

**ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΚΡΗΤΗΣ**  
**ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΠΟΛΥΜΕΣΩΝ**



**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**ΘΕΜΑ:** « Δυναμικό σύστημα αυτοματοποιημένης εγγραφής και έκδοσης βεβαιώσεων και πιστοποιητικών για το ΕΠΠ του ΤΕΙ Κρήτης. »

**Εισηγητής: Παχουλάκης Ιωάννης**

**Σπουδαστής: Λιατάκης Βασίλειος**

## **Ευχαριστίες**

Θα ήθελα 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 με στήριξε από όλες τις πλευρές.

Σας ευχαριστώ,  
*Λιατάκης Βασίλης*

<b>1. Κεφάλαιο – Το λογισμικό πακέτο</b>	<b>5</b>
1.1 Συνοπτική περιγραφή του συστήματος	5
1.2 Απαιτήσεις συστήματος	5
<b>2. Κεφάλαιο - Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης</b>	<b>7</b>
2.1 Πλατφόρμα .NET	7
2.1.1 Αρχιτεκτονική .NET	7
2.1.2 Visual Basic.NET	10
2.1.3 Web Forms και ASP.NET	13
2.2 Βάσεις δεδομένων και SQL Server 2000	16
<b>3. Κεφάλαιο – Περιγραφή της Βάσης Δεδομένων</b>	<b>17</b>
3.1 Πίνακες της Βάσης Δεδομένων	17
3.2 Διάγραμμα Βάσης Δεδομένων – Συσχετίσεις πινάκων	21
3.3 Αποθηκευμένες Διεργασίες – Stored Procedures	23
<b>4. Κεφάλαιο – Ανάλυση διαδικασιών σε ASP.NET</b>	<b>25</b>
4.1 Επικύρωση για της Web φόρμες	25
4.1.1 RequiredFieldValidator	26
4.1.2 RegularExpressionValidator	27
4.2 Ασφάλεια	28
4.2.1 Ασφάλεια του IIS	28
4.2.2 Το αρχείο web.config	28
4.3 Cookies	29
4.4 Σύνδεση με Βάση Δεδομένων	31

<b>5. Κεφάλαιο – Ανάλυση της εφαρμογής μας</b>	<b>33</b>
5.1 Η φόρμα Logon.aspx	33
5.2 Η φόρμα Main.aspx	34
5.3 Η φόρμα Results.aspx	35
5.4 Η φόρμα Registration_Main.aspx	36
5.5 Η φόρμα Registration_Confirm.aspx	37
5.6 Το Hyperlink <a href="http://eclass.epp.teiher.gr/eclass">http://eclass.epp.teiher.gr/eclass</a>	38
<b>6. Κεφάλαιο – Βιβλιογραφία – Πηγές</b>	<b>39</b>

## 1. Κεφάλαιο – Το λογισμικό πακέτο

### 1.1 Συνοπτική περιγραφή του συστήματος

Το αντικείμενο της παρούσας πτυχιακής εργασίας είναι ο σχεδιασμός και η υλοποίηση μιας εφαρμογής δυναμικού συστήματος αυτοματοποιημένης εγγραφής και αναλυτικής βαθμολογίας για το ΕΠΠ του ΤΕΙ Κρήτης. Η εφαρμογή έχει αναπτυχθεί σε τεχνολογία ASP.NET, με βάση δεδομένων σε SQL Server 2000.

Η βάση δεδομένων περιλαμβάνει όλα τα συστατικά που είναι απαραίτητα για την υλοποίηση ενός δυναμικού συστήματος αυτοματοποιημένης εγγραφής και αναλυτικής βαθμολογίας.

### 1.2 Απαιτήσεις Συστήματος

Οι απαιτούμενες τεχνικές προδιαγραφές της πλατφόρμας εγκατάστασης είναι οι ακόλουθες:

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

-  Microsoft Windows 2000 with Service Pack 3.0

-  Microsoft Windows XP


- ✓ Required Prerequisite Components:

-  Windows 2000 Service Pack 3


-  Microsoft FrontPage 2000 Web Extensions Client

-  Setup Runtime Files

-  Microsoft Internet Explorer 6 Service Pack 1

-  Microsoft Data Access Components 2.7 Service Pack 1 (Windows 2000 only)

-  Microsoft .NET Framework 1.1

-  Microsoft Visual J# .NET Redistributable Package 1.1

✓ Κοινές προδιαγραφές:

✚ Μνήμη: 128MB τουλάχιστον (προτεινόμενη 512MB)

✚ Επεξεργαστής: Pentium 133 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 ASP. NET που χρησιμοποιήθηκε είναι μια πλατφόρμα ανάπτυξης λογισμικού με πολλές δυνατότητες στις εφαρμογές δικτύου και μεγάλη ευκολία στη σύνδεση με διάφορες βάσεις δεδομένων. Εσωκλείει όλες τις τελευταίες τεχνολογίες στο δυναμικό προγραμματισμό. Οι εφαρμογές της είναι δυναμικές σελίδες που εκτελούνται στον Web Server και το αποτέλεσμα τους (σε html) εμφανίζεται στον Browser του χρήστη.

