

Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Εφαρμοσμένης Πληροφορικής και Πολυμέσων

# Πτυχιακή Εργασία



**Θέμα:** “Ανάπτυξη διαδικτυακής  
εκπαιδευτικής εφαρμογής για την θεωρητική εκπαίδευση των  
υποψήφιων οδηγών αυτοκινήτων.”

Εισηγητές : Δρ. Αθανάσιος Μαλάμος  
Μιχάλης Κλεισαρχάκης  
Σπουδάστρια : Δελητζάκη Μαρία Α.Μ. 1308

Ηράκλειο Δεκέμβριος 2010

## ΠΕΡΙΕΧΟΜΕΝΑ

Πρόλογος

### **1. Κεφάλαιο** – Το λογισμικό πακέτο

1.1 Συνοπτική περιγραφή του συστήματος.....	
1.2 Απαιτήσεις Συστήματος.....	
1.3 Δυναμικές και Στατικές Ιστοσελίδες.....	
1.3.1 Τι είναι ένα στατικό site.....	
1.3.2 Τι είναι ένα δυναμικό site.....	
1.3.3 Η χρήση ενός δυναμικού site μας παρέχει και άλλα πλεονεκτήματα.....	
1.3.4 Επιπλέον χρήσεις ενός δυναμικού site.....	
1.4 Το ASP αναλυτικά.....	
1.5 Αντικείμενα ASP.....	
1.6 Συστατικά ASP.....	
1.7 ASP και Βάσεις Δεδομένων.....	
1.8 Συγγραφή κώδικα με την SQL.....	
1.9 CSS.....	

### **2. Κεφάλαιο** – Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης

2.1 Πλατφόρμα .NET.....	
2.2 Χαρακτηριστικά του .NET Framework.....	

2.3	Πλεονεκτήματα του .NET.....
2.4	Μειονεκτήματα .NET (κυρίως στην ανάπτυξη παιχνιδιών).....
2.5	Αρχιτεκτονική .NET.....
2.6	Visual Basic .NET.....
2.7	ASP.NET C#.....
2.8	Web Forms και ASP .NET.....

### **3. Κεφάλαιο** – Περιγραφή της Βάσης Δεδομένων

3.1	Γενικά για τις βάσεις δεδομένων και ADO.NET.....
3.2	Σχεσιακές λειτουργίες και SQL Queries.....
3.3	Το ΣΔΒΔ Microsoft Access 2007.....
3.4	Γιατί Access 2007.....

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

4.1	Επικύρωση για τις Web φόρμες - ASP.NET Validation Controls.....
4.1.1	RequiredFieldValidator.....
4.1.2	RegularExpressionValidator.....
4.1.3	ASP.NET ΑΡΧΙΤΕΚΤΟΝΙΚΗ.....
4.2	Ασφάλεια.....
4.2.1	Ασφάλεια του IIS (Internet Information Server).....
4.2.2	Το αρχείο Web.config.....
4.3	Cookies.....
4.4	Σύνδεση με Βάση Δεδομένων.....

**5.Κεφάλαιο** - Προγράμματα που χρησιμοποιήθηκαν ( Adobe Photoshop)

Adobe Photoshop

5.1	Δυνατότητες του Photoshop.....
5.2	Η εργαλειοθήκη του Photoshop .....
5.3	Τα εργαλεία επιλογής.....
5.4	Τα Εργαλεία Ζωγραφικής .....
5.5	Τα Εργαλεία Επεξεργασίας .....
5.6	Οι Στρώσεις (Layers) .....
5.7	Γέμισμα περιοχής με άλλο χρώμα (το Εργαλείο του Κουβά).....
5.8	Το Εργαλείο Ντεγκραντέ .....
5.9	Τα Φίλτρα (Filters) του PhotoShop.....

**6. Κεφάλαιο** – Η εκπαιδευτική εφαρμογή

6.1	Τα μέρη της εφαρμογής αναλυτικά.....
6.2	Η εφαρμογή στο ιντερνέτ.....
6.3	Η λειτουργία.....

**7. Κεφάλαιο** – Επίλογος

**8. Κεφάλαιο**– Βιβλιογραφία Πηγές

## 1. Κεφάλαιο – Το λογισμικό πακέτο

### 1.1 Συνοπτική περιγραφή του συστήματος

*Το αντικείμενο αυτής της πτυχιακής εργασίας είναι ο σχεδιασμός, η ανάπτυξη και η υλοποίηση μιας εκπαιδευτικής εφαρμογής η οποία θα βοηθήσει τους υποψήφιους οδηγούς αυτοκινήτων στην θεωρητική εκπαίδευσή τους. Σε αυτή την εφαρμογή μπορεί να μπαίνει οποιοσδήποτε χρήστης και να πραγματοποιεί μια σειρά από τεστ των 30 ερωτήσεων το κάθε ένα και διάρκειας 20 λεπτών της ώρας.*

Κάθε φορά που ο χρήστης επιθυμεί να πραγματοποιήσει ένα τεστ, θα του παρέχεται ένα σετ 30 τυχαίων ερωτήσεων από τις συνολικές θεωρητικές ερωτήσεις του Κ.Ο.Κ. (κώδικα οδικής κυκλοφορίας). Το κάθε τεστ θα διαρκεί 20 λεπτά της ώρας και ακόμη θα δίνεται η δυνατότητα στο χρήστη μέσω της εφαρμογής να διαπιστώσει αν έχει επιτύχει στο τεστ ή αν έχει αποτύχει. *Η παραπάνω λειτουργία της εφαρμογής και η τυχαία επιλογή ερωτήσεων δίνει την δυναμικότητα στην εφαρμογή.*

Η εφαρμογή έχει αναπτυχθεί σε τεχνολογία **ASP.NET**, με το πρόγραμμα **Visual Studio 2008**, με βάση δεδομένων σε **MS Access 2007**. Η βάση δεδομένων περιλαμβάνει όλα τα συστατικά που είναι απαραίτητα για την υλοποίηση της εκπαιδευτικής εφαρμογής. (ερωτήσεις, πολλαπλές επιλογές απαντήσεων και πεδία με τις σωστές απαντήσεις)

**1.2 Απαιτήσεις Συστήματος** Οι απαιτούμενες τεχνικές προδιαγραφές της πλατφόρμας εγκατάστασης είναι οι ακόλουθες:

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

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

➤ Κοινές προδιαγραφές:

- Μνήμη: 128MB τουλάχιστον (προτεινόμενη 512MB)
- Επεξεργαστής: Pentium 133 MHz ή γρηγορότερος
- Web server: Microsoft Internet Information Services (IIS) 5.0+

### 1.3 Δυναμικές και Στατικές Ιστοσελίδες

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

#### 1.3.1 Τι είναι ένα στατικό site

Στατικό site είναι μία συλλογή ιστοσελίδων αμετάβλητων κειμένων, χρησιμοποιώντας οποιοδήποτε πρόγραμμα επεξεργασίας. Οι σελίδες αυτές δύνανται να περιλαμβάνουν εκτενείς πληροφορίες (όπως στοιχεία επικοινωνίας,

τιμές, προϊόντα / υπηρεσίες, ακόμα και γραφικά), αλλά δεν μπορούν να τροποποιούνται εύκολα, γρήγορα και με χαμηλό κόστος.

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

### 1.3.2 Τι είναι ένα δυναμικό site

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

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

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

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

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

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

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

Αποθήκευση πληροφοριών σχετικά με προτιμήσεις των πελατών και προσωποποίηση σελίδων σύμφωνα με τις ανάγκες κάθε πελάτη.

Δυνατότητα δημιουργίας στατιστικών αναφορών για λήψη αποφάσεων.

Παράδειγμα 1:

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

Παράδειγμα 2:

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

Παράδειγμα 3:

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

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

### **1.3.4 Επιπλέον χρήσεις ενός δυναμικού site**

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

Παράδειγμα:



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

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

Αποθήκευση πληροφοριών στη βάση δεδομένων γενικού περιεχομένου

Παράδειγμα:

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

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

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

### 1.4 Το ASP αναλυτικά

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

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

Δεν υπάρχει πραγματικό όριο στο ποσό της επεξεργασίας που μπορεί να εκτελέσει ένα site ανταποκρινόμενο στην είσοδο που λαμβάνει από τον χρήστη, όταν χρησιμοποιείται το ASP και άλλες συμβατές με αυτό τεχνολογίες. Τα καλύτερα

σημερινά Web Sites αναγνωρίζουν τους επισκέπτες τους και τους χαιρετούν με το όνομα τους όταν αυτοί επιστρέφουν στο ίδιο site. Τα καλύτερα εμπορικά Web Sites εκτελούν συνήθως "εξατομικευμένο" marketing, βασιζόμενα σε προφίλ που διατηρούν για τους χρήστες και στις ήδη καταγεγραμμένες αγοραστικές τους συνήθειες.

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

### 1.5 Αντικείμενα ASP

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

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

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

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

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

Αναλυτικά:

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

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

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

Επίσης :

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

Αναλυτικά:

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

**iv)** Όμοια με το αντικείμενο Application, το αντικείμενο Session διαθέτει αρκετά συμβάντα τα οποία μας επιτρέπουν να εκκινήσουμε ενέργειες κατά την

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

### 1.6 Συστατικά

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

### 1.7 ASP και Βάσεις Δεδομένων

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

Τέτοιες βάσεις είναι:

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

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

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

### 1.8 Αντικείμενα ADO

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

Αναλυτικά:

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

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

Ένα από τα βασικότερα αντικείμενα του ADO είναι το **Recordset (Σύνολο Εγγραφών)**, το οποίο μας παρέχει έλεγχο πάνω σε ένα σύνολο εγγραφών

και πεδίων προερχόμενων από μια πηγή δεδομένων, έτσι ώστε να μπορέσουμε να εκτελέσουμε όλες τις κοινές λειτουργίες διαχείρισης που έχουμε στην διάθεσή μας όταν δουλεύουμε με τους πίνακες μιας βάσης δεδομένων(π.χ. εύρεση, επεξεργασία, προσθήκη και διαγραφή εγγραφών, κ.λ.π.).

Αναλυτικά:

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

Variable = Recordset.Fields ("όνομα\_πεδίου")

ή

Recordset("όνομα\_πεδίου")

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

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

Ενδεικτικά αναφέρουμε:

### 1.8.1 Μέθοδοι αντικειμένου Recordset

Open (ανοίγει ένα αντικείμενο Recordset και δέχεται προαιρετικά ορίσματα)

Close (Κλείνει ένα αντικείμενο Recordset)

MoveFirst (Μετακινεί τον δείκτη στην αρχική εγγραφή)

MoveNext (Μετακινεί τον δείκτη στην επόμενη εγγραφή)

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

### 1.8.2 Ιδιότητες αντικειμένου Recordset

BOF (ο δείκτης βρίσκεται στην αρχή του αντικειμένου)

EOF (ο δείκτης βρίσκεται στο τέλος του αντικειμένου)

RecordCount (μας επιστρέφει το πλήθος των εγγραφών)

### 1.9 Συγγραφή κώδικα με την SQL

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

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

Η SQL περιλαμβάνει επίσης μια ποικιλία εντολών οι οποίες επηρεάζουν την δομή της βάσης δεδομένων ή των πινάκων της. Σε αυτές περιλαμβάνονται οι CREATE TABLE, ALTER TABLE, και DROP TABLE, καθώς και παρόμοιες εντολές οι οποίες επηρεάζουν τις προβολές, τα ευρήματα, τα σχήματα και τα domain. Υπάρχουν επίσης ορισμένες εντολές για τον έλεγχο των δικαιωμάτων πρόσβασης και των συναλλαγών καθώς και για τον ορισμό άλλων λειτουργιών μέσω του προγραμματισμού όπως π.χ. η χρήση των cursor και η διεξαγωγή δυναμικών λειτουργιών.

### 1.10 CSS

#### a) Τι Είναι τα CSS

Ο όρος CSS σημαίνει *Cascading Style Sheets*, δηλ. *Διαδοχικά Φύλλα Στυλ*.

Τα στυλ ορίζουν τον τρόπο εμφάνισης των στοιχείων της HTML.  
Τα Στυλ (Styles) αποθηκεύονται σε Φύλλα Στυλ (Style Sheets).

Τα Στυλ προστέθηκαν στην HTML 4.0 για να λύσουν κάποια προβλήματα.

Τα Εξωτερικά Φύλλα Στυλ (External Style Sheets) μπορούν να μας γλυτώσουν από πολλή δουλειά.

Πολλαπλοί ορισμοί στυλ μπορούν να συμπεριληφθούν σ' έναν.

### **β ) Τα Στυλ Λύνουν ένα Κοινό Πρόβλημα**

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

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

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

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

### **γ ) Τα Φύλλα Στυλ μάς Γλυτώνουν από Πολλή Δουλειά**

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

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

Τα CSS αποτελούν μια μεγάλη επιτυχία στον σχεδιασμό του Web (Web design) επειδή δίνουν τη δυνατότητα στους developers να ελέγξουν το στυλ και τη διάταξη πολλών ιστοσελίδων μονομιάς. Σαν Web developer μπορούμε να ορίσουμε ένα στυλ για κάθε HTML στοιχείο και να το εφαρμόσουμε σ' όσες ιστοσελίδες



θέλουμε. Για να κάνουμε μια καθολική (global) αλλαγή, απλά αλλάζουμε το στυλ μία φορά και όλα τα στοιχεία του Web ενημερώνονται αυτόματα.

