκεί σε αυτό το ListView να έχουμε συνδέσει και ένα asp:SqlDataSource στο οποίο ζητάμε να μας φέρει τα στοιχεία της φωτογραφίας. Ακόμη ζητάμε να μας φέρει και το όνομα του φωτογράφου έτσι ώστε να γνωρίσουμε ποιος έχει ανεβάσει αυτήν την εικόνα.

Υπάρχει και ένα ακόμη image το οποίο αρχικά δεν είναι ορατό. Για να δει ο χρήστης το εικονίδιο αυτό, θα πρέπει να έχει κάνει εγγραφή στο site αλλά και να ανήκει στον ρόλο downLoadImage, έτσι θα είναι εμφανές, και θα του δίνει την δυνατότητα να κατεβάσει την συγκεκριμένη εικόνα στον υπολογιστή του.

Όμως εδώ ο χρήστης μπορεί να δει όλες τις εικόνες που υπάρχουν. Για να υπάρχει μετάβαση και να μπορούμε να δούμε τις υπόλοιπες θα πρέπει να συνδέσουμε το ListView control με το asp:DataPage το οποίο χωρίζει σε σελίδες ανάλογα το μέγεθος του PageSize που έχουμε δώσει. Εμείς εδώ έχουμε ορίσει ως μέγεθος 16.

4.3 Η φόρμα viewImage



Αρχικά για να πάμε στην σελίδα viewImage θα πρέπει να διαλέξουμε μία από τις εικόνες, που μας εμφανίζονται στην σειρά στην σελίδα homepage. Αφού επιλέξουμε μας στέλνει στην σελίδα αυτή.

Στην σελίδα αυτή μας παρουσιάζεται η εικόνα σε μεγαλύτερο μέγεθος και δίπλα εμφανίζονται τα χαρακτηριστικά που είχε η φωτογραφική την ώρα της λήψης. Τα χαρακτηριστικά αυτά είναι αναγκαία για οποιονδήποτε χρήστη ανεβάζει φωτογραφία.

Για κάθε φωτογραφία κρατάμε τα στοιχεία της λήψης στην βάση δεδομένων και όταν θέλουμε να τα εμφανίσουμε αρκεί να κάνουμε ένα select στην βάση και να ζητήσουμε τα συγκεκριμένα στοιχεία. Αυτά εμφανίζονται σε όλους τους χρήστες είτε είναι εγγεγραμμένοι είτε όχι.

Όμως για να διαθέτη κάποιος την δυνατότητα να μπορεί να ψηφίσει θα πρέπει να έχει κάνει εγγραφή στο σύστημα. Έτσι εάν είναι εγγεγραμμένος θα του εμφανιστή η σελίδα όπως παρακάτω:

The screenshot shows a web application interface for a digital camera gallery. On the left, there is a user login section with fields for 'Όνομα' (Name) and 'Κωδικός' (Code), a 'Remember me next time.' checkbox, and an 'Είσοδος' (Login) button. Below this are links for 'Αλλαγή κωδικού' (Change code) and 'Υπενθύμιση κωδικού' (Remind code). A 'Κατηγορίες' (Categories) section lists various categories like 'Ανθρωποι', 'Θάλασσα', etc. Below that is a section for 'Οι Καλύτερες Φωτογραφίες' (The Best Photos) with a list of users: 'chez_paul', 'skoupa', 'gun', and 'bienni'. The main content area features a large image of a watch with the number '6' on the dial. To the right of the image is a sidebar with technical specifications for the photo, including title, camera model, lens, aperture, shutter speed, ISO, and focal length. At the bottom of the main area, there is a voting interface titled 'Ψηφίστε την Φωτογραφία' (Vote for the Photo) with a row of 10 radio buttons and a 'Ψηφίστε' (Vote) button.

Εδώ βλέπουμε ότι μας εμφανίζεται ένας πίνακας όπου μας δίνει την δυνατότητα να ψηφίσουμε την συγκεκριμένη φωτογραφία. Υπάρχουν αριθμημένα radiobutton όπου μπορούμε να επιλέξουμε τον αριθμό που θέλουμε, των αριθμό που θα θέλαμε να δώσουμε στην συγκεκριμένη φωτογραφία με άριστα το 10. Μας δίνει την δυνατότητα να ψηφίσουμε μόνο μια φορά την κάθε φωτογραφία καθώς εάν προσπαθήσουμε ξανά να δούμε αυτήν την φωτογραφία, αφού έχουμε ψηφίσει, θα δούμε τον βαθμό που έχουμε δώσει.