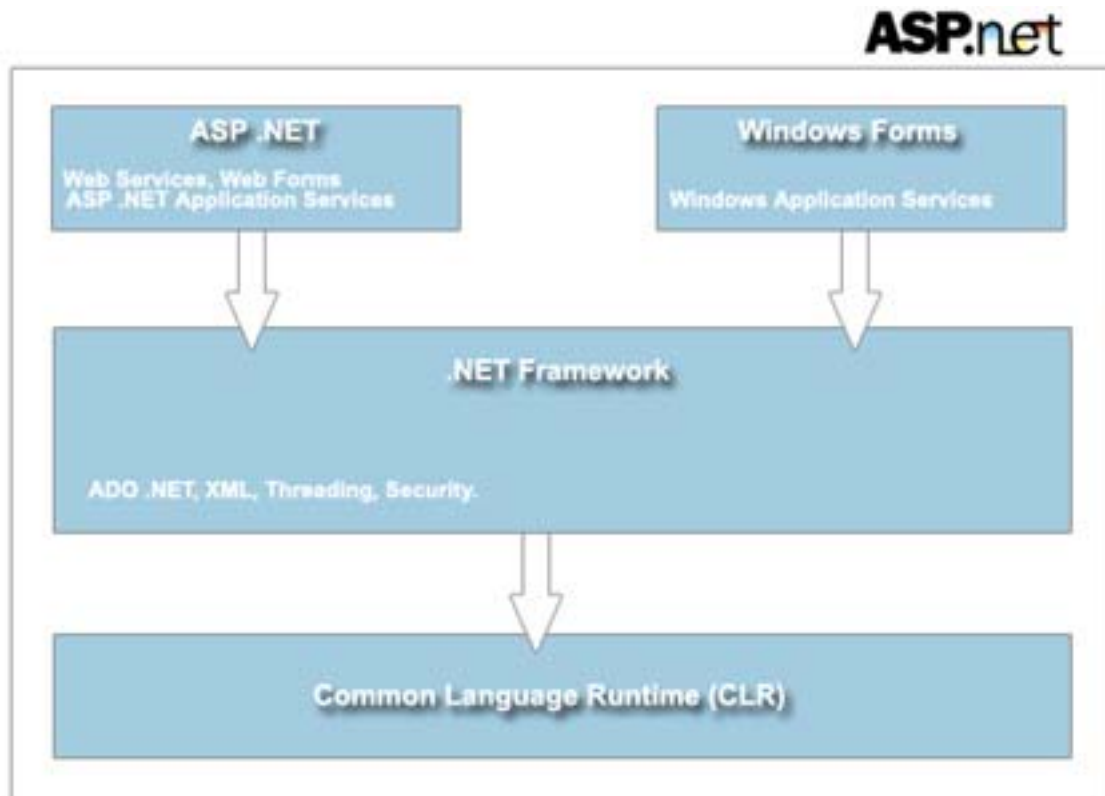
#### 2.1.1 Αρχιτεκτονική .NET

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

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

Τα πλεονεκτήματα αυτού του τρόπου εκτέλεσης των εφαρμογών είναι πολλά. Πρώτα από όλα προσφέρεται ένας ενιαίος τρόπος για τη χρήση των υπηρεσιών του λειτουργικού συστήματος, σε αντίθεση με την σημερινή κατάσταση, κατά την οποία μερικές υπηρεσίες βρίσκονται σε DLL και άλλες σε αρχεία τύπου COM. Αυτό σημαίνει επίσης ότι παύει η δυσχρηστία με τα DLL, δηλαδή το φαινόμενο κατά το οποίο η εγκατάσταση μιας εφαρμογής αντικαθιστά κάποιο DLL μιας άλλης εφαρμογής με απρόβλεπτα, όπως πάντα, αποτελέσματα. Στην πλατφόρμα .NET υπάρχει πια ο διαχωρισμός των στοιχείων της κάθε εφαρμογής, έτσι ώστε όταν αυτή εκτελείται, να φορτώνονται συγχρόνως και τα στοιχεία με τα οποία αναπτύχθηκε και ελέγχθηκε. Η δυνατότητα της εύκολης μεταφοράς μεταξύ διαφορετικών επεξεργαστών και λειτουργικών συστημάτων και η ευελιξία στον προγραμματισμό με τη χρήση πολλαπλών γλωσσών συμπληρώνουν με μια πρώτη ματιά τα πιο αξιόλογα πλεονεκτήματα της πλατφόρμας .NET. Σήμερα υπάρχουν αρκετές εκδόσεις των Windows, τα Windows 95, Windows 98, Windows 98 SE, Windows Millennium, Windows CE, Windows NT 4.0, Windows 2000, 64μπιτα Windows 2000, Windows XP, Windows Vista. Τα περισσότερα από αυτά τρέχουν σε επεξεργαστές x86. Η εξαίρεση αυτού του κανόνα έρχεται με τα Windows CE και τα 64μπιτα Windows. Υπάρχουν φυσικά και αρκετές νέες συσκευές στα άμεσα πλάνα των κατασκευαστών που αναμένονται να έχουν κάποιο, διαφορετικό ίσως, τρόπο επικοινωνίας με τον χρήστη. Θεωρητικά, μια εφαρμογή .NET μπορεί να εκτελείται σε λειτουργικά συστήματα διαφορετικά από τα Windows, αρκεί να υπάρχει ένα CLR για κάθε διαφορετική συσκευή, λύνοντας έτσι τα χέρια του προγραμματιστή.





Σχήμα 2.1.1.1 Αρχιτεκτονική .NET

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

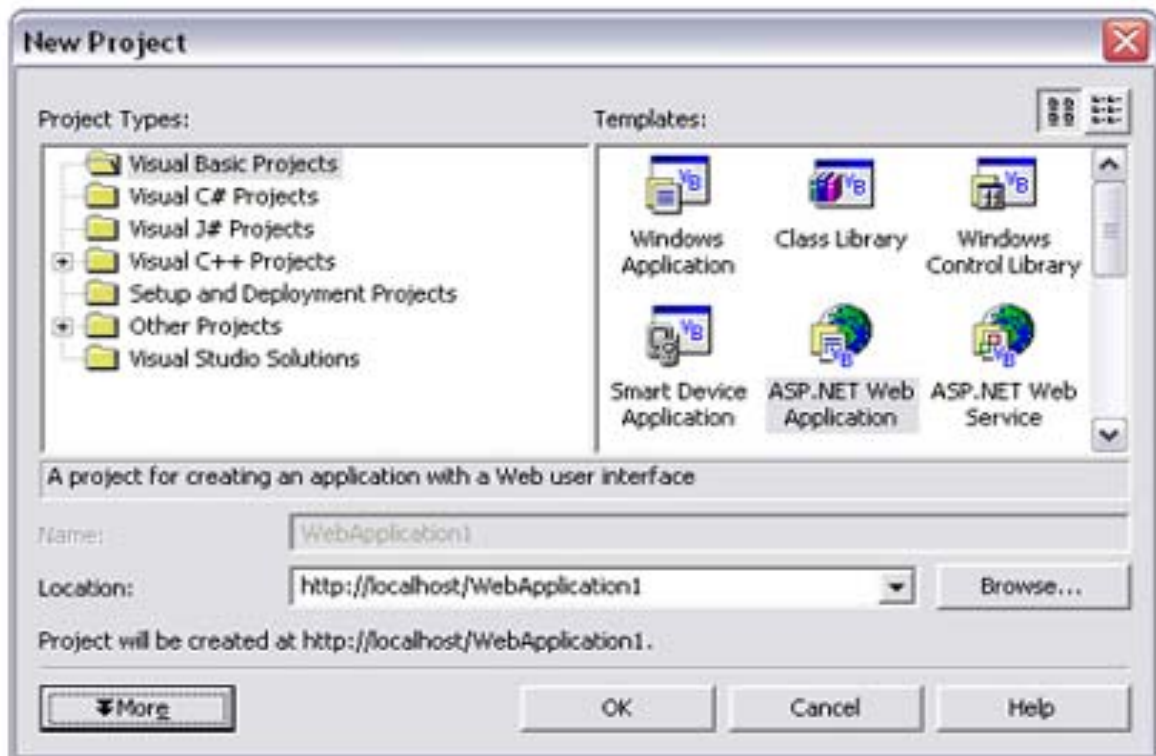
Η εγκατάσταση μιας εφαρμογής αποτελεί, σε όλες ανεξαιρέτως εκδόσεις των Windows, ένα από τα πιο δύστροπα σημεία. Συνήθως, ο κατασκευαστής συνοδεύει την εφαρμογή με αρκετά πρόσθετα αρχεία, εκτελέσιμα DLL και OCX, εγγραφές στο μητρώο και συντομεύσεις. Τα Windows 2000 έφεραν το Windows Installer που βελτίωσε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

### 2.1.2 Visual Basic .NET

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

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

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



**Σχήμα 2.1.2.1** Το Visual Studio απλοποιεί τις διαδικασίες χρησιμοποιώντας έτοιμα πρότυπα εφαρμογών.

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

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

Όπως θα έχετε καταλάβει, οι αλλαγές είναι πολλές και η μεταφορά από την έκδοση 6 απαιτεί την πληκτρολόγηση αρκετού κώδικα. Για αυτόν το λόγο η Microsoft ενσωμάτωσε

ένα βοηθητικό εργαλείο (Upgrade Tool), που αναλαμβάνει την αναβάθμιση της εφαρμογής στο νέο περιβάλλον, αλλάζοντας αυτόματα τις εντολές στην νέα μορφή. Αυτό το εργαλείο ενεργοποιείται αυτόματα αν ανοίξετε κάποια παλιότερη εφαρμογή μέσα από το Visual Studio .NET. Το αποτέλεσμα είναι να δημιουργηθεί ένα νέο έργο, αφήνοντας τα αρχικά αρχεία της έκδοσης 6.0 όπως έχουν. Δύσκολα όμως θα αποφύγετε την επέμβαση στον κώδικα, αφού τις περισσότερες φορές η αυτόματη αναβάθμιση θα δημιουργήσει αρκετές ασυμβατότητες.