### δ ) Πολλαπλά Στυλ Καταλήγουν σ' ένα

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

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

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

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

### ε ) Η Σύνταξη των CSS

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

*επιλογέας {ιδιότητα: τιμή}*

*selector {property: value}*

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

*body {color: black}*

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

*p {font-family: "sans serif"}*

Αν θέλουμε να ορίσουμε περισσότερες από μία ιδιότητες, πρέπει να ξεχωρίσουμε την κάθε ιδιότητα με τον χαρακτήρα ;. Το παρακάτω παράδειγμα

δείχνει πώς μπορούμε να ορίσουμε μια κεντραρισμένη παράγραφο με χρώμα κειμένου κόκκινο :

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

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

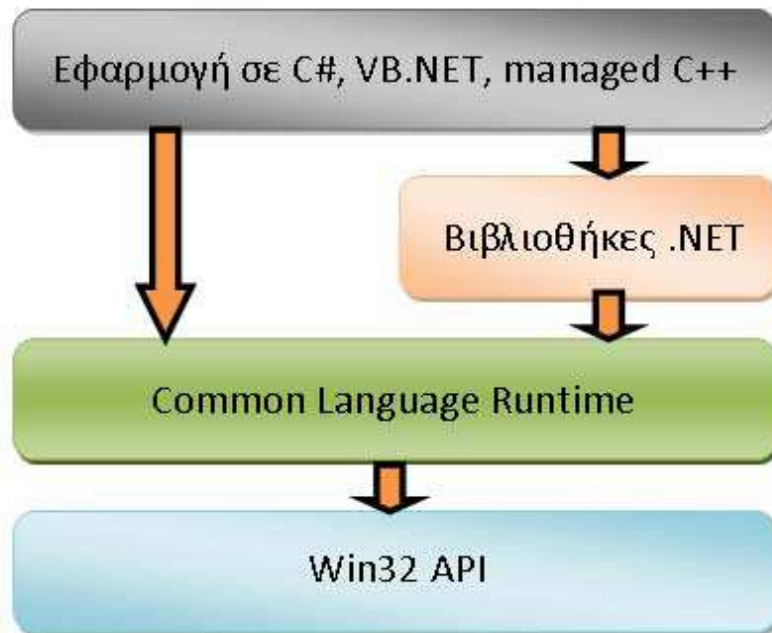
```
p  
{  
  
text-align: center;  
  
color: black;  
font-family: arial  
}
```

## **2. Κεφάλαιο – Εισαγωγή στις πλατφόρμες και τεχνικές ανάπτυξης**

### **2.1 Πλατφόρμα .NET**

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

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



## 2.2 Χαρακτηριστικά του .NET Framework

### 1. Ανεξαρτησία από μία συγκεκριμένη γλώσσα

Είναι δυνατή η δημιουργία εφαρμογών σε οποιαδήποτε γλώσσα  
Κυριότερες γλώσσες είναι η Visual Basic.NET, C#, J#, Visual C++.NET  
συμβατή με .NET  
Υποστηρίζονται ακόμα 20 γλώσσες

### 2. Ανεξαρτησία από μία συγκεκριμένη πλατφόρμα

Η μεταγλώττιση των προγραμμάτων γίνεται σε δύο βήματα. Πρώτα γίνεται το MSIL (Microsoft Intermediate Language) και έπειτα σε κώδικα μηχανής για μία συγκεκριμένη πλατφόρμα.  
Εάν το .NET Framework υπάρχει και είναι εγκατεστημένο για κάποιο λειτουργικό σύστημα τότε αυτό μπορεί να εκτελέσει οποιοδήποτε πρόγραμμα .NET.

### 3. Framework Class Library (FCL)

- Είναι η βασική βιβλιοθήκη κλάσεων που χρησιμοποιούν τα προγράμματα .NET

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

#### 4. Άλλα σημαντικά χαρακτηριστικά

- ASP.NET
- Δημιουργία εφαρμογών για PDA, Mobiles
- Web Services
- Διαλειτουργικότητα γλωσσών
- CLR (Common Language Runtime)

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

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

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

**2.3 Πλεονεκτήματα .NET** Το .NET έχει πολλά πλεονεκτήματα για την ανάπτυξη εφαρμογών:

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

### 2.4 Μειονεκτήματα .NET (κυρίως στην ανάπτυξη παιχνιδιών)

- Το .NET έχει 2 μειονεκτήματα που αφορούν ειδικά την ανάπτυξη βιντεοπαιχνιδιών και όχι την γενική ανάπτυξη εφαρμογών:
- Αυτοματοποιημένη διαχείριση μνήμης, ο χρήστης δεν χρειάζεται να ασχοληθεί με αποδέσμευση μνήμης.
- Το CLR εισάγει μια (μικρή ίσως) καθυστέρηση στην εκτέλεση της εφαρμογής.
- Το πρώτο, αν και είναι πολύ χρήσιμο χαρακτηριστικό για γενικό προγραμματισμό, σε ένα βιντεοπαιχνίδι μπορεί να δημιουργήσει πρόβλημα. Η απελευθέρωση μνήμης που δεν χρησιμοποιείται πλέον είναι μια ακριβή εργασία (από πλευράς χρόνου), η οποία επιπλέον είναι μη-ντετερμινιστική, μπορεί να συμβεί δηλαδή οποτεδήποτε. Αυτό μπορεί να έχει σαν αποτέλεσμα ένα παιχνίδι που τρέχει σε ένα σταθερό ρυθμό 60 καρέ το δευτερόλεπτο, να δει μια δραματική πτώση στο ρυθμό ανανέωσης για 1 δευτερόλεπτο, πράγμα ανεπίτρεπτο στην ανάπτυξη βιντεοπαιχνιδιών.
- Το δεύτερο αφορά τη Just in Time μεταγλώττιση που υποστηρίζει το CLR. Από τη μία εισάγει μια καθυστέρηση στην εκκίνηση του παιχνιδιού μιας και πρέπει να μεταγλωττιστεί ο κώδικας από την άλλη ο μεταγλωττιστής ο ίδιος δεν είναι βέλτιστος με την έννοια ότι δεν παράγει το καλύτερο δυνατό native κώδικα για Windows.
- Και τα δυο προβλήματα αυτά μπορούν να αντιμετωπιστούν αν τα λάβουμε υπόψη κατά την ανάπτυξη της εφαρμογής

### 2.5 Αρχιτεκτονική .NET

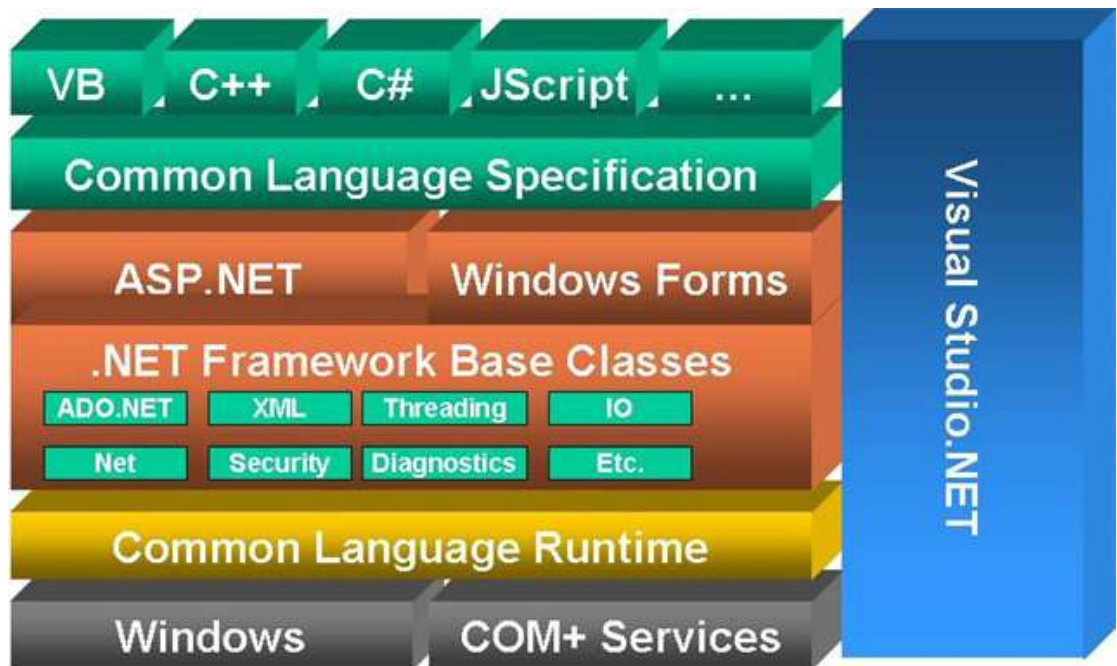
Μέχρι σήμερα, η μεταγλώττιση μιας εφαρμογής από τον αντίστοιχο compiler δημιουργούσε εκτελέσιμα αρχεία (EXE ή DLL). Αυτά περιελάμβαναν απευθείας κώδικα μηχανής για την πλατφόρμα x86 και φυσικά μπορούσαν να τρέξουν μόνο σε αυτού του είδους τα συστήματα.

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

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

.NET μπορεί να εκτελείται σε λειτουργικά συστήματα διαφορετικά από τα Windows, αρκεί να υπάρχει ένα CLR για κάθε διαφορετική συσκευή, λύνοντας έτσι τα χέρια του προγραμματιστή.

Σχήμα :Αρχιτεκτονική .NET



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

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

### 2.6 Visual Basic .NET

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

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

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

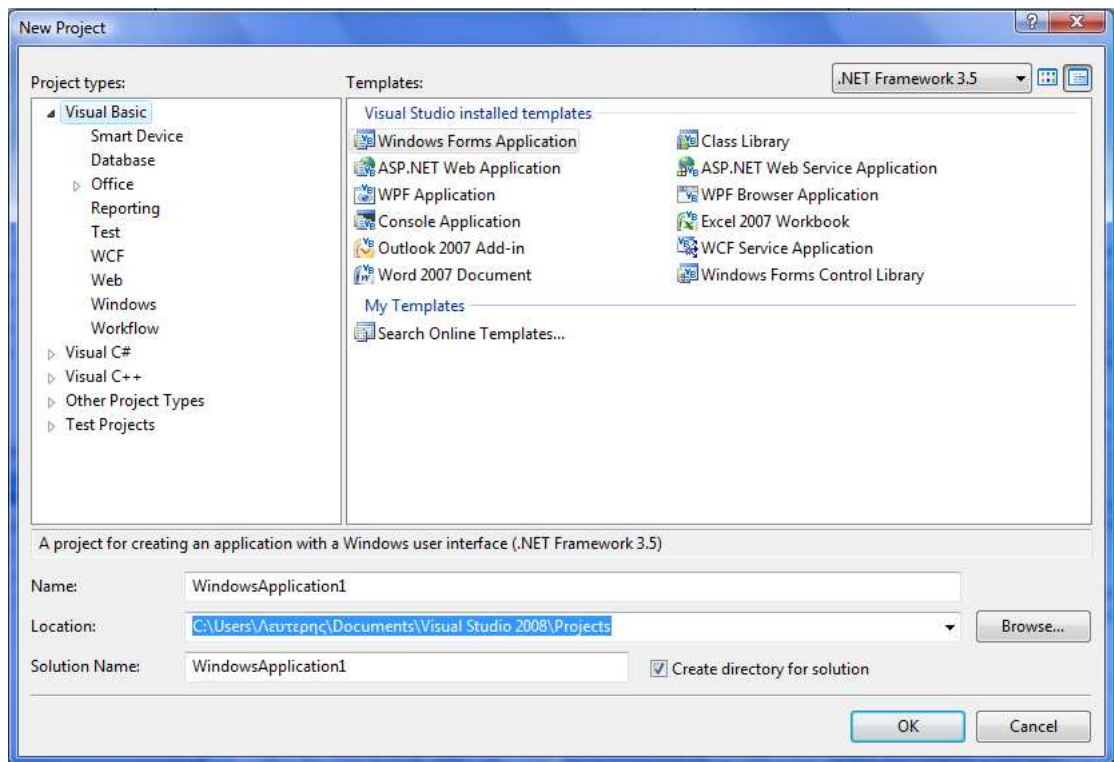
Η Visual Basic ακολούθησε εντυπωσιακή εξέλιξη την τελευταία δεκαετία, από την έκδοση 1.0 στην σημερινή έκδοση .NET, ή 7.0, όπως είναι η εσωτερική αρίθμηση του προϊόντος. Οι νέες εντολές, οι βελτιώσεις στον τρόπο ανίχνευσης των λαθών αλλά και όσα εργαλεία προστέθηκαν μέχρι τη σημερινή έκδοση, δεν άλλαξαν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

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

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



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



Οι εντολές On Error και Resume Next έχουν αντικατασταθεί με πιο σταθερές και συνεπείς λύσεις, όπως οι εντολές Try, Catch και Final, επιτρέποντας τη συγγραφή πιο καθαρού κώδικα. Υπάρχει επίσης η δυνατότητα της αρχικοποίησης μιας μεταβλητής κατά τη δήλωσή της (Dim Y As Integer = 1). Θα βρείτε αλλαγές και σε πολλά ακόμα σημεία που αφορούν σε μεμονωμένα στοιχεία ελέγχου, όπως το μήνυμα MsgBox. Επιπλέον, μια αλλαγή που πιθανόν θα δημιουργήσει προβλήματα με τον κώδικα της προηγούμενης έκδοσης, είναι ο τρόπος αντιμετώπισης των παραμέτρων. Στην έκδοση 6 οι παράμετροι σε μια διαδικασία (function DoSomething(X As Integer)) περνούσαν ως 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 στο νέο φoρμά. 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 σχεδιάσει οθόνες που εμφανίζονται σε διαφανή και πολυεπίπεδα παράθυρα, χαρακτηριστικό που επιτρέπουν τα Windows 2000 και τα Windows XP. Άλλο ένα ενδιαφέρον στοιχείο της Visual Basic .NET είναι η πλήρης υποστήριξη του Unicode, ακόμα και στις φόρμες.