Με την δυνατότητα που έχουμε να ψηφίσουμε την κάθε φωτογραφία παρουσιάζεται αριστερά στην σελίδα μας και ένας πίνακας που μας δείχνει τις καλύτερες φωτογραφίες. Το ποια φωτογραφία είναι η καλύτερη προκύπτει από το άθροισμα όλων των ψήφων, που έχουν δώσει οι χρήστες μας για την συγκεκριμένη φωτογραφία. Έτσι σε αυτήν τη στήλη παρουσιάζονται οι εννέα καλύτερες σύμφωνα με τις προτιμήσεις των χρηστών.

4.4 Η φόρμα upLoadImage

Αρχική Ανέβασμα φωτογραφίας Εγγραφή Επικοινωνία

Είσοδος
Όνομα
Κωδικός
 Remember me next time.
Είσοδος
[Αλλαγή κωδικού](#)
[Υπενθύμιση κωδικού](#)

Κατηγορίες
Ανθρωποι
Θάλασσα
Νυχτερινές
Σπόρ
Αεροφωτογραφίες
Ασπρόμαυρες
Λουλούδια

OLYMPUS
Digital Camera - Φωτογραφίες Καλώς ήρθες user5 [Αποσύνδεση](#)

Ανέβασμα Φωτογραφίας

Γενικές Πληροφορίες Τεχνικές Πληροφορίες

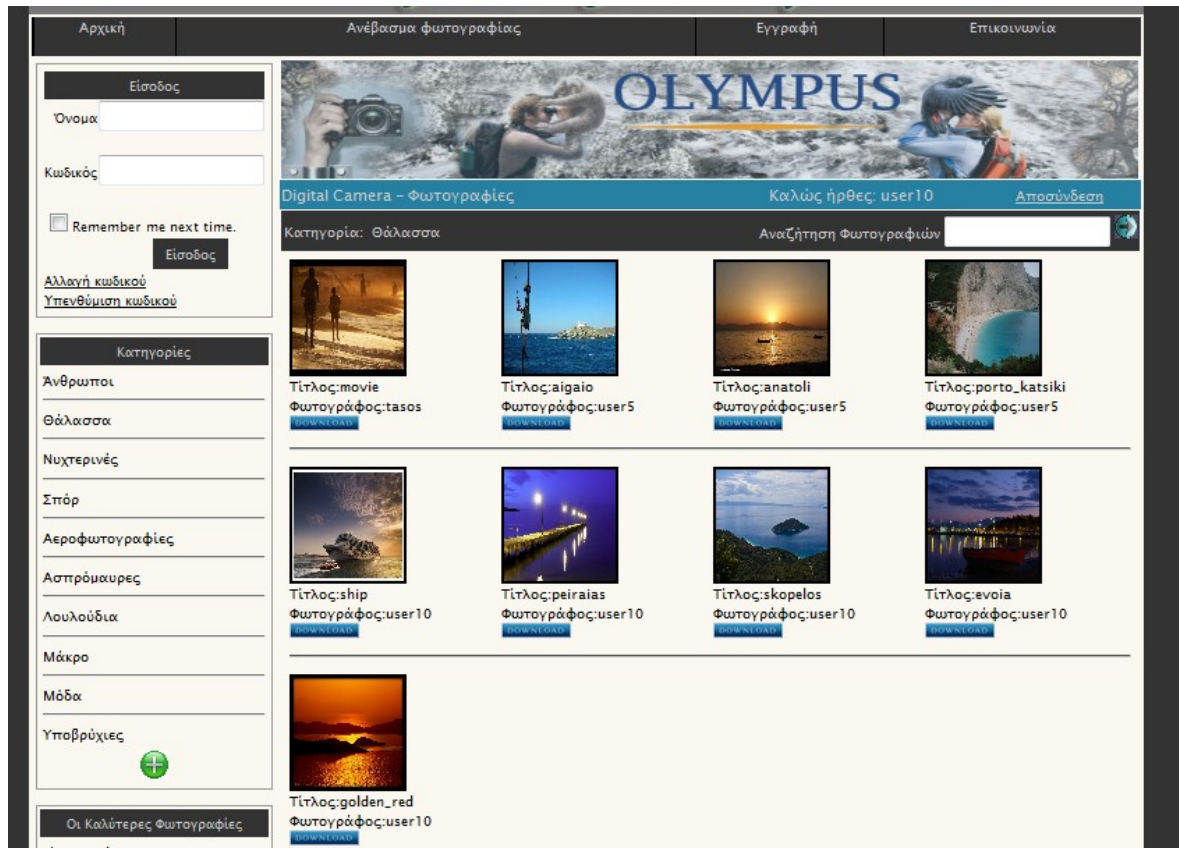
Τίτλος
Φωτογράφος user5
Ημ. Δημοσίευσης 11/4/2009 12:00:00
Κατηγορία Επιλέξτε Κατηγορία