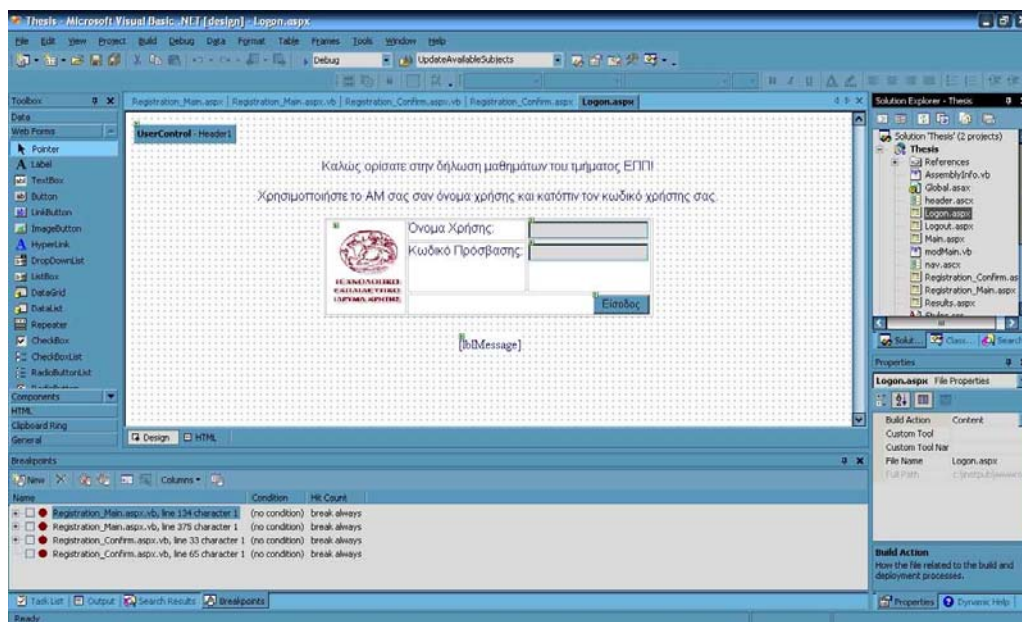
Ένα ακόμα ενδιαφέρον χαρακτηριστικό στις φόρμες της Visual Basic .NET είναι η μεγαλύτερη ακρίβεια. Χαρακτηριστικά όπως το Menu Designer, Control Anchoring, Control Docking και μια σειρά από νέα στοιχεία ελέγχου επιτρέπουν να αναπτύξει κανείς πιο καλαίσθητες φόρμες. Σε αυτό έρχεται να προστεθεί και η υποστήριξη του GDI+, του διαδόχου του GDI (Graphic Device Interface) που συναντούσαμε στις προηγούμενες εκδόσεις των Windows, το οποίο προσφέρει πιο πλούσια χαρακτηριστικά. Έτσι, μπορεί κανείς εύκολα να σχεδιάσει οθόνες που εμφανίζονται σε διαφανή και πολυεπίπεδα παράθυρα, χαρακτηριστικό που επιτρέπουν τα Windows 2000 και τα Windows XP. Άλλο ένα ενδιαφέρον στοιχείο της Visual Basic .NET είναι η πλήρης υποστήριξη του Unicode, ακόμα και στις φόρμες.

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

### 2.1.3 Web Forms και ASP .NET

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

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



Σχήμα 2.1.3.1 Ο σχεδιασμός μιας φόρμας (Web Form) για το Internet μοιάζει με αυτόν των φορμών της VB, με τον προγραμματιστή να έχει στη διάθεσή του τη γνωστή εργαλειοθήκη.

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

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

Η δημιουργία μιας απλής εφαρμογής, στην οποία ο τελικός χρήστης θα μπορεί να εισαγάγει το όνομα και τον κωδικό πρόσβασης, είναι υπόθεση λίγων λεπτών. Αρκεί να "τραβήξει" κανείς δύο αντικείμενα τύπου Label για τις περιγραφές, δύο άλλα αντικείμενα TextBox και ένα κουμπί εντολών. Το σκεπτικό που ακολουθούν οι φόρμες για το Web είναι το ίδιο όπως και στον κλασικό προγραμματισμό. Με διπλό κλικ, για παράδειγμα, πάνω στο κουμπί εντολών μπορείτε να γράψετε τον κώδικα που θέλετε και να αλλάξετε οποιοδήποτε από τα χαρακτηριστικά των αντικειμένων που βρίσκονται πάνω στη φόρμα. Αν χρησιμοποιήσετε ελληνικούς χαρακτήρες στις επιλογές, τα μενού και τα πεδία κειμένου, καλό είναι να επιλέξετε, με δεξί κλικ πάνω στη φόρμα, το ελληνικό σεντ χαρακτήρων.

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

Φυσικά, μπορείτε να γράψετε απλό κώδικα HTML και VB μέσα στην ίδια ιστοσελίδα από το FrontPage ή έναν άλλο editor, αρκεί να την αποθηκεύσετε σε ένα διακομιστή που διαθέτει την υποστήριξη του .NET Framework και το συγκεκριμένο αρχείο να έχει την κατάληξη .aspx. Δουλεύοντας με τη νέα δομή των ιστοσελίδων ASP

.NET, θα δείτε ότι υπάρχουν πολύ μεγάλες αλλαγές και για αυτό δεν ενδείκνυται καθόλου να προσπαθήσετε να μεταφέρετε τον παλιό σας κώδικα στη νέα μορφή. Αλλωστε, ένα διακομιστής Windows 2000 με το .NET Framework υποστηρίζει και τις δύο εκδόσεις.

## 2.2 Βάσεις Δεδομένων και SQL Server 2000

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

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

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

Το Microsoft SQL Server 2000 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

Η βάση δεδομένων αποτελείται από τους παρακάτω πίνακες:

##### 1. Πίνακας tblStudent

	Column Name	Data Type	Length	Allow Nulls
▶	Student_ID	int	4	
🔑	AM	int	4	
	First_Name	nvarchar	100	✓
	Last_Name	nvarchar	100	✓
	Password	varchar	50	

Σχήμα 3.1.1 Πίνακας tblStudent

Ο παραπάνω πίνακας (Σχήμα 3.1.1) είναι το πρώτο που διαβάζεται από το πρόγραμμα. Διαβάζεται όταν ένας εγγραμμένος χρήστης προσπαθεί να εισέρθει στο σύστημα, το σύστημα ελέγχει αν το AM του χρήστη συμπίπτει με το δηλωμένο κωδικό πρόσβασης και αν ναι του επιτρέπεται η πρόσβαση στο πρόγραμμα, εάν όχι ζητείται να βάλει τον σωστό κωδικό αλλιώς δεν επιτρέπεται η πρόσβαση στο σύστημα.

Δύο από τα παραπάνω πέντε πεδία χρησιμοποιούνται για το LOGON το AM και το Password το πρωτεύον κλειδί είναι το AM που βέβαια χρειάζεται να είναι μοναδικό στοιχείο σε κάθε περίπτωση.

## 2. Πίνακας tblSubject

	Column Name	Data Type	Length	Allow Nulls
▶	Subject_Code	nvarchar	50	
	Subject_Name	nvarchar	200	
	LabCode	nvarchar	50	✓
	Semester	int	4	
	IsTheory	bit	1	
	Theory_Hours	int	4	✓
	Blackboard_Hours	int	4	✓
	Lab_Hours	int	4	✓
	SubjectCategory	int	4	

Σχήμα 3.1.2 Πίνακας tblSubject

Ο παραπάνω πίνακας (Σχήμα 3.12) είναι από τους πιο βασικούς στην βάση δεδομένων, αποτελείται από εννιά πεδία, τα οποία περιγράφουν όλα τα μαθήματα της σχολής και της σχετικές τους λεπτομέρειες όπως: Κωδικός μαθήματος που είναι και το πρωτεύων κλειδί, Όνομα μαθήματος, Κωδικός Εργαστηρίου, Εξάμηνο, Είναι Θεωρία?, Ώρες Θεωρίας, Ώρες επί πίνακα, ώρες εργαστηρίου, και κατηγορία μαθήματος.

## 3. Πίνακας tblStudentSubject

	Column Name	Data Type	Length	Allow Nulls
▶	AM	int	4	
▶	Subject_Code	nvarchar	50	
	Grade	float	8	✓
	IsRegistered	bit	1	✓

Σχήμα 3.1.3 Πίνακας tblStudentSubject

Ο παραπάνω πίνακας (Σχήμα 3.1.3) περιέχει τέσσερα πεδία AM, Κωδικός μαθήματος, Βαθμολογία, και Εγγραμμένος? Σε αυτών των πίνακα φαίνετε οι αναλυτική βαθμολογία του κάθε φοιτητή επίσης εάν είναι δηλωμένο το μάθημα η όχι, οι επιλογές μέσου του προγράμματος για εγγραφή σε κάποιο μάθημα επιτρέπεται βεβαίως μόνο στην περίπτωση που δεν έχει ήδη περαστή το μάθημα.