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

## 2.7 ASP.NET C#

### 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++, άλλα ένα ενδιάμεσο κώδικα που στοχεύει το .NET.

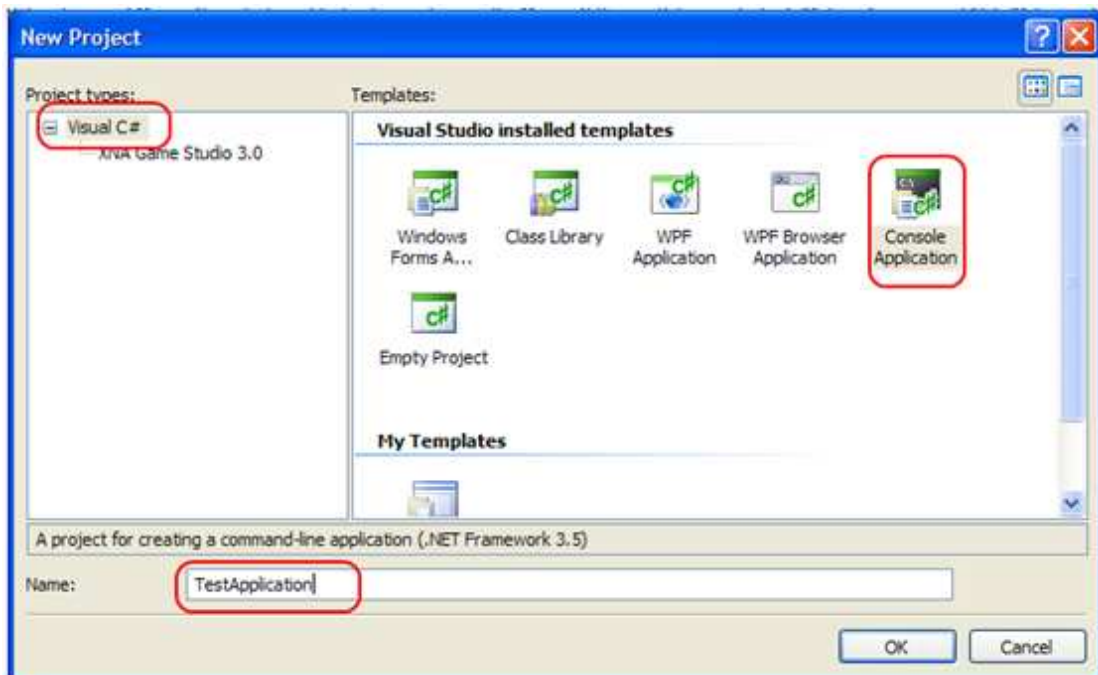
Η C# (προφέρεται σι σαρπ) είναι η γλώσσα επιλογής για ανάπτυξη .NET εφαρμογών. Εν αντιθέσει με την Visual Basic που προσαρμόστηκε για να υποστηρίξει το .NET, η C# έχει σχεδιαστεί εξ αρχής με γνώμονα την πλατφόρμα αυτή και συνεπώς μπορεί να την αξιοποιήσει καλύτερα.

Η C# είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού που δανείζεται στοιχεία από την C++ και την Java. Ο μεταγλωττιστής της γλώσσας παράγει ένα ενδιάμεσο κώδικα, ανάλογα με τον object code της Java, την Microsoft Intermediate Language. Κατά την εκτέλεση της εφαρμογής, το Common Language Runtime, το περιβάλλον δηλαδή μέσα στο οποίο θα τρέξει το πρόγραμμα, διαβάζει τον ενδιάμεσο κώδικα (MSIL) και τον μεταγλωττίζει σε κώδικα Windows.

Για να δούμε πρακτικά πως είναι ο κώδικας μιας C# εφαρμογής τρέχουμε το [Visual C# 2008 Express Edition](#), και δημιουργούμε ένα νέο project.

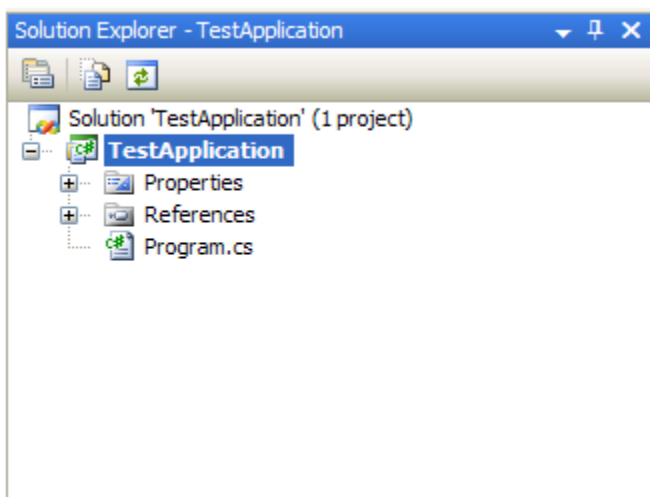


Στην συνέχεια επιλέγουμε σαν Project Type το **Visual C#**, στην συνέχεια ως template το **Console Application** και ονομάζουμε την εφαρμογή μας **TestApplication**, και πατάμε το OK.



Ουσιαστικά με τον τρόπο αυτό καθοδηγήσαμε το Visual Studio να φτιάξει το σκελετό μιας C# εφαρμογής που θα τρέχει σε DOS παράθυρο (console), με απλό κείμενο χωρίς γραφικά.

Το Visual Studio θα δημιουργήσει ένα Solution με το όνομα TestApplication και ένα Project με το ίδιο όνομα.



Κάνοντας διπλό κλικ στο αρχείο Program.c εμφανίζεται ο σκελετός του προγράμματος:

```
01 using System;
```

```
02 using System.Collections.Generic;

03 using System.Linq;

04 using System.Text;

05

06 namespace TestApplication

07 {

08     class Program

09     {

10         static void Main(string[] args)

11         {

12         }

13     }

14 }
```

Πολλά από τα χαρακτηριστικά ενός C# προγράμματος εμφανίζονται ήδη. Οι δηλώσεις **using** στην αρχή του προγράμματος καθορίζουν ποιες βιβλιοθήκες θα χρησιμοποιήσουμε στην εφαρμογή. Το **namespace** ορίζει ένα «χώρο» μέσα στο οποίο θα δημιουργούνται οι μεταβλητές στο πρόγραμμα. Είναι για καθαρά ονοματολογικούς σκοπούς και επιτρέπει το διαχωρισμό 2 μεταβλητών με το ίδιο όνομα που κάθε μια ανήκει σε διαφορετικές βιβλιοθήκες. Μπορεί να το δει κανείς σαν το όνομα ενός καταλόγου (folder) στο δίσκο. Δύο αρχεία με το ίδιο όνομα μπορούν να υπάρχουν σε δυο διαφορετικούς καταλόγους. Για να κάνω αναφορά σε κάποιο από τα δύο χρησιμοποιώ και το όνομα του καταλόγου. Το ίδιο ακριβώς συμβαίνει και με το namespace.

Το βασικό πρόγραμμα είναι μια κλάση (**class**), με το όνομα **Program**. Στην C# όλος ο κώδικας της εφαρμογής πρέπει να είναι μέρος μιας κλάσης, εν αντιθέσει με την C++ που επιτρέπει δημιουργία κώδικα και εκτός κλάσης.

Τέλος η συνάρτηση-μέλος της κλάσης Program, η **Main**, είναι το σημείο εισόδου της εφαρμογής, ανάλογα με την main() της C++. Τα ονόματα της κλάσης του

προγράμματος (Program) και της συνάρτησης εισόδου (Main) είναι προκαθορισμένα και δεν μπορούν να αλλάξουν.

Πατώντας το πλήκτρο F5, η εφαρμογή μεταγλωττίζεται και τρέχει. Πολύ γρήγορα θα δούμε ένα παράθυρο κονσόλας να ανοίγει και να κλείνει. Αυτό γίνεται γιατί ουσιαστικά το πρόγραμμα δεν εκτελεί καμία ενέργεια (η Main είναι άδεια).

Αλλάζουμε το κώδικα της Main λίγο ώστε να βγάλει κάποια έξοδο στο παράθυρο της κονσόλας, το παραδοσιακό "Hello World!":

Εδώ κάνουμε χρήση μιας βιβλιοθήκης του .NET, της **System**, χρησιμοποιώντας το αντικείμενο της **Console**. Το αντικείμενο αυτό διαχειρίζεται είσοδο-έξοδο στο παράθυρο της κονσόλας. Υποστηρίζονται πολλά τέτοια αντικείμενα στο .NET, που υλοποιούν ένα πλήθος λειτουργιών. Χρησιμοποιούμε τη συνάρτηση **WriteLine()** για να απεικονίσουμε μια γραμμή κειμένου και την **ReadLine()** για να διαβάσουμε μια γραμμή κειμένου – στην περίπτωση μας το πλήκτρο ENTER. Στο τέλος της συνάρτησης, η εντολή **return** σηματοδοτεί την λήξη της. Στην συγκεκριμένη περίπτωση είναι περιττή η χρήση της μιας και η Main δεν επιστρέφει τίποτα.

Παρατηρούμε επίσης ότι κάθε εντολή στην C#, όπως και στην C++ πρέπει να τερματίζεται με το ελληνικό ερωτηματικό.

Πατώντας πάλι το F5 ανοίγει το παράθυρο, και απεικονίζεται η γραμμή κειμένου «Hello World!». Για να κλείσει το παράθυρο πρέπει να πατήσουμε Enter.

Ενημερωτικά, ο Intermediate Language κώδικας που παράγεται για την συνάρτηση Main είναι ο εξής:

Είναι σχετικά εύκολο να καταλάβει κανείς την λειτουργία του, συγκρίνοντας τον με τον αρχικό κώδικα της συνάρτησης Main (C#). Είναι για παράδειγμα εμφανές ότι φορτώνει το κείμενο «Hello World!» σε μια προσωρινή μεταβλητή (εντολή IL\_001) και στην συνέχεια καλεί την συνάρτηση WriteLine για να την τυπώσει την οθόνη. Επίσης γίνεται φανερό ότι η βιβλιοθήκη System με το αντικείμενο της Console ανήκουν στο .NET και δεν αποτελούν μέρος της C#.