Μηχανή
Διάφραγμα
Ταχύτητα
ISO
Εστιακή Απόσταση
Φακός
Φωτογραφία

Το παραπάνω interface που βλέπουμε αφορά το ανέβασμα της φωτογραφίας. Τη δυνατότητα αυτή δεν την διαθέτουν όλοι οι επισκέπτες του site, καθώς θα πρέπει να είναι εγγεγραμμένοι στο σύστημα αλλά και να ανήκουν στον ρόλο του upLoadImage.

Αρχικά θα πρέπει να συμπληρώσουν τα παραπάνω πεδία που αφορούν τα τεχνικά χαρακτηριστικά της φωτογραφίας και αφού τα συμπληρώσουν να ανεβάσουν την συγκεκριμένη εικόνα. Όμως υπάρχουν κάποιοι περιορισμοί στο μέγεθος της εικόνας αλλά και στον τύπο αρχείου. Θα πρέπει το μέγεθος της να μην ξεπερνά τα 4MB αλλά και το αρχείο να είναι της μορφής jpg η gif γιατί αλλιώς δεν θα μπορούσαμε να ανεβάσουμε την photo.

4.5 Η φόρμα viewCategory.aspx



Η σελίδα αυτή καλείται όταν επιλέξουμε μια από τις κατηγορίες που υπάρχουν στην αριστερή στήλη μας και δεν απαιτείται να είναι εγγεγραμμένος ο χρήστης. Εδώ εμείς επιλέξαμε την κατηγορία θάλασσα και μας έφερε όλες εκείνες που σχετίζονται με την θάλασσα. Πάνω στην μπάρα που υπάρχει και η μηχανή αναζήτησης μας εμφανίζει και την κατηγορία που επιλέξαμε έτσι ώστε να είναι πιο εύκολα αντιληπτό. Για να το πετύχουμε αυτό χρησιμοποιήσαμε έναν `SqlDataSource` όπου μας φέρνει την κατηγορία που επιλέξαμε από την στήλη Κατηγορίες, και για να την εμφανίσουμε χρησιμοποιούμε το control `DataList`.

4.6 Η φόρμα rememberPassword.aspx

Αρχική Εγγραφή Επικοινωνία

Είσοδος

Όνομα

Κωδικός

Remember me next time.

Είσοδος

[Αλλαγή κωδικού](#)

[Υπενθύμιση κωδικού](#)

Κατηγορίες

OLYMPUS

Digital Camera - Φωτογραφίες Καλώς ήρθατε στην : Digital Gallery

Υπενθύμιση κωδικού πρόσβασης

Εισάγεται όνομα χρήστη για να λάβετε κωδικό πρόσβασης

Όνομα χρήστη *

Submit

Ο λόγος που δημιουργήθηκε αυτή η σελίδα, είναι για να δίνει την δυνατότητα στον εγγεγραμμένο χρήστη να του υπενθύμιση ποιος είναι ο κωδικός πρόσβασης που έχει ορίσει στο σύστημα. Η υπενθύμιση του κωδικού πρόσβασης γίνεται με την αποστολή ενός ηλεκτρονικού μηνύματος από τον mail server. Επειδή όμως τα Vista δεν διαθέτουν έναν Default SMTP όπως διαθέτουν τα XP για αυτό και κατεβάσαμε έναν Free SMTP Server ο οποίος μας δίνει την δυνατότητα να στείλουμε 10 mail την ημέρα.