#### 4. Πίνακας tblSubjectCategory:

	Column Name	Data Type	Length	Allow Nulls
▶	[Subject Category]	char	100	
?	[Subject Code]	char	10	

Σχήμα 3.1.4 Πίνακας tblSubjectCategory

Στο παραπάνω πίνακα (Σχήμα 3.1.4) περιλαμβάνονται μόνο δύο πεδία. κατηγορία μαθήματος και κωδικός μαθήματος αυτός ο πίνακας έχει σαν πρωτεύων κλειδί το κωδικό μαθήματος και χρησιμοποιείτε από το κομμάτι της αυτοματοποιημένης επιλογής μαθημάτων για να διευκρινίζει σε ποια κατηγορία μαθήματος ανήκει δηλαδή εάν είναι υποχρεωτικό, προαιρετικό η επιλογής υποχρεωτικό

#### 5. Πίνακας Prerequisites

	Column Name	Data Type	Length	Allow Nulls
▶	PrerequisiteID	int	4	
	Subject_Code	nvarchar	50	✓
	Subject_Prerequisite	nvarchar	50	✓

Σχήμα 3.1.5 Πίνακας Prerequisites

Ο παραπάνω πίνακας (Σχήμα 3.1.5) περιέχει 3 πεδία τα οποία είναι απαραίτητα, για τον έλεγχο στο θέμα των αλυσίδων πχ ένας μαθητής δεν έχει το δικαίωμα να δηλώσει το μάθημα Μαθηματικά II εάν δεν έχει περασμένο το μάθημα του Μαθηματικά I.

## 6. Πίνακας ΕΥ

	Column Name	Data Type	Length	Allow Nulls
▶	Semester	int	4	
	N_EY	int	4	

Σχήμα 3.1.6 Πίνακας ΕΥ

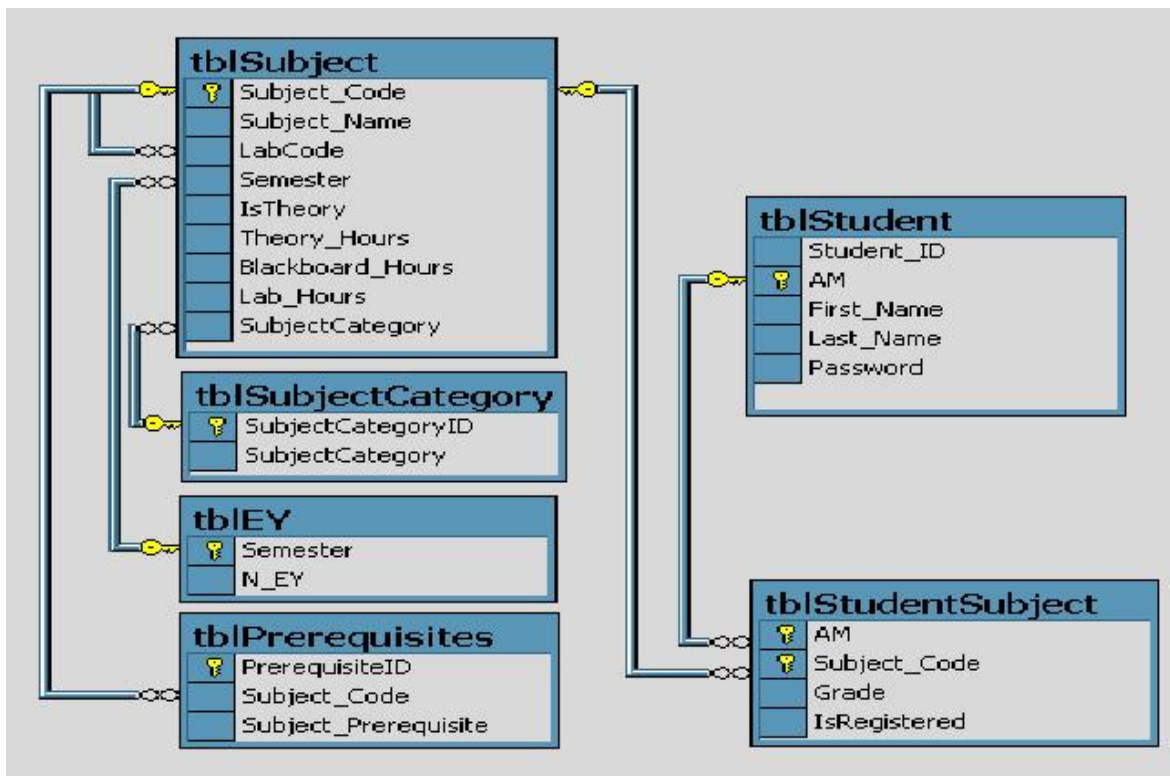
Ο παραπάνω πίνακας (Σχήμα 3.1.6) περιέχει δύο πεδία, το Εξάμηνο και ο αριθμός επιλογών. Αυτός ο πίνακας μας δείχνει πόσα μαθήματα επιλογής είναι δυνατόν να επιλεγούν κάθε εξάμηνο. Πχ στο 7<sup>ο</sup> εξάμηνο επιλέγονται μόνο 2 μαθήματα επιλογής.

### 3.2 Διάγραμμα βάσης δεδομένων – Συσχετίσεις πινάκων

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

Στο Σχήμα 3.2.1 φαίνεται το διάγραμμα της βάσης δεδομένων. Μπορούμε να παρατηρήσουμε τις σχέσεις μεταξύ των πινάκων.

1. Ο πίνακας tblSubject συνδέεται με τον πίνακα tblPrerequisites απο το πεδίο Subject\_Code με σχέση «ένα προς πολλά» δηλαδή το ένα μάθημα μπορεί να έχει πολλά προαπαιτούμενα μαθήματα.
2. Ο πίνακας tblEY συνδέεται με τον πίνακα tblSubject στο πεδίο Semester με σχέση «ένα προς πολλά» δηλαδή πολλα μαθήματα είναι του ίδιου εξαμήνου.
3. Ο πίνακας tblSubject συνδέεται με τον εαυτό του απο το πεδίο Subject\_Code προς το πεδίο LabCode με σχέση «ένα προς ένα» δηλαδή ένα μάθημα μπορεί να έχει ένα κωδικό για το ανάλογο του εργαστήριο.
4. Ο πίνακας tblSubjectCategory συνδέεται με τον πίνακα tblSubject απο το πεδίο SubjectCategoryID προς το πεδίο SubjectCategory με σχέση «ένα προς ένα» δηλαδή κάθε μάθημα είναι μίας συγκεκριμένης κατηγορίας.
5. Ο πίνακας tblSubject συνδέεται με τον πίνακα tblStudentSubject στο πεδίο Subject\_Code με σχέση «ένα προς πολλά» δηλαδή ένα μάθημα μπορεί να έχει περαστή απο πολλούς μαθητές.
6. Καί ο πίνακας tblStudent συνδέεται με τον πίνακα tblStudentSubject στο πεδίο AM με σχέση «ένα προς πολλά» δηλαδή ένας μαθητής μπορεί να δηλώση πολλά μαθήματα.



Σχήμα 3.2.1 Διάγραμμα βάσης δεδομένων

### 3.3 Αποθηκευμένες Διαδικασίες - Stored Procedures

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

Ενδεικτικά μερικά από τα stored procedure που χρησιμοποιήθηκαν είναι:

```
CREATE PROCEDURE spFillSubjects AS
(select Subject_Code, Subject_Name, Semester, SubjectCategory
from tblSubject
where tblSubject.Subject_Code Not In
(select Subject_Code
From tblStudentSubject
where Grade >= 5 and AM=830 and tblSubject.Subject_Code not in
(select distinct tblPrerequisites.Subject_Code
from tblPrerequisites
where tblPrerequisites.Subject_Prerequisite Not In
(select tblStudentSubject.Subject_Code
from tblStudentSubject
where tblStudentSubject.Grade >= 5 and AM=830))))
ORDER BY Subject_Code
GO
```

```
CREATE PROCEDURE dbo.spRegistration_ins
(
    @AM nvarchar(200),
    @SubjectCode nvarchar(50)
)
AS
SET NOCOUNT OFF;
INSERT INTO tblStudentSubject(AM, Subject_Code, Grade, IsRegistered)
VALUES (@AM, @SubjectCode, Null, 1);
GO
```

Τα stored procedure προσφέρουν :

- ο ΕΥΚΟΛΙΑ : Όταν ένα stored procedure τροποποιηθεί στον SQL, αν το όνομά του και οι μεταβλητές, που χρειάζεται, παραμείνουν ίδια δεν είναι αναγκαίο να επέμβει κάποιος στον κώδικα ASP.NET.