## 2.8 Web Forms και ASP .NET

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

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

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

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

### Σχήμα : Ο σχεδιασμός ενός website σε asp.net



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

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

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

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

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



### 3. Κεφάλαιο – Περιγραφή της Βάσης Δεδομένων

#### Βάσεις Δεδομένων και ADO.NET



#### 3.1) Γενικά

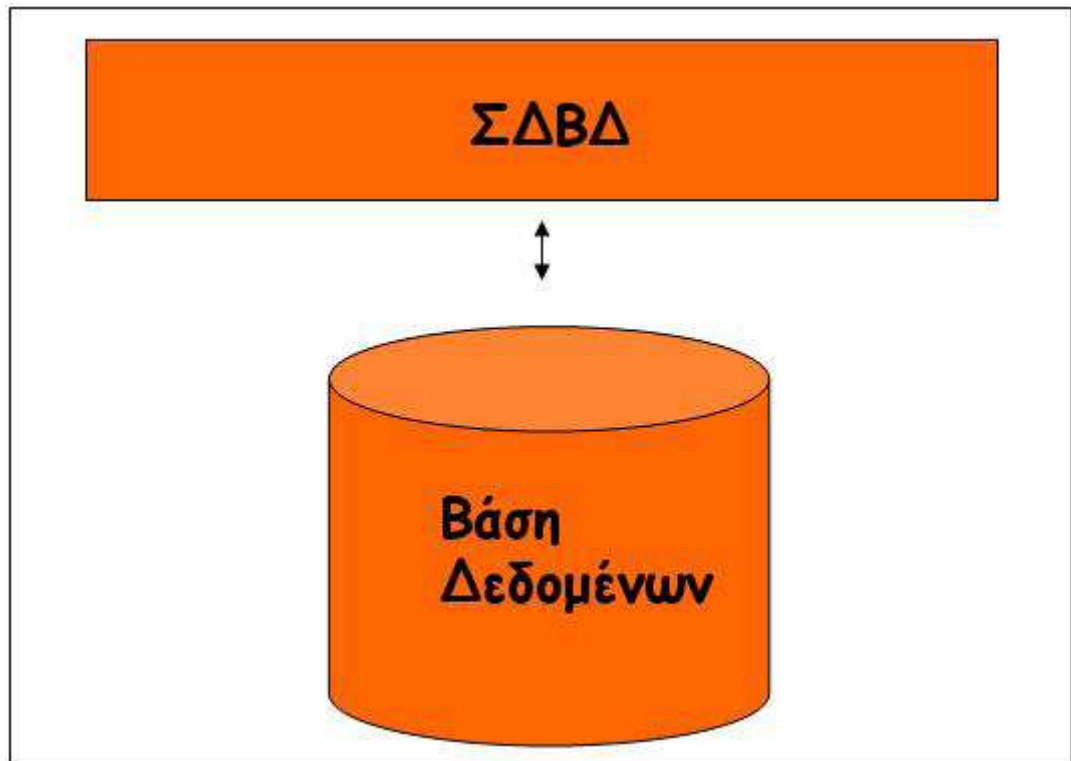
Ο έλεγχος και η διαχείριση της πληροφορίας που σχετίζεται με οποιοδήποτε εφαρμογή είναι από τις σημαντικότερες δυνατότητες που παρέχει ο υπολογιστής στον άνθρωπο, κυρίως χάρη στις βάσεις δεδομένων (Β.Δ.). Μια βάση δεδομένων είναι ουσιαστικά μια δομή οργανωμένης συλλογής δεδομένων. Ένα πιθανό ορισμό θα αποτελούσε : «Μια βάση δεδομένων είναι η συλλογή εγγραφών, αποθηκευμένων σε έναν υπολογιστή με συστηματικό τρόπο, ώστε ένα πρόγραμμα να μπορεί να ανατρέχει σε αυτήν με σκοπό να απαντήσει σε ερωτήσεις». Για καλύτερη ανάκληση (retrieval) και ταξινόμηση των δεδομένων, κάθε εγγραφή οργανώνεται ως ένα σύνολο από στοιχεία-πεδία. Τα επιλεγμένα ανακληθέντα στοιχεία, που αποτελούν την απάντηση στις ερωτήσεις που τίθενται στη βάση (queries), γίνονται η πληροφορία που θα χρησιμοποιηθεί για τη λήψη αποφάσεων ή για την απόκτηση απαραίτητης γνώσης για κάποια εργασία. Το πρόγραμμα που χρησιμοποιείται για να διαχειρίζεται και να θέτει ερωτήματα στη βάση δεδομένων ονομάζεται Σύστημα Διαχείρισης Βάσεων Δεδομένων-ΣΔΒΔ (database management system-DBMS) .

Η κεντρική ιδέα γύρω από μια βάση δεδομένων είναι ότι αποτελεί μια συλλογή εγγραφών ή κομματιών γνώσης. Τυπικά, για κάθε βάση δεδομένων υπάρχει και μια δομική περιγραφή του είδους των στοιχείων που αποθηκεύονται σε αυτήν, κι αυτή η περιγραφή ονομάζεται σχήμα (schema) της Βάσης Δεδομένων. Το σχήμα της βάσης περιγράφει τον τρόπο με τον οποίο απεικονίζονται τα αντικείμενα στη βάση και τις σχέσεις μεταξύ τους. Υπάρχουν διάφοροι τρόποι για τον σχεδιασμό του σχήματος, οι οποίοι ονομάζονται μοντέλα δεδομένων (data models). Το πιο διαδεδομένο σήμερα μοντέλο δεδομένων είναι το σχεσιακό.

Αυστηρά, ο όρος «βάση δεδομένων» αναφέρεται μόνο στην συλλογή των μεταξύ τους σχετιζόμενων εγγραφών, παρόλα αυτά πλέον ο όρος συγχέεται με το ΣΔΒΔ, οπότε πλέον αντιστοιχεί στην ίδια τη βάση και στο σύστημα διαχείρισής της μαζί. Τα ΣΔΒΔ συνήθως κατηγοριοποιούνται ανάλογα με το μοντέλο δεδομένων που υποστηρίζουν: σχεσιακό, αντικειμενοστραφές κλπ. Το μοντέλο δεδομένων είναι αυτό που τείνει να ορίσει ουσιαστικά και την προγραμματιστική γλώσσα, που θα

χρησιμοποιηθεί για την εκτέλεση ερωτημάτων, ώστε αυτή να μπορεί να έχει πρόσβαση στη βάση.

Ένα μεγάλο μέρος της υλοποίησης ενός ΣΔΒΔ αποτελούν και τα στοιχεία που δεν έχουν σχέση με το μοντέλο δεδομένων και αφορά διαχειριστικούς παράγοντες όπως η απόδοση, η συγχρονικότητα (concurrency), η συνέπεια και η ανάκαμψη από αποτυχίες του υλικού (hardware) του υπολογιστή. Στους παράγοντες αυτούς διαφοροποιούνται αρκετά τα ΣΔΒΔ που διατίθενται.



Το μοντέλο δεδομένων σύμφωνα με το οποίο σχεδιάστηκε η δικιά μας βάση δεδομένων για την εφαρμογή της πτυχιακής εργασίας, ακολουθεί το μοντέλο Οντοτήτων-Συσχετίσεων (Entity-Relationship, E-R). Στο μοντέλο αυτό τα δεδομένα περιγράφονται ως οντότητες και συσχετίσεις μεταξύ αυτών. Μια οντότητα μπορεί να είναι ένα αντικείμενο με φυσική ύπαρξη (π.χ. ένας σύνεδρος) ή μια έννοια (π.χ. abstract). Υπάρχουν ισχυροί και μη ισχυροί τύποι οντοτήτων. Οι ισχυρές οντότητες ορίζονται από μόνες τους ενώ οι μη ισχυρές οντότητες ορίζονται μόνο από τις σχέσεις τους με κάποιες ισχυρές, οπότε δεν υπάρχουν χωρίς αυτές. Σε φυσικό επίπεδο οι οντότητες υφίστανται ουσιαστικά ως πίνακες, οπότε η βάση περιέχει ένα σύνολο πινάκων, των οποίων οι σχέσεις δεν ορίζονται ρητά, αλλά μέσω των «κλειδιών» (keys) τους και ξένων κλειδιών (foreign keys) που αναφέρονται σε αυτά. Κάθε εγγραφή μέσα στον πίνακα της οντότητας έχει μια σειρά από πεδία, δημιουργώντας έτσι στήλες από πεδία με την αντίστοιχη τιμή αυτών για κάθε εγγραφή. Το ξένο κλειδί είναι μια στήλη (ή και ο συνδυασμός περισσοτέρων) της

οποίας οι τιμές είναι ίδιες με αυτές στην στήλη κάποιου άλλου πίνακα, οδηγώντας μας έτσι σε εγγραφή άλλης οντότητας. Το πεδίο που αποτελεί το «πρωτεύον κλειδί» (primary key) ενός πίνακα είναι αυτό το οποίο είναι μοναδικό για κάθε εγγραφή και το οποίο χαρακτηρίζει τον κάθε πίνακα μονοσήμαντα.

Από τη χρήση μιας βάσης δεδομένων, προκύπτουν ορισμένα πολύ σημαντικά πλεονεκτήματα :

Δεν υπάρχουν διασκορπισμένα στοιχεία, αντιθέτως τα στοιχεία είναι οργανωμένα έτσι ώστε να μην έχουμε επανάληψη των δεδομένων.

Υπάρχει υποστήριξη πολλαπλών όψεων χρήστη. Έτσι μπορούμε να δώσουμε συγκεκριμένες πληροφορίες σε συγκεκριμένα άτομα ή να αποκρύψουμε πληροφορίες από αυτούς.

Υπάρχει ανεξαρτησία μεταξύ προγραμμάτων και δεδομένων.

Υπάρχει η δυνατότητα προσπέλασης και διαχείρισης της πληροφορίας από απομακρυσμένα μεταξύ τους σημεία .

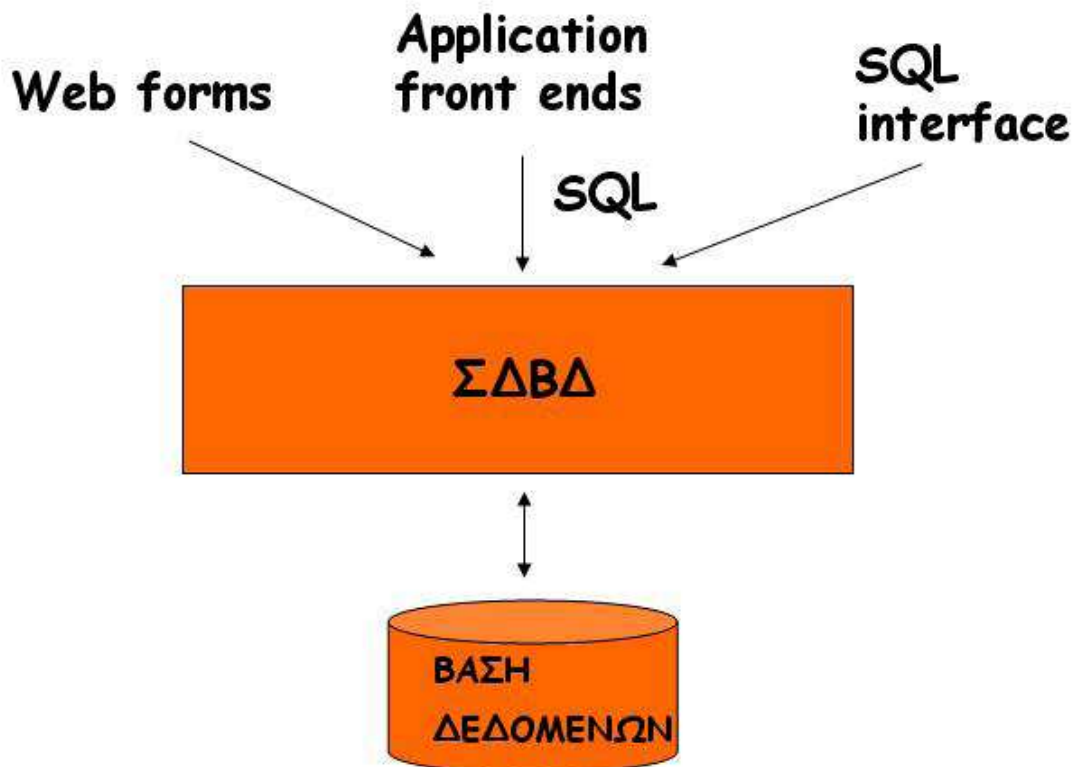
### 3.2) Σχισιακές Λειτουργίες & SQL Queries

Οι χρήστες (ή τα προγράμματα) ζητούν δεδομένα από τις σχισιακές βάσεις δεδομένων στέλνοντας μια ερώτηση, η οποία είναι γραμμένη στην κατάλληλη γλώσσα- συνήθως κάποια διάλεκτο της SQL (Structured Query Language). Παρότι η SQL αρχικά προοριζόταν για τελικούς χρήστες, είναι πλέον πιο συνηθισμένο να είναι ενσωματωμένη σε κάποιο λογισμικό που προσφέρει καλύτερη και ευκολότερη διεπαφή (interface) με τον χρήστη. Στις περισσότερες ιστοσελίδες, που γίνεται αναζήτηση πάνω από κάποια βάση δεδομένων, χρησιμοποιείται SQL για να κάνει τις SQL ερωτήσεις (SQL queries) ώστε να επιστραφούν τα αποτελέσματα. Τα αποτελέσματα κάποιας ερώτησης παρουσιάζονται από την βάση ως μια συλλογή εγγραφών-πλειάδων που περιέχουν τις απαντήσεις. Η πιο απλή ερώτηση είναι η ανάκληση όλων των πλειάδων ενός πίνακα, αλλά τις περισσότερες φορές αυτό που ζητάμε είναι το φιλτράρισμα αυτών βάσει των κριτηρίων που θέτουμε .

Συχνά τα δεδομένα από διαφορετικούς πίνακες συνδυάζονται σε έναν μέσω του τελεστή σύνδεσης (join), το οποίο αποτελεί ουσιαστικά το καρτεσιανό γινόμενο σε συνδυασμό με μια επιπλέον επιλογή (select) στο αποτέλεσμα που προκύπτει από το καρτεσιανό γινόμενο. Η πιο απλή πράξη σύνδεσης είναι η φυσική σύνδεση (natural join), στην οποία επιλέγονται οι εγγραφές που προκύπτουν από το καρτεσιανό γινόμενο, για τις οποίες όλα τα ξένα κλειδιά είναι ίσα με τα αντίστοιχα πρωτεύοντα κλειδιά ενός ή περισσότερων βασικών πινάκων. Η προβολή συγκεκριμένων πεδίων από έναν, περισσότερους πίνακες ή συνδυασμούς πινάκων γίνεται με την δήλωση "SELECT" της γλώσσας SQL (π.χ. "SELECT Name FROM REPRESENTATIVES"). Επίσης, η συνθήκη WHERE της SQL επιτρέπει να γίνεται φιλτράρισμα των επιστρεφόμενων αποτελεσμάτων αλλά λειτουργεί και ως η συνθήκη που θα ορίσει με ποιον τρόπο θα γίνει η σύνδεση μεταξύ 2 πινάκων, δηλαδή ποια κλειδιά θα πρέπει να ταιριαστούν.