Η λειτουργία του είναι απλή καθώς το μόνο που απαιτείται είναι μια απλή εγκατάσταση στον υπολογιστή μας, και όταν θελήσουμε να στείλουμε mail αρκεί να ανοίξουμε το πρόγραμμα.

Για να στείλουμε ένα ηλεκτρονικό μήνυμα μέσω της εφαρμογής μας θα πρέπει αρχικά να επιλέξουμε στο control asp:Login την υπενθύμιση κωδικού και ύστερα να συμπληρώσουμε στην φόρμα που εμφανίζεται το όνομα του χρήστη που έχουμε δώσει. Ύστερα εάν το όνομα που δώσαμε υπάρχει στην βάση δεδομένων μας ζητάει να δώσουμε μια απάντηση στην ερώτηση που έχουμε ορίσει όταν δημιουργήσαμε τον λογαριασμό αυτό. Αφού κάνουμε αυτή την διαδικασία και ανοίξουμε το πρόγραμμα Free SMTP Server μπορούμε να πατήσουμε Submit και ένα νέο μήνυμα θα έρθει στο ηλεκτρονικό ταχυδρομείο που είχαμε δηλώσει κατά την διάρκεια της εγγραφής μας.

4.7 Η φόρμα registration.aspx

Αρχική Εγγραφή Επικοινωνία

Είσοδος

Όνομα

Κωδικός

Remember me next time.

Είσοδος

[Αλλαγή κωδικού](#)

[Υπενθύμιση κωδικού](#)

Κατηγορίες

Ανθρωποι

Θάλασσα

Νυχτερινές

Σπόρ

Αεροφωτογραφίες

OLYMPUS

Digital Camera - Φωτογραφίες Καλώς ήρθατε στην : Digital Gallery

Εγγραφή Χρήστη

Δημιουργία χρήστη

Όνομα Χρήστη*

Κωδικός*

Επιβεβαίωση Κωδικού*

E-mail*

Ερώτηση Ασφαλείας

Η απάντησή σας

Δημιουργία χρήστη

Όλοι οι μη εγγεγραμμένοι χρήστες έχουν την δυνατότητα να κάνουν εγγραφή στο σύστημα. Όμως υπάρχουν περιορισμοί στα στοιχεία που θα δηλώσει ο χρήστης. Θα πρέπει να δώσει ένα όνομα χρήστη που δεν χρησιμοποιείται από κάποιον άλλον γιατί αλλιώς θα του εμφανίσει μήνυμα λάθους και δεν θα ολοκληρώσει την διαδικασία.

Ακόμη θα πρέπει ο κωδικός να είναι ίδιος με αυτόν που έδωσε και στο πεδίο κωδικός, αλλά και στο πεδίο επιβεβαίωση κωδικού. Το πεδίο email θα πρέπει και αυτό να είναι μοναδικό δηλαδή το mail που θα δώσουμε να μην χρησιμοποιείται από κάποιον άλλον εγγεγραμμένο χρήστη. Θα πρέπει να δώσουμε μια ερώτηση ασφαλείας και να θυμόμαστε τη απάντηση έχουμε δώσει καθώς θα μας χρειαστεί στην υπενθύμιση κωδικού. Αφού συμπληρώσουμε τα πεδία και πατήσουμε Δημιουργία χρήστη μας εμφανίζεται ένα ListBox έτσι ώστε να επιλέξουμε σε ποιον από τους παρακάτω ρόλους θέλουμε να ανήκουμε.

Έτσι ολοκληρώνεται η διαδικασία της εγγραφής και όλα τα στοιχεία που δώσαμε περνάνε στην βάση δεδομένων.

4.8 Η φόρμα communication.aspx

Είσοδος

Όνομα

Κωδικός

Remember me next time.