- ΤΑΧΥΤΗΤΑ ΣΤΗΝ ΕΦΑΡΜΟΓΗ : Τα stored procedure εκτελούνται γρήγορα διότι δεν χρειάζεται να περάσουν οι παράμετροι στις υπορουτίνες κάθε φορά που καλούνται. Αυτό γίνεται μόνο την πρώτη φορά που καλούνται τα stored procedure.
- ΚΑΛΥΤΕΡΗ ΧΡΗΣΗ ΤΩΝ ΠΟΡΩΝ : Τα stored procedure εκτελούνται στον SQL, δεν επιβαρύνουν το σύστημα του χρήστη και εκμεταλλεύονται καλύτερα το δίκτυο, αφού ο πελάτης δεν στέλνει μέσω δικτύου μεγάλα ερωτήματα, αλλά μόνο το όνομα του stored procedure με τις απαραίτητες παραμέτρους. Ο server λαμβάνει το μήνυμα του πελάτη, εκτελεί το stored procedure και το αποτέλεσμα αποστέλλεται στον πελάτη. Τέλος τα stored procedure εκμεταλλεύονται την ταχύτητα του SQL server και γενικά το υλικό (Hardware) του.
- ΑΞΙΟΠΙΣΤΙΑ - ΣΤΑΘΕΡΟΤΗΤΑ - ΑΝΕΞΑΡΤΗΣΙΑ : Όλοι οι χρήστες της Βάσης Δεδομένων μπορούν να χρησιμοποιούν τα ίδια stored procedure χωρίς να παίρνουν μεγάλο ρίσκο λάθους και είναι ευκολότερη η διόρθωση των διεργασιών αυτών.



## 4. Κεφάλαιο – Ανάλυση Διαδικασιών σε ASP.NET

### 4.1 Επικύρωση για τις Web φόρμες

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

Το ASP.NET έχει έξι είδη στοιχείων ελέγχου επικύρωσης: το RequiredFieldValidator, RegularExpressionValidator, CompareValidator, RangeValidator, CustomValidator και το ValidationSummary. Όλα αυτά τα στοιχεία ελέγχου έχουν ένα κοινό σύνολο από ιδιότητες και μεθόδους, οι περισσότερες από τις οποίες, κληρονομούνται από την κλάση BaseValidator και από την κλάση WebControl που βρίσκονται στο χώρο ονομάτων System.UI.WebControls και την κλάση Control που μπορεί να βρεθεί στο χώρο ονομάτων System.Web.UI. Μια

εξάιρεση σε αυτόν τον κανόνα είναι η ιδιότητα Text, η οποία κληρονομείται από την κλάση Label.

#### 4.1.1 RequiredFieldValidator

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

Η σύνταξη για αυτό το στοιχείο ελέγχου είναι η παρακάτω:

```
<asp:requiredfieldvalidator id="reqCardName"
    runat="server" CssClass="validator"
    ErrorMessage="*You must enter Name"
    ControlToValidate="txtCardName"
    EnableClientScript="False">
</asp:requiredfieldvalidator>
```

Αυτό το παράδειγμα δημιουργεί ένα στοιχείο ελέγχου RequiredFieldValidator με όνομα reqCardName που επικυρώνει το στοιχείο ελέγχου txtCardName. Αν δεν υπάρχει τιμή στο στοιχείο ελέγχου θα εμφανιστεί το μήνυμα σφάλματος «\*You must enter Name». Παρατηρούμε ότι λείπει η ιδιότητα InitialValue που από default έχει τιμή το κενό. Επίσης υπάρχει η ιδιότητα EnableClientScript="False" η οποία ορίζει ότι δεν είναι ενεργοποιημένη η επικύρωση από την πλευρά του πελάτη.

#### 4.1.2 RegularExpressionValidator

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

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

```
<asp:regularexpressionvalidator id="regMail" runat="server" CssClass="validator"
    EnableClientScript="False"
    ValidationExpression="\w+([+.]w+)*@w+([-.]w+)*\w+([-.]w+)*"
    ControlToValidate="txtMail"
    ErrorMessage="*Invalid E-Mail Address">
</asp:regularexpressionvalidator>
```

Αυτό το παράδειγμα δημιουργεί ένα στοιχείο ελέγχου RegularExpressionValidator με όνομα regMail που επικυρώνει το στοιχείο ελέγχου txtMail. Όταν εισάγεται μια τιμή συγκρίνεται με την ιδιότητα ValidationExpression που στην συγκεκριμένη περίπτωση επιβεβαιώνει την σωστή εισαγωγή ενός mail. Επίσης υπάρχει η ιδιότητα EnableClientScript="False" η οποία ορίζει ότι δεν είναι ενεργοποιημένη η επικύρωση από την πλευρά του πελάτη.

## 4.2 Ασφάλεια

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

### 4.2.1 Ασφάλεια του IIS (Internet Information Server)

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

### 4.2.2 Το αρχείο Web.config

Το web.config που είναι αποθηκευμένο σε ένα xml αρχείο στο κατάλογο ρίζα περιγράφει διάφορες ρυθμίσεις διαμόρφωσης. Υπάρχει ένας κόμβος σε αυτό το αρχείο που ονομάζεται <security> και περιέχει πληροφορίες διαμόρφωσης ασφαλείας για την ASP.NET εφαρμογή μας. Ακόμα σε αυτό το αρχείο υπάρχουν και διάφορες άλλες ιδιότητες που ανάλογα το πως θα ορισθούν επιτρέπουν ή απαγορεύουν σε διάφορους χρήστες την πρόσβαση στον πόρο που ζητούν.

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

```

<appSettings>

<add key="CONNECTION_STRING" value="workstation id=localhost;packet
size=4096;
    _integrated security=SSPI;data source=localhost;persist
security info=False;
    _initial catalog=Thesis" />

<add key="PAGE_EXPIRED_STRING" value="Page has expired. Please log in
again" />

</appSettings>

```

### 4.3 Cookies

Ένας δημοφιλής τρόπος προσαρμογής των αλληλεπιδράσεων με ιστοσελίδες είναι μέσω των cookies. Ένα cookie είναι ένα αρχείο κειμένου, αποθηκευμένο από ένα δικτυακό τόπο στον υπολογιστή κάποιου, που επιτρέπει στον τόπο να παρακολουθεί τις ενέργειες του επισκέπτη. Την πρώτη φορά που ο χρήστης επισκέπτεται τον δικτυακό τόπο, ο υπολογιστής του χρήστη μπορεί να δεχθεί ένα cookie. Αυτό το cookie ενεργοποιείται στην συνέχεια κάθε φορά που ο χρήστης επισκέπτεται ξανά αυτόν τον τόπο. Οι συλλεγόμενες πληροφορίες προορίζονται να είναι μια ανώνυμη εγγραφή που περιέχει δεδομένα τα οποία χρησιμοποιούνται για την προσωποποίηση των μελλοντικών επισκέψεων του χρήστη στην τοποθεσία. Για παράδειγμα, τα cookies σε μια εφαρμογή αγορών μπορεί να αποθηκεύουν μοναδικά προσδιοριστικά για τους χρήστες. Όταν ένας χρήστης προσθέτει τεμάχια σε μια online αγορά ή εκτελεί μια άλλη εργασία που προκύπτει από μια αίτηση στον Web server, ο server λαμβάνει ένα cookie που περιέχει τα μοναδικά προσδιοριστικά του χρήστη. Ο server χρησιμοποιεί κατόπιν τα μοναδικά προσδιοριστικά για να εντοπίσει το «καλάθι» αγορών και εκτελεί οποιαδήποτε αναγκαία επεξεργασία.