Επιπροσθέτως, πρέπει να αναφέρουμε ότι πληροφορία δεν παρουσιάζεται μόνο πρωτογενώς στους πίνακες οντοτήτων αλλά και δευτερογενώς στις «όψεις» (views). Μια όψη μπορεί να οριστεί ως ένα συγκεκριμένο, προκαθορισμένο SELECT ερώτημα στη βάση, το οποίο αναφέρεται σε έναν ή περισσότερους πίνακες της και έχει τη δυνατότητα να συμπεριφερθεί ως ένας εικονικός πίνακας. Πιο απλά, είναι ένα υποσύνολο της βάσης ταξινομημένο και προβαλλόμενο με συγκεκριμένο τρόπο. Για κάθε όψη μπορούμε να ορίσουμε και συγκεκριμένα ποια πεδία επιθυμούμε να βλέπουμε, σε ποια σειρά, πόσο μεγάλη θα είναι η κάθε στήλη, πώς θα αποθηκεύονται τα δεδομένα στην όψη (σε περίπτωση υλοποιημένων όψεων) και ποιος τύπος εγγραφών θα εμφανίζεται. Έτσι, τελικά ο χρήστης βλέπει είτε το υποσύνολο των εγγραφών ή των πεδίων ενός πίνακα, είτε τη σύνδεση 2 ή περισσότερων πινάκων, είτε την ένωση (union) αυτών, είτε μια στατιστική περίληψη ενός πίνακα.

Στο διάγραμμα παρακάτω, βλέπουμε μια εικονική αναπαράσταση αυτών που αναφέρθηκαν παραπάνω, της διαχείρισης δηλαδή μιας βάσης δεδομένων από clients μέσω web browsing.



Στο διάγραμμα παρακάτω, βλέπουμε μια εικονική αναπαράσταση αυτών που αναφέρθηκαν παραπάνω, της διαχείρισης δηλαδή μιας βάσης δεδομένων από clients μέσω web browsing.

Για την διαχείριση των βάσεων δεδομένων, έχουν βγει κατά καιρούς διάφορα συστήματα διαχείρισης βάσεων δεδομένων ή και ανανεωμένες εκδόσεις των είδη υπαρχόντων. Τα πιο γνωστά αλλά και πιο ευρέως διαδεδομένα είναι Microsoft Access, MySQL, Microsoft SQL Server, Microsoft FoxPro και Oracle. Εμείς για τη διαχείριση της βάσης στο παράδειγμα της πτυχιακής, χρησιμοποιήσαμε την Microsoft Access 2007.

### 3.3) Το ΣΔΒΔ της MS Access 2007



Η Microsoft Access, αν και αποτελεί τμήμα του πακέτου λογισμικών Microsoft Office, διαφέρει αρκετά από τα υπόλοιπα προγράμματα (Word, Excel). Εστιάζεται στη δημιουργία και διαχείριση Σχεσιακών Βάσεων Δεδομένων, λειτουργώντας ως πλήρες RDBMS (relational database management system). Το σύστημα Access, από την πρώτη έκδοσή του το 1992 μέχρι σήμερα έχει επιτύχει ένα πολύ μεγάλο αριθμό πωλήσεων σε ολόκληρο το κόσμο και έχει καταστεί πολύ δημοφιλές. Από την πρώτη εμφάνισή του μέχρι και σήμερα, το πρόγραμμα Access έχει πραγματοποιήσει πωλήσεις περισσότερες από 7.000.000 αντιγράφων, σύμφωνα με ανακοινώσεις της εταιρίας Microsoft.

Στις Βάσεις Δεδομένων της Access αποθηκεύονται περιγραφικά δεδομένα, σε διάφορες μορφές. Πέρα από τη δημιουργία πινάκων και τη διατύπωση ερωτημάτων, η Access επιτρέπει την κατασκευή εφαρμογών οι οποίες έχουν πρόσβαση στα δεδομένα. Λειτουργεί και χρησιμοποιείται στο περιβάλλον των windows της ίδιας εταιρίας και χαρακτηρίζεται ως ένα πακέτο διαχείρισης βάσεων δεδομένων, προορισμένο για σχετικά μικρές ή μέσες ανάγκες και απαιτήσεις σε όγκους δεδομένων. Γενικά παρέχει πολλές διευκολύνσεις και είναι αρκετά εύχρηστο. Το πρόγραμμα Access χρησιμοποιείται επίσης με μεγάλη επιτυχία ως εργαλείο **front end**, σε πολλές περιπτώσεις στις οποίες το **back end** της βάσης εξυπηρετείται από άλλο RDBMS με μεγαλύτερες δυνατότητες. Επιτρέπει την

συνεργασία με άλλα προγράμματα διαχείρισης βάσεων δεδομένων, έτσι ώστε να είναι δυνατή η ανταλλαγή δεδομένων ή η μετατροπή και μεταφορά τους από το ένα σύστημα στο άλλο χωρίς μεγάλη δυσκολία. Για παράδειγμα το πρόγραμμα Access επιτρέπει την εργασία σε δεδομένα που προέρχονται από άλλα συστήματα, όπως τα προγράμματα Dbase, Paradox, FoxPro καθώς και άλλα συστήματα DBMS.

Η Access, όπως τα περισσότερα προγράμματα διαχείρισης βάσεων δεδομένων, παρέχει τη δυνατότητα δημιουργίας βάσεων δεδομένων με εύκολο τρόπο, ενώ επιπλέον υποστηρίζει τη διαχείρισή τους. Η διαχείριση υποστηρίζεται είτε μέσω προγραμμάτων εφαρμογών, είτε μέσω ειδικών εργαλείων που παρέχουν τα-DBMS. Τα προγράμματα εφαρμογών αναπτύσσονται για να εξυπηρετούν προκαθορισμένες ανάγκες. Χρησιμοποιούν τα συστήματα DBMS και αυτοματοποιούν τις διαδικασίες, έτσι ώστε οι χρήστες να επιλέγουν τις επιθυμητές διαδικασίες μέσω συγκεκριμένων μενού. Οι χρήστες δεν απαιτείται να γνωρίζουν περί βάσεων δεδομένων, ούτε και περί των συστημάτων διαχείρισής τους. Η ανάπτυξη των προγραμμάτων εφαρμογών πραγματοποιείται με συνδυασμένη χρήση των γλωσσών διαχείρισης (Data Manipulation Language, όπως η SQL) και άλλων ειδικών γλωσσών υψηλού επιπέδου, που υποστηρίζονται από τα συστήματα DBMS και αναφέρονται συνήθως ως γλώσσες 4<sup>ης</sup> Γενιάς (4<sup>th</sup> GL). Αντί των γλωσσών 4<sup>ης</sup> Γενιάς μπορεί να χρησιμοποιούνται και γλώσσες προγραμματισμού, όπως για παράδειγμα οι γλώσσες C, Pascal, Cobol και άλλες.



Θα πρέπει να σημειωθεί ότι η Access παρέχει ένα πολύ εύχρηστο σύστημα ανάπτυξης εφαρμογών κάτω από το λειτουργικό σύστημα των Windows της εταιρίας Microsoft. Οι εφαρμογές αναπτύσσονται πολύ εύκολα και είναι δυνατό ακόμη και στον μη εξειδικευμένο χρήστη να αναπτύξει απλές εφαρμογές, με κάποιες βέβαια προϋποθέσεις. Η γλώσσα προγραμματισμού VBA (Visual Basic for Applications) χρησιμοποιείται εύκολα με την Access και δίνει την δυνατότητα δόμησης απλών εφαρμογών σε σχετικά μικρό χρόνο.

Γενικά όπως όλα σχεδόν τα συστήματα διαχείρισης βάσεων δεδομένων, έτσι και η Access υποστηρίζει με απλό τρόπο, χωρίς την ανάγκη σύνταξης προγράμματος τη δημιουργία:

- Φορμών (Forms)
- Ερωτημάτων (Queries)
- Αναφορών (Reports)

Οι φόρμες εμφανίζονται στις οθόνες των υπολογιστικών συστημάτων, σχεδιάζονται εύκολα και μπορεί να χρησιμοποιούνται από τους τελικούς χρήστες σε πολλές διαφορετικές περιπτώσεις. Συνήθεις φόρμες είναι αυτές που χρησιμοποιούνται κατά την εισαγωγή δεδομένων, καθώς και οι φόρμες ανάκλησης και διαχείρισης των δεδομένων.

Οι αναφορές είναι και αυτές ένα είδος φορμών που συνήθως εκτυπώνονται και περιέχουν πληροφορίες που εξυπηρετούν τις ανάγκες των χρηστών.

Τέλος, τα ερωτήματα σχεδιάζονται και διατυπώνονται έτσι, ώστε να εξυπηρετούν και αυτά τις απαιτήσεις των χρηστών.

Θα πρέπει επίσης να αναφερθεί ότι η εκτέλεση των περισσότερων λειτουργιών της Access δεν παρουσιάζει μεγάλες διαφορές από τα άλλα προγράμματα της Microsoft όπως για παράδειγμα το Word και το Excel και αν ο χρήστης έχει κάποιο βαθμό εξοικείωσης με αυτά, λογικά δεν θα έχει πρόβλημα στη χρήση της. Αν και οι δυνατότητές της δεν πλησιάζουν αυτές των περισσότερο "επαγγελματικών" DBMS, η Access είναι ευρύτατα διαδεδομένη λόγω της φιλικότητάς της. Αποτέλεσμα είναι πολλά GIS πακέτα να περιέχουν ρουτίνες πρόσβασης δεδομένων σε format Access (.mdb).

### 3.4) Γιατί Access 2007



Στις μέρες μας, πολλοί είναι αυτοί που δυσκολεύονται να αποφασίσουν για το ποιο θα ήταν το καλύτερο σύστημα διαχείρισης για τη βάση δεδομένων που θα χρησιμοποιήσουν στην εφαρμογή τους. Όπως αναφέρθηκε προηγουμένως, υπάρχουν διάφορα ΣΔΒΔ, καθ' ένα από τα οποία έχει τα πλεονεκτήματα αλλά και τα μειονεκτήματά του. Είναι λοιπόν φανερό ότι είναι στην κρίση του κάθε προγραμματιστή πιο θα χρησιμοποιήσει και κατά πόσο αυτό θα αντεπεξέλθει στις προσδοκίες του. Στην δικιά μας εφαρμογή, χρησιμοποιήθηκε το ΣΔΒΔ της Access και θεωρήθηκε σωστό, εκτός από αυτά που αναφέρθηκαν παραπάνω, να αναφερθούν

μερικοί από τους λόγους που την προτιμήσαμε και που θα την προτιμούσε κάποιος άλλος προγραμματιστής.

Μερικοί από τους λόγους που την επιλέξαμε είναι οι εξής :

- i. Η Access πιθανόν να αποτελεί ένα γνώριμο περιβάλλον. Κατά πάσα πιθανότητα, στην καθημερινή σας εργασία χρησιμοποιείτε το Word για τη συγγραφή κειμένων και το Excel για τους υπολογισμούς σας. Είστε εξοικειωμένοι με το περιβάλλον τους, το ίδιο και οι υπόλοιποι συνεργάτες της επιχείρησής σας. Είστε ευχαριστημένοι από την ευκολία χρήσης και την αποτελεσματικότητα των εφαρμογών του Microsoft Office. Είναι λογικό λοιπόν, για την ανάπτυξη της εφαρμογής σας να επιλέξετε το εργαλείο που είναι πιο κοντά σ' αυτά που χρησιμοποιείτε καθημερινά, την Microsoft Access.
- ii. Η Access είναι από τα πιο σταθερά συστήματα βάσεων δεδομένων. Η σταθερότητα που παρέχει κατά τη χρήση της σας εξασφαλίζει μία χωρίς προβλήματα λειτουργία της εφαρμογής σας, ακόμα και κάτω από συνθήκες υψηλής πίεσης.
- iii. Η Access είναι αδιαφιλονίκητα η κορυφαία εφαρμογή ανάπτυξης βάσεων δεδομένων για προσωπικούς υπολογιστές. Προέρχεται από τους κατασκευαστές του λειτουργικού συστήματος που χρησιμοποιείτε, την Microsoft. Έτσι, μπορείτε να είστε σίγουροι ότι:
  - η εφαρμογή σας θα εκτελείται, χωρίς προβλήματα, για πολύ καιρό
  - θα ενσωματώνει άμεσα όλες τις νέες τεχνολογίες
  - θα υποστηρίζεται από τους μεγαλύτερους κατασκευαστές software στον κόσμο, τη Microsoft
- iv. Ορθότερη καταχώρηση των δεδομένων σας, εξάλειψη των υπολογιστικών και ελαχιστοποίηση των ανθρώπινων λαθών καθώς και άμεση και ορθή πληροφόρηση ανά πάσα στιγμή που διευκολύνει την λήψη αποφάσεων.
- v. Αποτελεί το ιδανικό εργαλείο για την ανάπτυξη εφαρμογών κατά παραγγελία, εφαρμογών που ανταποκρίνονται πλήρως στις ανάγκες σας και αναπροσαρμόζονται στον τρόπο λειτουργίας της επιχείρησής σας όσο συχνά θέλετε. Η Access σας παρέχει τα εργαλεία για την ταχύτερη ανάπτυξη πλήρως λειτουργικών εφαρμογών.