[Αλλαγή κωδικού](#)
[Υπενθύμιση κωδικού](#)

Κατηγορίες

Ανθρωποι

Θάλασσα

Νυχτερινές

Σπόρ

Αεροφωτογραφίες

Ασπρόμαυρες

Λουλούδια

Μάκρο

Μόδα

OLYMPUS

Digital Camera - Φωτογραφίες Καλώς ήρθατε στην : Digital Gallery

Επικοινωνία χρήστη

Ταχυδρομική διεύθυνση: Φοτητικές Εστίες Ηρακλείου
Τηλέφωνο: 2810...

Μη διστάσετε να επικοινωνήσετε μαζί μας, για οποιοδήποτε θέμα, καθώς και για σχόλια ή υποδείξεις που αφορούν στην ιστοσελίδα μας, μέσω της παρακάτω ηλεκτρονικής φόρμας:

Από:

Πρός:

Θέμα:

Με την φόρμα communication ο κάθε χρήστης εγγεγραμμένος και μη, μπορεί να έρθει σε επικοινωνία με τους υπεύθυνους της διαδικτυακής εφαρμογής. Αυτό που θα πρέπει να συμπληρώσει είναι το ηλεκτρονικό ταχυδρομείο που διαθέτει ο ίδιος καθώς και το μήνυμα που θέλει να μεταφέρει. Το που θα αποσταλεί το μήνυμα είναι συμπληρωμένο και σχεδιασμένο έτσι ώστε να μην μπορεί να αλλάξει από τον χρήστη. Έτσι δίνουμε την δυνατότητα στο χρήστη να μας μεταφέρει την άποψη του για το site και να βελτιωθούμε ακόμη περισσότερο.

4.9 Η φόρμα changePassword.aspx

Σε αυτήν εδώ την φόρμα ο εγγεγραμμένος χρήστης μπορεί να αλλάξει τον κωδικό πρόσβασης που είχε δώσει αρχικά κατά την διάρκεια της εγγραφής του. Αρχικά θα πρέπει να δώσει τον σωστό κωδικό που χρησιμοποιούσε, και να δώσει στα πεδία νέος κωδικός και επιβεβαίωση κωδικού τον νέο κωδικό που θέλει να ορίσει. Εάν δεν συμπληρώσει σωστά κάποια από τα πεδία τότε θα του εμφανιστεί μήνυμα λάθους. Το Visual Studio 2008 για την αλλαγή του κωδικού μας έχει έτοιμο control που κάνει αυτή την δουλειά. Έτσι το μόνο που έχουμε να κάνουμε είναι να αλλάξουμε το όνομα των πεδίων αλλά και την εμφάνιση τους έτσι ώστε να ταιριάζει με την όλη εφαρμογή. Μετά την ολοκλήρωση της αλλαγής και της εγγραφής του νέου κωδικού στην βάση δεδομένων μας εμφανίζει ένα μήνυμα επιτυχίας.

5. Βιβλιογραφία – Πηγές

1. Beginning ASP .NET 3.5 in C# 2008 from Novice to Professional Second Edition, Matthew MacDonald.
2. O'Reilly.Head.First.C.Sharp.Nov.2007, Andrew Stellman & Jennifer Greene.
3. C# for Java Programmers , Brian Bagnall, Philip Chen, Stephen Goldberg, Jeremy Faircloth (Technical Reviewer), Harold Cabrera (Technical Editor).
4. C# - APress - Beginning ASP.NET 2.0 In C# 2005 From Novice To Professional - Matthew MacDonald – 2006.
- 5.Apress.Beginning.HTML.with.CSS.and.XHTML.Modern.Guide.and.Reference.Jun.2007, David Schultz and Craig Cook.
6. Apress - Pro Asp Net 3.5 In C# 2008, Second Edition, Matthew MacDonald and Mario Szpuszta.
7. Manual - Wrox.Professional.ADO.dot.NET.2.Programming.with.SQL.Server.2005.Oracle.and.MySQL.Dec.2005.eBook-DDU
8. Expert Service-Oriented Architecture in C# 2005_ Defining Web services development with ASP.NET and WSE 3.0, Second Edition, Jeffrey Hassan with Mauricio Duran
9. Visual Studio 2008 - Introducing ASP.NET AJAX, Dino Esposito
10. For.Dummies.ASP.NET.2.0.Everyday.Apps.For.Dummies.Feb.2006, Doug Lowe
11. John.Wiley&Sons-SQL.for.Dummies,5th.Edition, Allen G.Taylor