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

Κάθε αλληλεπίδραση βασισμένη στο HTTP ανάμεσα σε έναν client και έναν server, περιλαμβάνει μια επικεφαλίδα που περιέχει πληροφορίες είτε για την αίτηση (όταν η επικοινωνία ξεκινά από τον client προς τον server), είτε για την απόκριση (όταν η επικοινωνία ξεκινά από τον server προς τον client.) Όταν μια Web Form λαμβάνει μια

αίτηση, η επικεφαλίδα περιλαμβάνει πληροφορίες όπως ο τύπος αίτησης (π.χ Get) και οποιαδήποτε cookies που έχουν προηγουμένως αποσταλεί από τον server για να αποθηκευθούν στο σύστημα του client. Όταν ο server διατυπώσει την απάντηση του, οι πληροφορίες επικεφαλίδας περιλαμβάνουν οποιαδήποτε cookies θέλει ο server να αποθηκεύσει στον υπολογιστή του client και άλλες πληροφορίες, όπως ο τύπος MIME της απάντησης.

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

#### 4.4 Σύνδεση με Βάση Δεδομένων

Όπως προαναφέραμε απαραίτητη είναι η δημιουργία ενός connection string για την σύνδεση με τη βάση. Παρακάτω παραθέτουμε τις εντολές που χρησιμοποιήσαμε για την σύνδεση με την βάση μας.

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

    Dim con As New SqlConnection
    Dim cmd As New SqlCommand
    Dim dr As SqlDataReader

    If Not Page.IsValid Then Exit Sub

    Try
        con.ConnectionString =
            System.Configuration.ConfigurationSettings.AppSettings("CONNE
            CTION_STRING")
        con.Open()
        cmd.CommandType = CommandType.Text

        'Database connection & query for password validation

        cmd.CommandText = "SELECT tblStudent.AM, tblStudent.Password
        FROM tblStudent WHERE " & _
        "AM = '" & tbUsername.Text & "' AND " & "Password = '" &
        tbPassword.Text & "'"
        cmd.Connection = con
        dr = cmd.ExecuteReader(CommandBehavior.SequentialAccess)

        If dr.Read Then
            Session("Student_ID") = dr.GetValue(0).ToString.Trim
            Response.Redirect("Main.aspx", False)
        Else
            Session("PAGE") = "logon.aspx"
            lblMessage.Text = "<font color=red>Sorry, " & _
            "invalid username or password!</font><p>"
        End If

    Catch ex As Exception

        Session("PAGE") = "logon.aspx"

        lblMessage.Text = "<font color=red>Sorry, " & _
            "invalid username or password!</font><p>"

    Finally
        If Not dr Is Nothing Then dr.Close()
        cmd.Dispose()
        con.Close()