- vi. Περιέχει τη γλώσσα προγραμματισμού Visual Basic, μία από τις ισχυρότερες γλώσσες προγραμματισμού σε περιβάλλον Windows και σας επιτρέπει να ενσωματώνετε προχωρημένες λύσεις στην εφαρμογή σας. Για ακόμα πιο εξεζητημένες περιπτώσεις, η Access μπορεί να κάνει χρήση ActiveX Controls και βιβλιοθηκών dll, μέσω των οποίων πρακτικά δεν υπάρχει κανένας περιορισμός στις δυνατότητες της.
  - vii. Η Access υποστηρίζει πλήρως το σχεσιακό μοντέλο, επιτρέποντας σας να ορίσετε με λεπτομέρεια τη δομή αποθήκευσης της πληροφορίας σας, τις σχέσεις και τις προϋποθέσεις που αυτή πρέπει να πληρεί.
- Αυτοί είναι μερικοί από τους λόγους για τους οποίους πιστεύουμε ότι η Access είναι το ιδανικό εργαλείο στο οποίο πρέπει να αναπτύξετε την εφαρμογή σας.

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

### 4.1 επικύρωση για τις Web φόρμες - ASP.NET Validation Controls

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

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

Στο ASP.NET, μπορείτε να προσθέσετε στοιχεία ελέγχου επικύρωσης σε όλα σχεδόν τα στοιχεία ελέγχου δεδομένων εισόδου της σελίδας, που είναι είτε HTML στοιχεία ελέγχου διακομιστή, είτε στοιχεία ελέγχου διακομιστή μιας Web φόρμας. Μπορούν να ορισθούν περισσότερα από ένα είδη επικύρωσης σε ένα συγκεκριμένο στοιχείο ελέγχου εισόδου, κάνοντας δυνατή την επικύρωση με διάφορα κριτήρια. Όταν εκτελείται η ASP.NET σελίδα με τα στοιχεία ελέγχου επικύρωσης, οι τιμές των στοιχείων ελέγχου εισόδου που επηρεάζονται από τα στοιχεία ελέγχου επικύρωσης υπόκεινται σε επεξεργασία σε σχέση με την αντίστοιχη λογική που τα διέπει, όπως αναφέρεται στα στοιχεία ελέγχου επικύρωσης. Μετά τους υπολογισμούς, οι ιδιότητες των στοιχείων ελέγχου επικύρωσης παίρνουν την τιμή True ή False, ανάλογα με το αποτέλεσμα. Μόλις γίνει η επεξεργασία όλων των στοιχείων ελέγχου επικύρωσης, η ίδια η σελίδα ορίζει μια ιδιότητα με βάση τις τιμές των ιδιοτήτων επικύρωσης των στοιχείων ελέγχου. Αν 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 `Validation Summary`. Όλα αυτά τα στοιχεία ελέγχου έχουν ένα κοινό σύνολο από ιδιότητες και μεθόδους, οι περισσότερες από τις οποίες, κληρονομούνται από την κλάση `BaseValidator` και από την κλάση `WebControl` που βρίσκονται στο χώρο ονομάτων `System.UI.WebControls` και την κλάση `Control` που μπορεί να βρεθεί στο χώρο ονομάτων `System.Web.UI`. Μια εξαίρεση σε αυτόν τον κανόνα είναι η ιδιότητα `Text`, η οποία κληρονομείται από την κλάση `Label`.

### 4.1.1 RequiredFieldValidator

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

Η σύνταξη για αυτό το στοιχείο ελέγχου είναι η παρακάτω:

```
<asp:RequiredFieldValidator runat="server" Display="dynamic"
ID="valRequireName" SetFocusOnError="true"
```

```
ControlToValidate="txtName" ErrorMessage="Το όνομα σας είναι
απαραίτητο">*</asp:RequiredFieldValidator>
```

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

### 4.1.2 RegularExpressionValidator

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

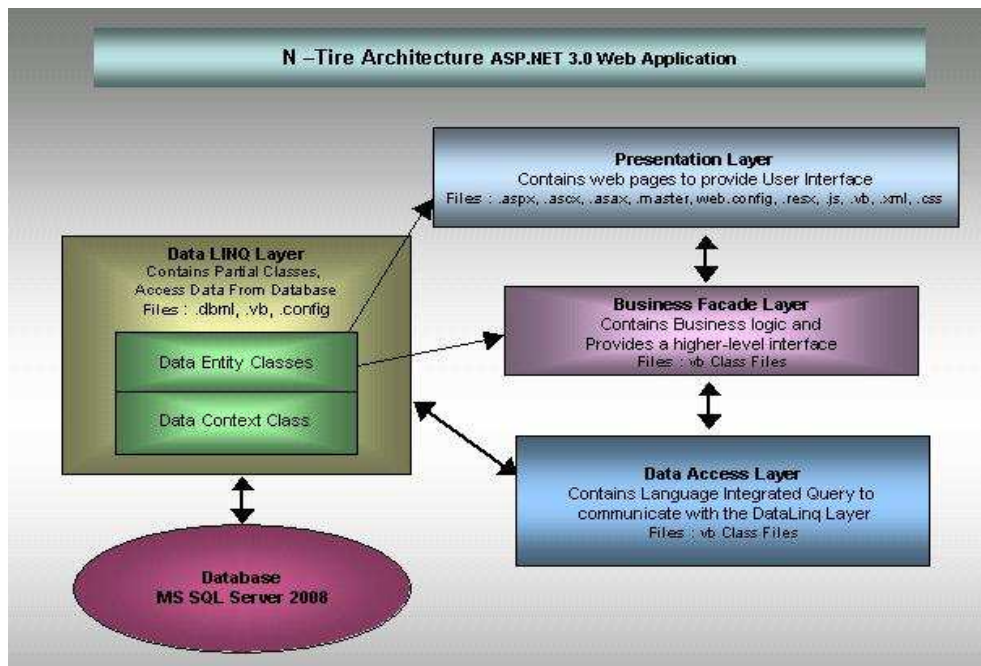
Αυτό το είδος της επικύρωσης σας επιτρέπει να ψάχνετε για προβλεπόμενες ακολουθίες χαρακτήρων, όπως εκείνες που χρησιμοποιούνται για τις διευθύνσεις ηλεκτρονικού ταχυδρομείου, για αριθμούς τηλεφώνου κτλ. Υπάρχει μια μοναδική ιδιότητα που σχετίζεται με τα στοιχεία ελέγχου `RegularExpressionValidator`: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

```
<asp:RegularExpressionValidator runat="server" Display="dynamic"
    ID="valEmailPattern" SetFocusOnError="true"

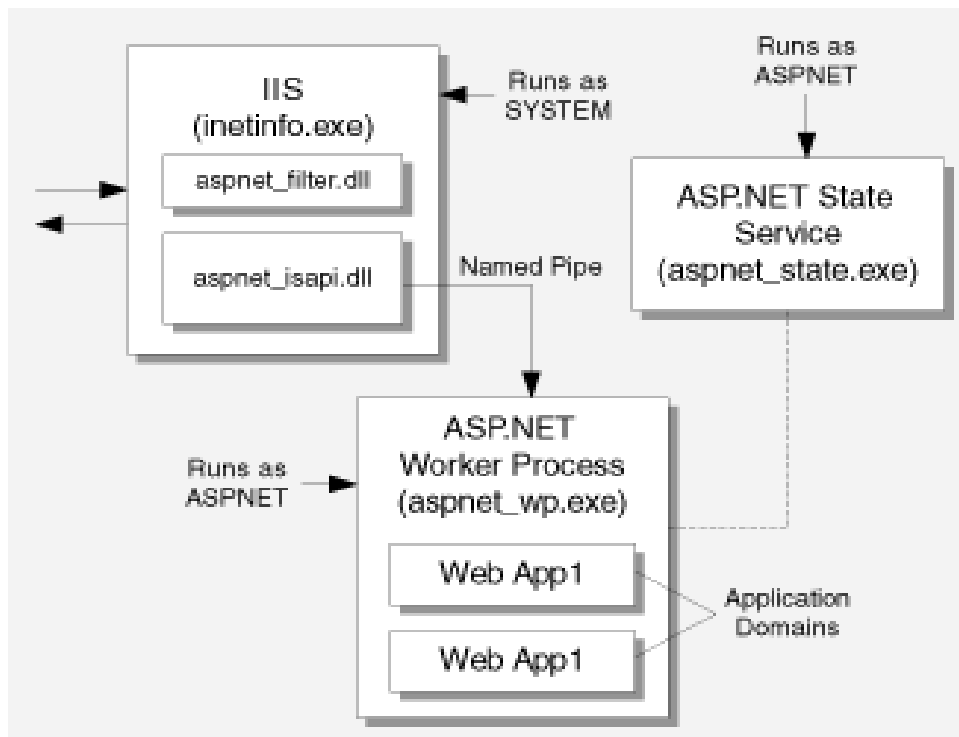
    ControlToValidate="txtEmail" Validation Expression="\w+([-
+.']\w+)*@\w+([-.\w+]*\.\w+([-.\w+])*" ErrorMessage="The e-mail
    address you specified is not well-formed">*</asp:
    RegularExpressionValidator>
```

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

### 4.1.3 ASP.NET APXITEKTONIKH



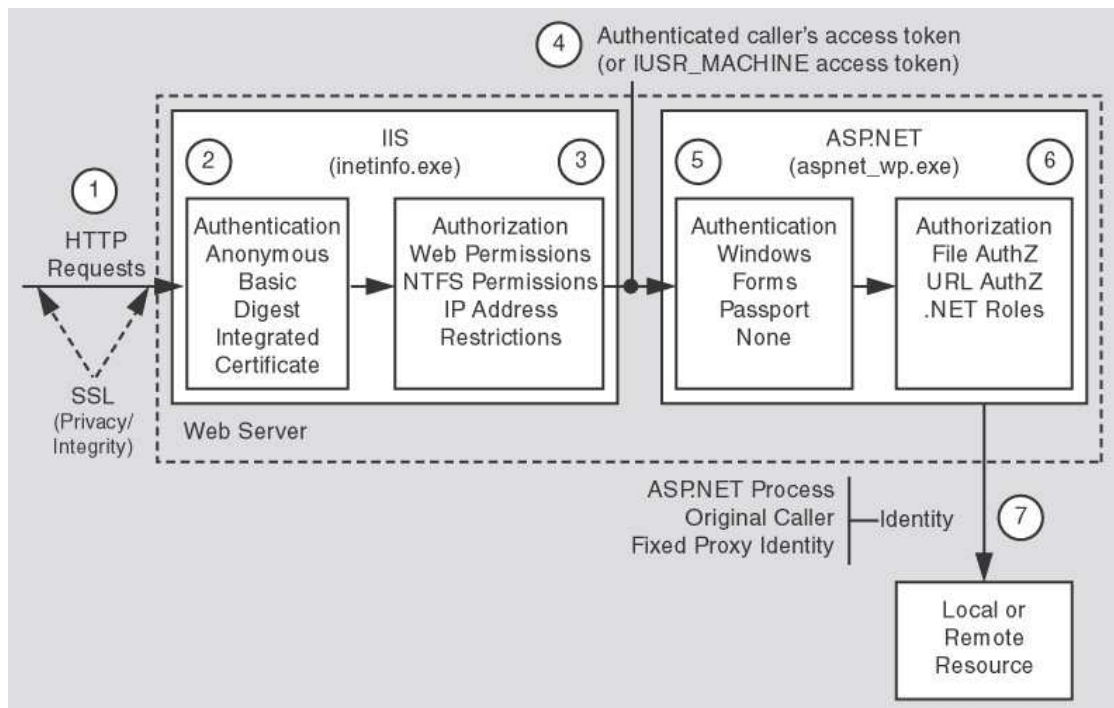
Όπως δείχνει το παράδειγμα, όλοι οι πελάτες επικοινωνούν με Ιστού ASP.NET εφαρμογές μέσω του Microsoft Internet Information Services (IIS). IIS πιστοποιεί την αίτηση, εφόσον απαιτείται, και στη συνέχεια τοποθετεί την πηγή που ζητήθηκε (όπως μια εφαρμογή ASP.NET). Εάν ο πελάτης έχει το δικαίωμα, ο πόρος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



## 4.2 Ασφάλεια

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

### Σχήμα: Η αρχιτεκτονική ασφαλείας του ASP.NET



## ASP.NET Configuration File Security Settings

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

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

## Η ASP.NET πιστοποίηση και IIS Security

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

**Forms Authentication . Έντυπα Authentication.** Η χρήση αυτής της παροχής προκαλεί χωρίς έλεγχο αιτήματα να κατευθυνθούν προς μια συγκεκριμένη μορφή HTML χρησιμοποιώντας ανακατεύθυνση πλευρά του πελάτη. Ο χρήστης μπορεί να παρέχει πιστοποιήσεις σύνδεσης, στη συνέχεια, και μετά τη μορφή πίσω στον server. Εάν η αίτηση πιστοποιεί την αίτηση (με συγκεκριμένες εφαρμογές λογική), τα θέματα ASP.NET ένα cookie που περιέχει τα διαπιστευτήρια ή ένα κλειδί για τον επαναπροσδιορισμό της ταυτότητας του πελάτη. Οι επόμενες αιτήσεις που εκδίδονται με το cookie στο αίτημα, το οποίο σημαίνει ότι οι επόμενες πιστοποιήσεις είναι περιττές.

**Passport Authentication . Authentication διαβατηρίου.** Αυτή είναι μια κεντρική υπηρεσία ελέγχου ταυτότητας που προβλέπεται από τη Microsoft που προσφέρει μια ενιαία δυνατότητα σύνδεσης και την ένταξη των υπηρεσιών για τις συμμετέχουσες τοποθεσίες. Η ASP.NET, σε συνδυασμό με το διαβατήριο του Microsoft ® Software Development Kit (SDK), παρέχει παρόμοιες λειτουργίες όπως Forms Authentication στους χρήστες διαβατηρίου.

**Windows Authentication . Windows Authentication.** Ο φορέας αυτός, χρησιμοποιεί τις δυνατότητες ελέγχου ταυτότητας των IIS. Μετά IIS ολοκληρώνει ταυτότητας του, ASP.NET χρησιμοποιεί την επαλήθευση της ταυτότητας του για να επιτρέψει την πρόσβαση.

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

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

Το ακόλουθο παράδειγμα δείχνει τη σύνταξη του αρχείου διαμόρφωσης:

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

```
</authentication>

<membership default Provider="AspNetMembershipProvider">

  <providers>

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

  </providers>

</membership>
```

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

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

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

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

Το πρώτο βήμα για την ασφάλεια της εφαρμογής μας ξεκινά με το Web διακομιστή οποίος φιλοξενεί την τοποθεσία μας. Στις περισσότερες περιπτώσεις, αυτός ο Web διακομιστής είναι ο Internet Information Server 5.0+, ο οποίος



εγκαθίσταται αυτόματα με τα Windows 2000+ Server και μπορεί να προστεθεί με μη αυτόματο τρόπο στα Windows 2000+ Professional από τα Windows Components. Το IIS μας δίνει αρκετές μεθόδους για να ασφαλίσουμε τη Web τοποθεσία μας. Αυτές οι επιλογές βρίσκονται στην καρτέλα Directory Security (ασφάλεια καταλόγου) στο παράθυρο Properties (ιδιότητες) της τοποθεσίας μας.

### Δημιουργία ενός αρχείου Web.config

Μπορείτε να δημιουργήσετε ένα αρχείο Web.config, χρησιμοποιώντας ένα πρόγραμμα επεξεργασίας κειμένου όπως το Notepad. Πρέπει να δημιουργήσετε ένα αρχείο κειμένου με το όνομα **Web.config** στο ριζικό κατάλογο της εφαρμογής ASP.NET. Το αρχείο Web.config πρέπει να είναι ένα καλά μορφοποιημένο έγγραφο XML και πρέπει να έχει μορφή παρόμοια με τον % SystemRoot% \ Microsoft.NET \ Framework \ % VersionNumber% \ CONFIG \ αρχείο Machine.config. Το αρχείο Web.config πρέπει να περιέχει μόνο καταχωρήσεις για στοιχεία ρύθμισης που παρακάμπτουν τις ρυθμίσεις του αρχείου Machine.config. Τουλάχιστον, το αρχείο Web.config πρέπει να έχουν το στοιχείο **<configuration>** και την **<system.web>** στοιχείο. Τα στοιχεία αυτά θα περιέχουν μεμονωμένα στοιχεία ρύθμισης. Το ακόλουθο παράδειγμα δείχνει ένα ελάχιστο αρχείο Web.config:

```
<? xml version = "1.0" encoding = "utf-8"?>

<configuration>

<system.web>

</ system.web>

</ configuration>
```

Η πρώτη γραμμή του αρχείου Web.config περιγράφει το έγγραφο σε μορφή XML και καθορίζει τον τύπο κωδικοποίησης χαρακτήρων. Αυτή η πρώτη γραμμή πρέπει να είναι η ίδια για όλους, Config αρχεία. Οι γραμμές που ακολουθούν σηματοδοτούν την αρχή και το τέλος της **<configuration>** στοιχείο και το στοιχείο της **<system.web>** αρχείο Web.config. Με τον εαυτό τους, οι γραμμές αυτές δεν κάνουν τίποτα. Ωστόσο, οι γραμμές παρέχουν μια δομή που σας επιτρέπει να προσθέσετε μελλοντικές ρυθμίσεις. Μπορείτε να προσθέσετε την πλειοψηφία του ASP.NET ρυθμίσεις μεταξύ των **<system.web>** και **</ system.web>** γραμμές. Οι γραμμές αυτές σηματοδοτούν την αρχή και το τέλος του ASP.NET ρυθμίσεις.

### 4.2.2 Το αρχείο Web.config

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

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

```
<appSettings/>

<ConnectionStrings>

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

</connectionStrings>
```

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

```
<location path="Admin">

<system.web>

<authorization>

<allow roles="Administrators" />

<deny users="*" />

</authorization>

</system.web>
```

</location>

### 4.3 Cookies

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

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

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

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

### 4.4 Σύνδεση με Βάση Δεδομένων

Μέσα στον κώδικα έχουμε πρόσβαση στην βάση δημιουργώντας μία σύνδεση:

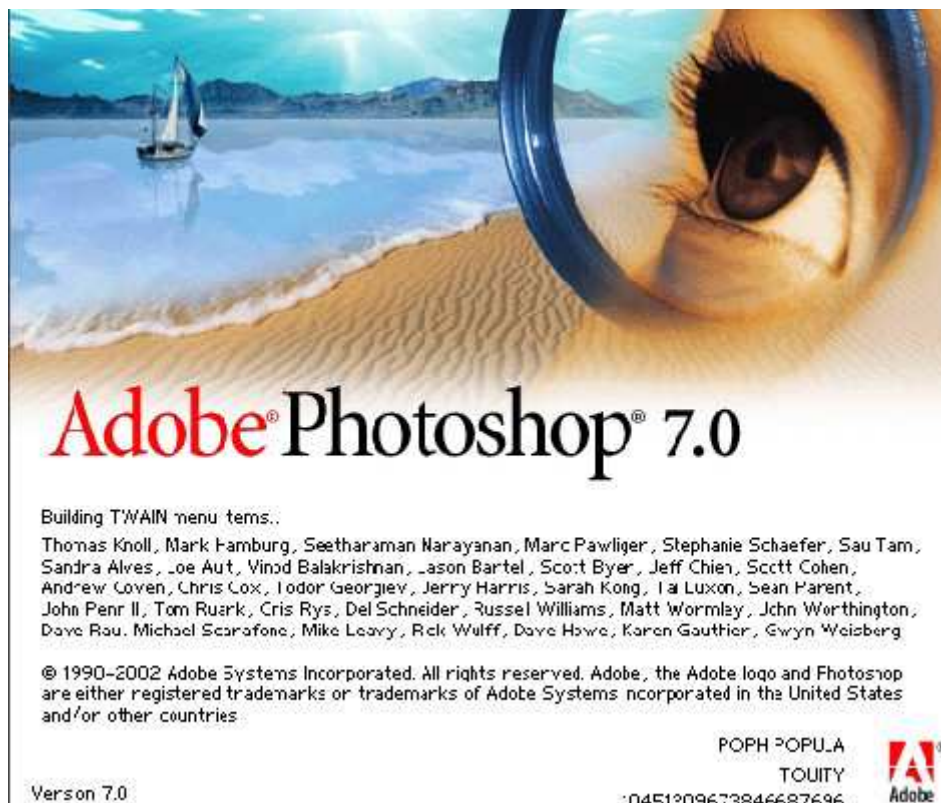
```
<asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="False"
    DataKeyNames="id" DataSourceID="AccessDataSource1"
    EmptyDataText="There are no data records to display.">
    <Columns>
        <asp:BoundField DataField="id" HeaderText="id"
ReadOnly="True"
            SortExpression="id" Visible="False" />
        <asp:ImageField DataImageUrlField="id"
            DataImageUrlFormatString="images/{0}.jpg">
        </asp:ImageField>
        <asp:BoundField DataField="ervtish"
HeaderText="Ερωτήσεις"
            SortExpression="ervtish" />
        <asp:TemplateField SortExpression="id"
HeaderText="Επιλογές">
            <EditItemTemplate>
                <asp:Label ID="Label1" runat="server"
Text='<%# Eval("id") %>'></asp:Label>
            </EditItemTemplate>
            <ItemTemplate>
                <asp:Label ID="Label1" runat="server"
Text='<%# Bind("id") %>' Visible="False"></asp:Label>
                <asp:RadioButtonList ID="RadioButtonList1"
runat="server"
                    DataSourceID="AccessDataSource2"
DataTextField="apantisi"
                    DataValueField="sosti">
                </asp:RadioButtonList>
```

```
runat="server" <asp:AccessDataSource ID="AccessDataSource2"
DataFile="App_Data\DriveTest.mdb"
```

## 5.Κεφάλαιο

Προγράμματα που χρησιμοποιήθηκαν ( **Adobe Photoshop 7.0**)

## 5. Photoshop



Το Photoshop υπήρξε ανέκαθεν το πρότυπο στην επεξεργασία ψηφιακής φωτογραφίας αλλά και στην δημιουργία layout για ιστοσελίδες (template).Μια πολύ χρήσιμη λειτουργία που παρέχει το Photoshop,

είναι τα layers. Φανταστείτε τα layers σαν διαφορετικά επίπεδα το ένα πάνω από το άλλο τα οποία όλα

μαζί δημιουργούν μια εικόνα. (γραφικά ή φωτογραφία). Το Photoshop πέρα από τις εκπληκτικές δυνατότητες που έχει στην επεξεργασία εικόνας, με τα χρόνια έχει εξελιχθεί αρκετά, προσφέροντας στους web designers ένα δυνατό εργαλείο για την σχεδίαση web sites.

Αυτή τη στιγμή αποτελεί ηγέτη της αγοράς (*market leader*) των προγραμμάτων επεξεργασίας εικόνων, και είναι το προϊόν - σήμα κατατεθέν της Adobe Systems.

Χαρακτηρίζεται ως "απαραίτητο" εργαλείο για τους επαγγελματίες γραφίστες και θεωρείται πως προώθησε τις αγορές των Macintosh, και στη συνέχεια των Windows.

### 5.1 Δυνατότητες του Photoshop

Οι εφαρμογές(προγράμματα) ψηφιακής επεξεργασίας εικόνας, ένα τέτοιο πρόγραμμα είναι και το Photoshop, μας δίνουν σήμερα την δυνατότητα να επέμβουμε δυναμικά και με απόλυτα αποτελεσματικό τρόπο στην αρχική φωτογραφία. Συγκεκριμένα μας δίνουν την δυνατότητα:

- Να επεξεργαστούμε, κατ' αρχήν, είτε ολόκληρη την εικόνα είτε μέρος της, που είναι εφικτό να ορισθεί με σαφήνεια.
- Να αλλάξουμε τις διαστάσεις της εικόνας, τόσο κρατώντας τις αρχικές τις αναλογίες, όσο και παραμορφώνοντας την.
- Να διορθώσουμε τις λανθασμένες λήψεις με την χρήση εντολών απόλυτου ελέγχου της φωτεινότητας και του contrast, ακόμα και για τους μεσαίους τόνους (ρύθμιση gamma).
- Να αναπλάσουμε κατεστραμμένες περιοχές της φωτογραφίας.
- Να αφαιρέσουμε κάποιο τμήμα της φωτογραφίας και να το αντικαταστήσουμε με κάτι άλλο.

Επίσης, υπάρχει η δυνατότητα να δημιουργήσουμε συνθέσεις με κομμάτια που προέρχονται από πολλές διαφορετικές φωτογραφίες.

- Να ρετουσάρουμε περιοχές της φωτογραφίας αποτελεσματικά και με ακρίβεια με την χρήση ειδικών εργαλείων.
- Να επιχρωμάσουμε ασπρόμαυρες φωτογραφίες.
- Να δημιουργήσουμε υφές και πρωτότυπα φόντα.

- Να αλλάξουμε την προοπτική της φωτογραφίας.
- Να αλλοιώσουμε την αρχική εικόνα με την βοήθεια εντολών που βρίσκονται κάτω από την γενική κατηγορία: φίλτρα(filters). Τα φίλτρα αυτά άλλοτε έχουν σχέση και αναλογίες με πραγματικά φωτογραφικά φίλτρα, άλλοτε απλώς ειδικευμένες εντολές.