    End Try

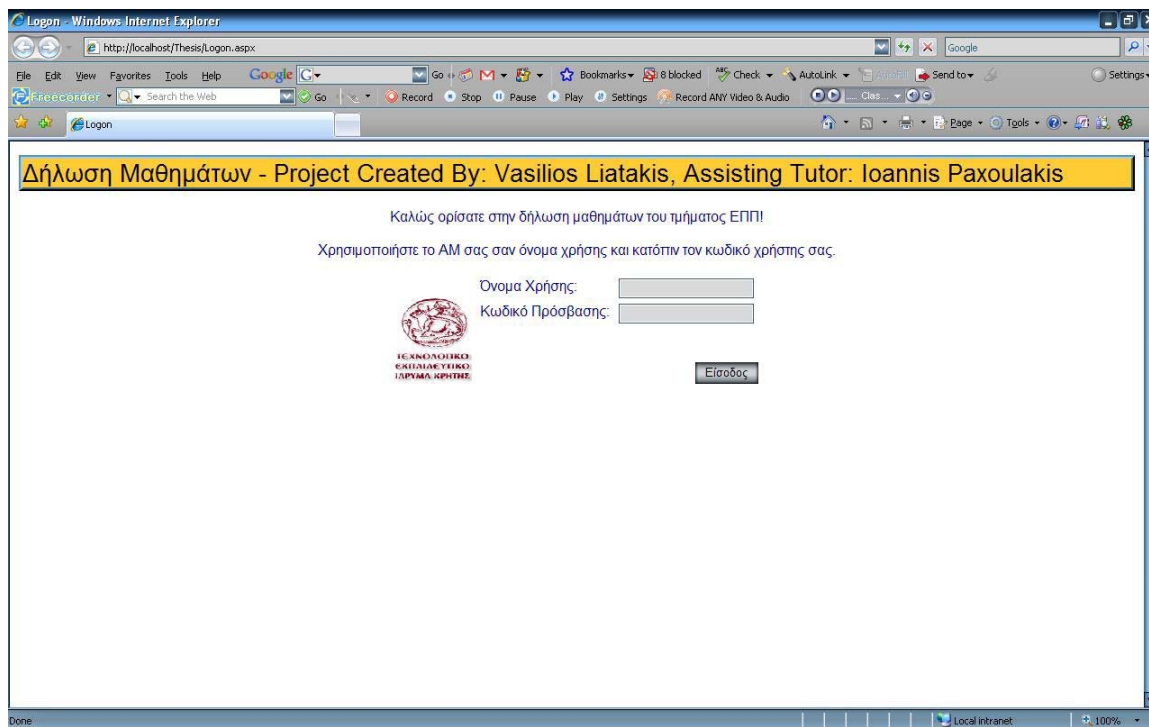
End Sub
```

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



## 5. Κεφάλαιο – Ανάλυση της εφαρμογής μας

### 5.1 Η φόρμα Logon.aspx



Σχήμα 5.1 Logon.aspx

Η παραπάνω φόρμα (Σχήμα 5.1) είναι η πρώτη φόρμα που εμφανίζεται όταν ξεκινάμε την web εφαρμογής μας, στην σελίδα αυτή επιτρέπετε η είσοδος μόνο από χρήστες στους οποίους τους έχει δοθεί κωδικό πρόσβασης. Το Όνομα Χρήστης είναι σε μορφή ΑΜ πχ 830.

Για της ανάγκες της εφαρμογής μας έχει δοθεί πρόσβαση μόνο σε 3<sup>ης</sup> χρήστες, ο καθένας από τους οποίους βρίσκεται σε διαφορετικό στάδιο των σπουδών του:

1. **Username:** 830    **Password:** 26111977

Ο χρήστης αυτός, είναι τελειόφοιτος, χρωστάει μόνο 2 θεωρίες, την πτυχιακή και την πρακτική του εξάσκηση.

2. **Username:** 1227 **Password:** 021951

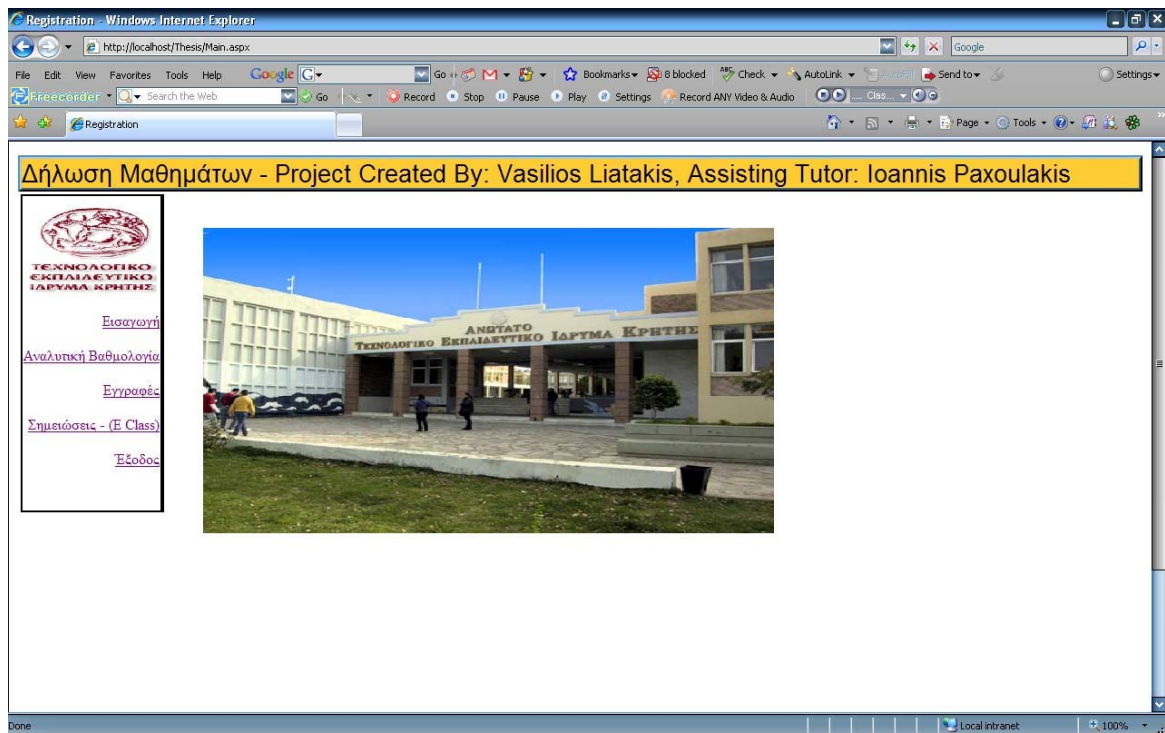
Ο χρήσης αυτός, είναι μαθητής 6<sup>ου</sup> εξαμήνου. Μαζί με αυτά χρωστάει τα ακόλουθα μαθήματα μαζί και με της ανάλογες αλυσίδες τους

1. Μαθηματικά Ι
2. Δίκτυα Δεδομένων Θεωρία

3. **Username:** 2350 **Password:** 021951

Ο χρήστης αυτός είναι νεόφυτος, και μόλις εντάχθηκε στην σχολή δηλαδή μαθητής 1<sup>ου</sup> εξαμήνου, και τώρα για πρώτη φορά θα γραφτή σε κάποια μαθήματα μέσω της σελίδας μας.

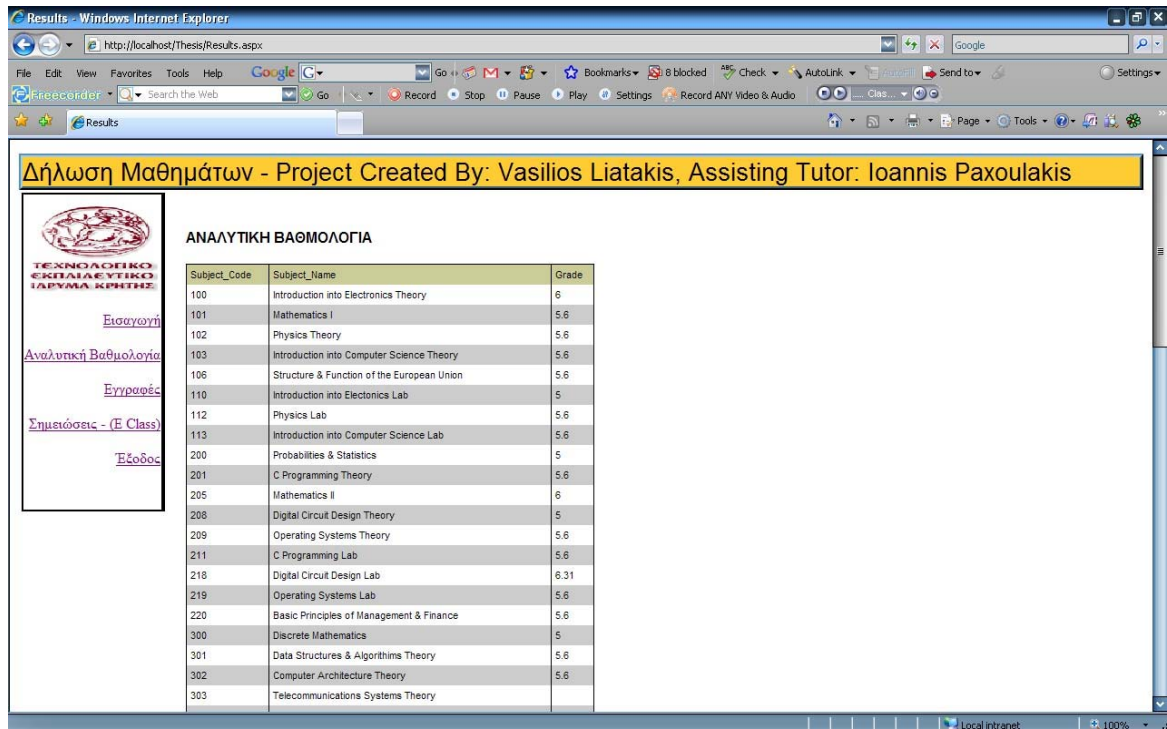
## 5.2 Η φόρμα Main.aspx



Σχήμα 5.2 Main.aspx

Η παραπάνω φόρμα (Σχήμα 5.2) είναι η πρώτη φόρμα που μας εμφανίζετε μόλις καταφέρει ο χρήστης μας να εισέρθη στο σύστημα μας με επιτυχία.

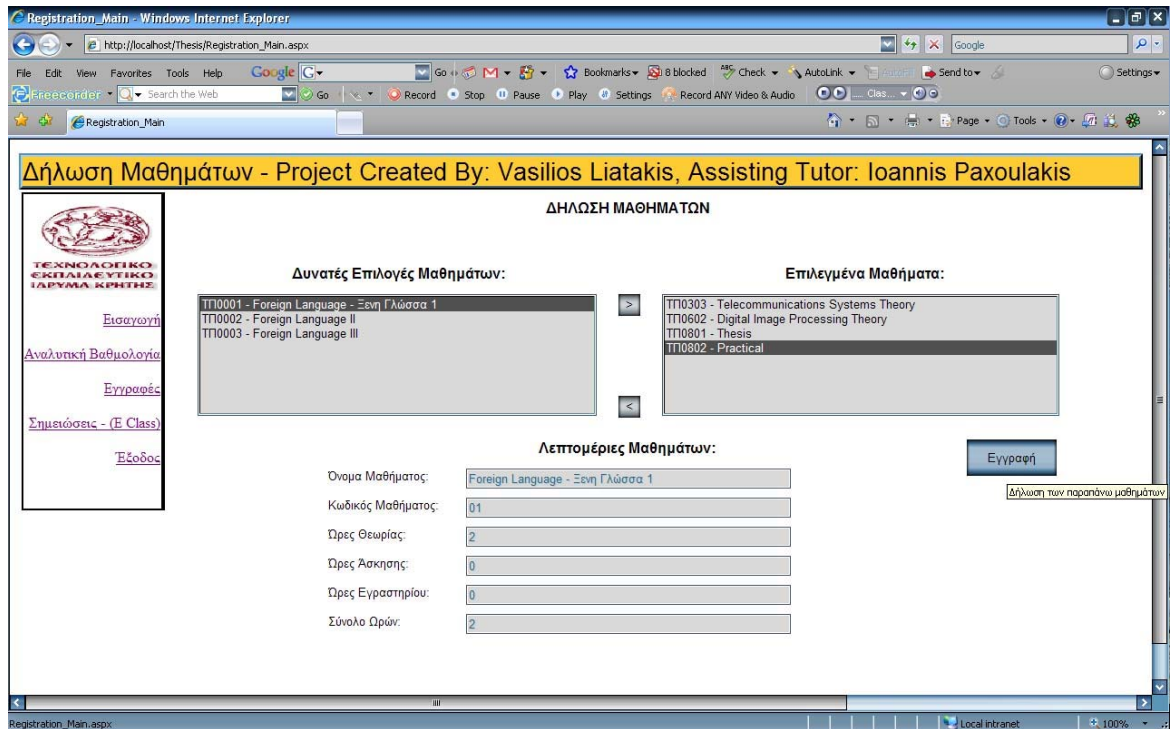
### 5.3 Η φόρμα Results.aspx



Σχήμα 5.3 Results.aspx

Η παραπάνω φόρμα (Σχήμα 5.3) εμφανίζεται όταν πατηθεί το hyperlink Αναλυτική Βαθμολογία. Αυτή η σελίδα είναι διαφορετική για των κάθε μαθητή και μας δείχνει ποια μαθήματα έχουνε περαστεί από των συγκεκριμένο μαθητή μαζί και ποια μαθήματα έχουνε δηλωθεί, και ας μην είναι περασμένα.

## 5.4 Η φόρμα Registration\_Main.aspx



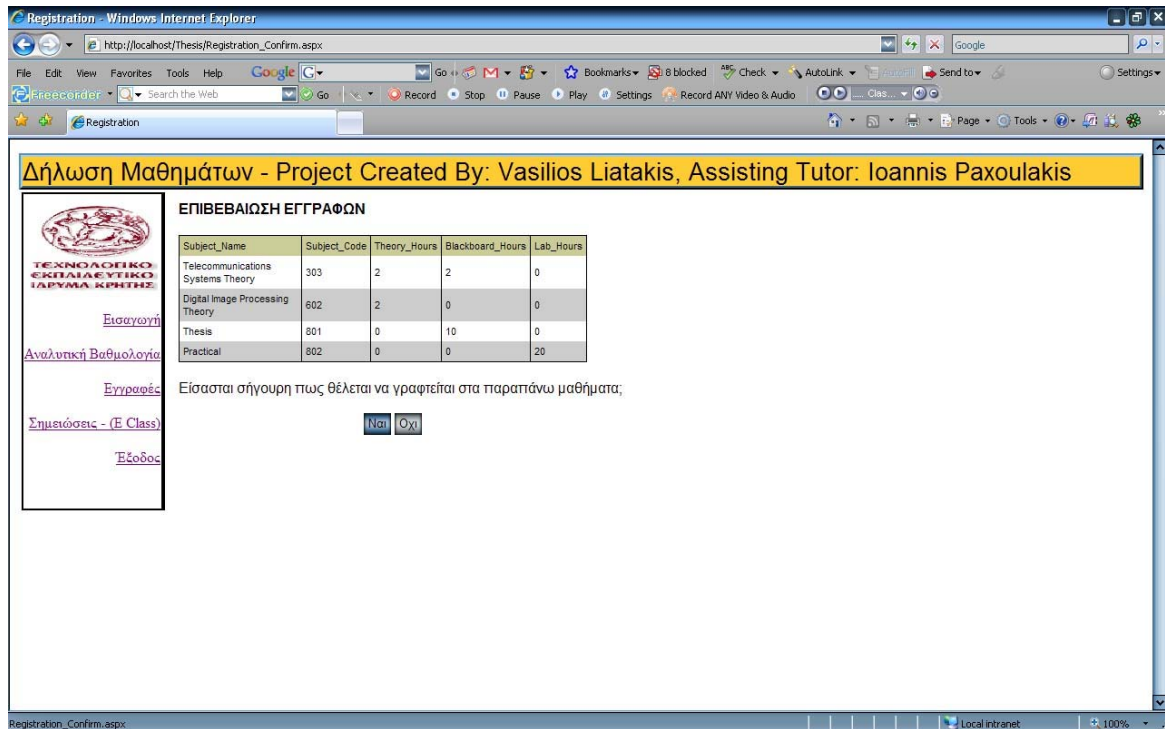
Σχήμα 5.4 Registration\_Main.aspx

Η παραπάνω φόρμα (Σχήμα 5.4) εμφανίζεται όταν επιλέγουμε το Hyperlink Εγγραφές. Εδώ μας εμφανίζονται στο αριστερό πλαίσιο τα μαθήματα που έχουμε δικαίωμα να επιλέξουμε, εάν επιλέξουμε κάποιο από αυτά και πατήσουμε το βελάκι δεξιά γίνεται επιλεγόμενο μάθημα και φέυγη σαν δυνατή επιλογή, και γίνεται πλέον επιλεγόμενο μάθημα.

Επιλέγοντας κάποιο μάθημα «όταν τα μαθήματα είναι δυνατές επιλογές», στο κάτω πλαίσιο «Λεπτομέρειες Μαθημάτων» μας εμφανίζονται και οι σχετικές λεπτομέρειες του μαθήματος, πχ Κωδικός Μαθήματος, ώρες θεωρίας, ώρες εργαστηρίου κτλ

Όταν έχουμε αποφασίσει και επιλέξει όσα μαθήματα έχουμε σκοπό να δηλώσουμε, πατάμε το κουμπί Εγγραφή όπου μας εμφανίζεται οι επόμενη σελίδα σαν φάση επιβεβαίωσης.

## 5.5 Η φόρμα Registration\_Confirm.aspx



Σχήμα 5.5 Registration\_Confirm.aspx

Η παραπάνω φόρμα (Σχήμα 5.5) μας εμφανίζεται σαν φάση επιβεβαίωσης των μαθημάτων που έχουμε επιλέξει να γραφτούμε. Πατώντας το κουμπί ναι γραφόμαστε στα παραπάνω μαθήματα – όταν γίνει αυτό μας εμφανίζεται στην δεξιά πλευρά της οθόνης ένα μήνυμα «Η εγγραφή ολοκληρώθηκε». Πατώντας το όχι επιστρέφουμε στην προηγούμενη σελίδα Registration\_Main στην οποία δεν έχει επιλεγεί κάποιο μάθημα, δηλαδή βρίσκεται στην αρχική του κατάσταση.

## 5.6 To Hyperlink <http://eclass.epp.teiher.gr/eclass/>



Σχήμα 5.6 <http://eclass.epp.teiher.gr/eclass/>

Το Hyperlink eclass (Σχήμα 5.6) είναι απλά ένα Link που ενσωμάτωσα στην πτυχιακή μου για να δίνει στους φοιτητές που την χρησιμοποιούν την δυνατότητα να έχουνε και πρόσβαση στις online σημειώσεις των μαθημάτων τους.

## 6. Βιβλιογραφία – Πηγές

- Sams Teach Yourself ASP.NET Second Edition in 21 Days, Chris Payne
- Sams Teach Yourself MySQL in 10 Minutes, Sams Publishing 2006
- ASP.NET , Ryan, Danny , John Wiley and Sons Ltd 2001
- Programming ASP.NET , Liberty, Jesse, Hurwitz, Dan , O'Reilly 2003
- Asp.Net 2.0 Revealed , Lorenz, Patrick , Springer-Verlag Berlin and Heidelberg GmbH & Co. KG 2003
- Mastering ASP.NET with VB.Net , Jones, A.Russell , John Wiley and Sons Ltd 2002
- Building Secure Microsoft ASP.NET Applications , Microsoft Press , Microsoft Press,U.S. 2003
- Distributed Data Applications with ASP.NET , Homer, Alex, Sussman, Dave , Springer-Verlag Berlin and Heidelberg GmbH & Co. KG 2004
- ASP.NET Data Web Controls , Mitchell, Scott , Macmillan Computer Publishing (a Pearson Education company) 2003
- Professional ASP.NET 2.0 , Evjen, Bill, Hanselman, Scott, Muhammad, Farhan , John Wiley and Sons Ltd 2005
- Beginning ASP.Net 2.0 Databases , Kauffman, John, Thangarathinam, Thiru , John Wiley and Sons Ltd , 2005
- Programming Microsoft ASP.NET 2.0 Applications , Esposito, D. , Microsoft Press,U.S. 2006
- Beginning ASP.NET 2.0 , Ullman, Chris, Kauffman, John, Hart, Chris , John Wiley and Sons Ltd , 2005
- Beginning ASP.Net 1.1 Databases , Maharry, Dan, Foggon, Damien , APress,US 2004
- Visual Basic .Net “Εισαγωγή στο .NET και τα Web Services”, δεύτερη έκδοση , Deitel & Nieto , Μ. Γκιούρδας , 2003
- Πλήρες Εγχειρίδιο της Visual Basic .NET , Πετρούτσος , Μ. Γκιούρδας , 2005
- SQL Server 2000 Web Application Developer's Guide* , Craig Utley, Berkley 2001.
- Rebecca M. Riordan, *Προγραμματισμός στο SQL Server 2000 Βήμα Βήμα* (μτφρ. Τάκης Άλβας), Αθήνα 2001.