## 5.2 Η εργαλειοθήκη του Photoshop



1. Ορθογώνιο πλαίσιο επιλογής (Rectangular) – Elliptical Marquee Tool κ.ά.
2. Μετακίνηση
3. Λάσο (lasso)
4. Μαγικό ραβδί (magic wand)
5. Κοπή (crop)
6. Τεμαχισμός (slice)
7. Αερογράφος (Airbrush)
8. Πινέλο (Paintbrush)
9. Σφραγίδα (clone stamp)
10. Πινέλο ιστορικού (history brush)
11. Γόμα (eraser)

12. Ντεγκραντέ (Gradient) – Κουβάς (Paint Bucket)
13. Εργαλείο μουτζουρώματος (smudge)–Θολώματος (Blur)–Όξυνσης (Sharpen)
14. Εργαλείο κάλυψης (dodge) – Καψίματος (Burn) – Σπόγγου (Sponge)
15. Επιλογή συστατικών διαδρομής
16. Κείμενο
17. Πένα
18. Ορθογώνιο
19. Σημειώσεις
20. Σταγονόμετρο
21. Χέρι
22. Ζουμ

### 5.3 Τα εργαλεία επιλογής

Με τα εργαλεία επιλογής μπορούμε να επιλέξουμε (απομονώσουμε) ένα κομμάτι της εικόνας (κάποια pixels) και μετά να το μετακινήσουμε, αντιγράψουμε ή να του εφαρμόσουμε διάφορες τροποποιήσεις.

#### **Το Ορθογώνιο πλαίσιο επιλογής (Rectangular Marquee Tool):**

Χρησιμοποιείται για να επιλέξουμε ένα ορθογώνιο τμήμα της εικόνας.

#### **Το Ελλειπτικό εργαλείο επιλογής (Elliptical Marquee Tool):**

Χρησιμοποιείται για να επιλέξουμε ένα κυκλικό (ελλειπτικό) τμήμα μιας εικόνας π.χ. μια μπάλα.

Εμφανίζεται με δεξί κλικ πάνω στο Ορθογώνιο εργαλείο επιλογής.

#### **Το λάσο:**

Μπορούμε να σύρουμε με το λάσο (lasso) μέσα στην εικόνα για να επιλέξουμε περιοχές ελεύθερης

μορφής και το σχήμα της επιλογής εξαρτάται από τη διαδρομή που κάνουμε όταν σύρουμε. Όταν

αφήσουμε το πλήκτρο του ποντικιού, η επιλογή μας ολοκληρώνεται. Με δεξί κλικ πάνω στο εργαλείο,

εμφανίζονται τρεις επιλογές: Το απλό λάσο (Lasso tool), το πολυγωνικό (Polygonal) και το μαγνητικό (magnetic).

#### **Το πολυγωνικό λάσο:**

Με το πολυγωνικό λάσο (polygonal lasso), μπορούμε να κάνουμε πολυγωνικές επιλογές, δηλ.

επιλογές που αποτελούνται από ευθύγραμμα τμήματα. Κάνουμε διαδοχικά κλικ στα σημεία που



Θέλουμε και για να κλείσει το περίγραμμα επιλογής, κάνουμε διπλό κλικ ή τοποθετούμε τον δείκτη του ποντικιού κοντά στο σημείο εκκίνησης, ώστε να εμφανισθεί ένας μικρός κύκλος και μετά κάνουμε κλικ για να ολοκληρωθεί το περίγραμμα επιλογής.

### **Το μαγνητικό λάσο:**

Με το μαγνητικό λάσο (magnetic lasso) κάνουμε διαδοχικά κλικ σε κοντινά σημεία μέχρι να ξαναγυρίσουμε στο σημείο εκκίνησης για να κλείσουμε την επιλογή μας. Για να τροποποιήσουμε την απόδοση των εργαλείων λάσο, υπάρχει το πλαίσιο κειμένου Feather (περίγραμμα εξομάλυνσης) και το πλαίσιο ελέγχου Anti-aliased (εξομάλυνση). Αν το πλαίσιο ελέγχου Anti-aliased είναι απενεργοποιημένο, τα περιγράμματα θα έχουν οδοντωτές άκρες. Η τιμή που καταχωρούμε στο πλαίσιο κειμένου Feather καθορίζει την ακτίνα εξομάλυνσης σε pixels (ξεθώριασμα στις άκρες).

### **Το μαγικό ραβδί:**

Το χρησιμοποιούμε συνήθως, συμπληρωματικά με το λάσο για να βάλουμε μέσα ορισμένες μικρές περιοχές που μας ξέφυγαν απ' το λάσο αλλά και για να επιλέξουμε γρήγορα κάποιο τμήμα της εικόνας που ξεχωρίζει χρωματικά από την υπόλοιπη εικόνα. Tolerance: Όσο μεγαλύτερη τιμή βάλουμε, τόσο περισσότερες παραλλαγές του ίδιου χρώματος θα επιλεγούν. Μικρή τιμή επιλέγει αυστηρότερα δηλ. μικρότερη περιοχή. Contiguous: Επιλεγμένο, επιλέγει μόνο γειτονικά ίδια ή παρόμοια χρώματα.

### **Το Εργαλείο Σταγονόμετρου:**

Με το εργαλείο σταγονόμετρου μπορούμε να αντιγράψουμε χρώματα από μια εικόνα και να τα εφαρμόσουμε σαν χρώματα προσκηνίου ή παρασκηνίου. Κάνουμε κλικ στο εργαλείο Eyedropper Tool της εργαλειοθήκης και μετά κάνουμε κλικ στο σημείο της εικόνας που περιέχει το χρώμα που θέλουμε να χρησιμοποιήσουμε.

## **5.4 Τα Εργαλεία Ζωγραφικής**

**Το μολύβι (pencil)**, με το οποίο σχεδιάζουμε γραμμές με κοφτό περίγραμμα και σε όποιο πάχος θέλουμε.

**Το πινέλο (paintbrush)**, με το οποίο σχεδιάζουμε μαλακές γραμμές με κάπως πιο θολό περίγραμμα.

**Τον αερογράφο (airbrush)**, με τον οποίο ζωγραφίζουμε μαλακές γραμμές, όπως με το πινέλο, αλλά με τον αερογράφο μπορούμε να εκτοξεύουμε χρώμα ακόμα και όταν δεν τον σέρνουμε. Σέρνοντας τον αερογράφο, δημιουργούμε σκιές. Για να δημιουργήσουμε μια απόλυτα οριζόντια ή κατακόρυφη ευθεία γραμμή, επιλέγουμε ένα εργαλείο ζωγραφικής, κρατάμε πατημένο το πλήκτρο Shift και σύρουμε το ποντίκι οριζόντια ή κατακόρυφα αντίστοιχα.

### **Επιπλέον επιλογές στα εργαλεία ζωγραφικής:**

Διαλέγουμε Brush, Mode, Opacity κατά τα γνωστά. Στη δεξιά πάνω άκρη της γραμμής επιλογών του εργαλείου, εμφανίζεται ένα πινέλο το οποίο πατώντας το βγάζει το παράθυρο Brush Dynamics.

Επιλέγοντας:

**Size: Fade** κάνει πινελιά που τελειώνει σε μύτη.

**Opacity: Fade** κάνει το χρώμα της πινελιάς να σβήνει προς το τέλος της πινελιάς. Τα βήματα που βάζουμε σε κάθε επιλογή, αφορούν το μέγεθος της πινελιάς.

**Color : Fade** Ξεκινά η πινελιά από το χρώμα προσκηνίου και καταλήγει στο χρώμα φόντου. Τα βήματα αφορούν το πόσο γρήγορα θα αλλάξει το χρώμα.

## **5.5 Τα Εργαλεία Επεξεργασίας**

- Το εργαλείο μουτζουρώματος (smudge) χρησιμοποιείται για να απλώνουμε το χρώμα σε κάποια περιοχή.
- Το εργαλείο θολώματος (blur) θολώνει τα άκρα ανάμεσα στα χρώματα, έτσι ώστε τα χρώματα να συγχωνευτούν στα όριά τους. π.χ. εφαρμόζοντάς το πάνω στο φόντο μιας εικόνας, διαχωρίζει καλύτερα το θέμα απ' το φόντο.

- Το εργαλείο όξυνσης (sharpen) αυξάνει την ποσότητα έντασης μεταξύ των χρωμάτων και αναδεικνύει τα τελειώματα των επιφανειών.
- Το εργαλείο κάλυψης (dodge) κάνει φωτεινότερη μια περιοχή.
- Το εργαλείο καψίματος (burn) κάνει σκουρότερη μια περιοχή.
- Το εργαλείο σπόγγου (sponge) διορθώνει την ένταση των χρωμάτων. Με επιλεγμένο το Desaturate ξεθωριάζει τα χρώματα, με επιλεγμένο το Saturate κάνει πιο έντονα τα χρώματα.

### 5.6 Οι Στρώσεις (Layers)

Οι στρώσεις μιας εικόνας του PhotoShop είναι σαν τις διαφάνειες στις οποίες σχεδιάζουμε διάφορα αντικείμενα και όταν τις τοποθετούμε τη μια πάνω στην άλλη, φαίνεται η ολοκληρωμένη εικόνα σαν ένα σύνολο. Η σειρά των στρώσεων στην παλέτα Layers δείχνει και τη σειρά εμφάνισής τους στην εικόνα, δηλ. η στρώση που εμφανίζεται στην κορυφή της παλέτας είναι και η κορυφαία στρώση της εικόνας. Μπορούμε να επεξεργαζόμαστε μόνο μία στρώση κάθε φορά, την ενεργό στρώση. Η ενεργός στρώση ξεχωρίζει γιατί είναι έντονα φωτισμένη στην παλέτα Layers και έχει αριστερά της το εικονίδιο ενός μικρού πινέλου. Για να γίνει ενεργή μια άλλη στρώση, κάνουμε κλικ στον όνομά της στην παλέτα

Layers. Το εικονίδιο του ματιού που βρίσκεται δίπλα στο όνομα μιας στρώσης, σημαίνει ότι η στρώση αυτή είναι ορατή. Για να κρύψουμε μια στρώση, πατάμε πάνω στο εικονίδιο του ματιού της. Αν κρύψουμε τη στρώση φόντου, θα εμφανισθεί ένα μοτίβο σκακιέρας που αντιπροσωπεύει τις διαφανείς

περιοχές των ορατών στρώσεων. Με διπλό κλικ πάνω στο όνομα μιας στρώσης, εμφανίζεται το παράθυρο Blending Options από όπου μπορούμε να προσθέσουμε διάφορα εφέ όπως σκιά, ανάγλυφη υφή κ.ά. καθώς και να μειώσουμε την αδιαφάνεια της στρώσης (opacity). Με δεξί κλικ πάνω στο όνομα μιας στρώσης μπορούμε να διαγράψουμε τη στρώση (Delete) ή να της αλλάξουμε το όνομα

(Layer Properties)

### 5.7 Γέμισμα περιοχής με άλλο χρώμα (το Εργαλείο του Κουβά)

Με το εργαλείο κουβά (Paint Bucket Tool) μπορούμε να γεμίσουμε μια συνεχή περιοχή χρώματος μ' ένα άλλο χρώμα, πατώντας σ' ένα σημείο της περιοχής. Το PhotoShop χρησιμοποιεί το χρώμα προσκηνίου για να γεμίσει τη συνεχή περιοχή. Η τιμή του πλαισίου κειμένου Tolerance καθορίζει τα pixels της εικόνας που θα επηρεαστούν από τον κουβά. Αν το πλαίσιο ελέγχου Anti-aliased είναι

ενεργό, θα εξομαλύνονται τα άκρα της γεμισμένης περιοχής. Για να βρούμε τη σωστή τιμή του πλαισίου κειμένου Tolerance, πρέπει να κάνουμε αρκετές δοκιμές και να χρησιμοποιούμε την αναίρεση. Μια άλλη δυνατότητα χρωματισμού μιας συνεχούς περιοχής, είναι να την επιλέξουμε και μετά να την χρωματίσουμε με την εντολή Fill... του μενού Edit, οπότε θα έχουμε περισσότερες επιλογές στη διάθεσή μας.

### 5.8 Το Εργαλείο Ντεγκραντέ

Με το εργαλείο ντεγκραντέ (Gradient Tool) μπορούμε να γεμίσουμε μια επιλεγμένη περιοχή με μια διαβάθμιση χρωμάτων, που να ξεκινάει από ένα χρώμα και να καταλήγει σε κάποιο άλλο. Εξ ορισμού, τα δύο αυτά χρώματα είναι το χρώμα προσκηνίου και το χρώμα φόντου. Στο μενού του εργαλείου υπάρχουν τα εξής πέντε πλήκτρα για λειτουργίες ντεγκραντέ: Linear, Radial, Angle, Reflected και Diamond. Αν δεν επιλέξουμε κάποιο τμήμα της εικόνας πριν χρησιμοποιήσουμε το εργαλείο ντεγκραντέ, το PhotoShop θα γεμίσει ολόκληρη την εικόνα ή αν εργαζόμαστε με κάποια στρώση, θα γεμίσει ολόκληρη τη στρώση. Η εξ ορισμού επιλογή είναι Foreground to Background, δηλ. σταδιακή μετάβαση από το χρώμα προσκηνίου στο χρώμα παρασκηνίου.

### 5.9 Τα Φίλτρα (Filters) του PhotoShop

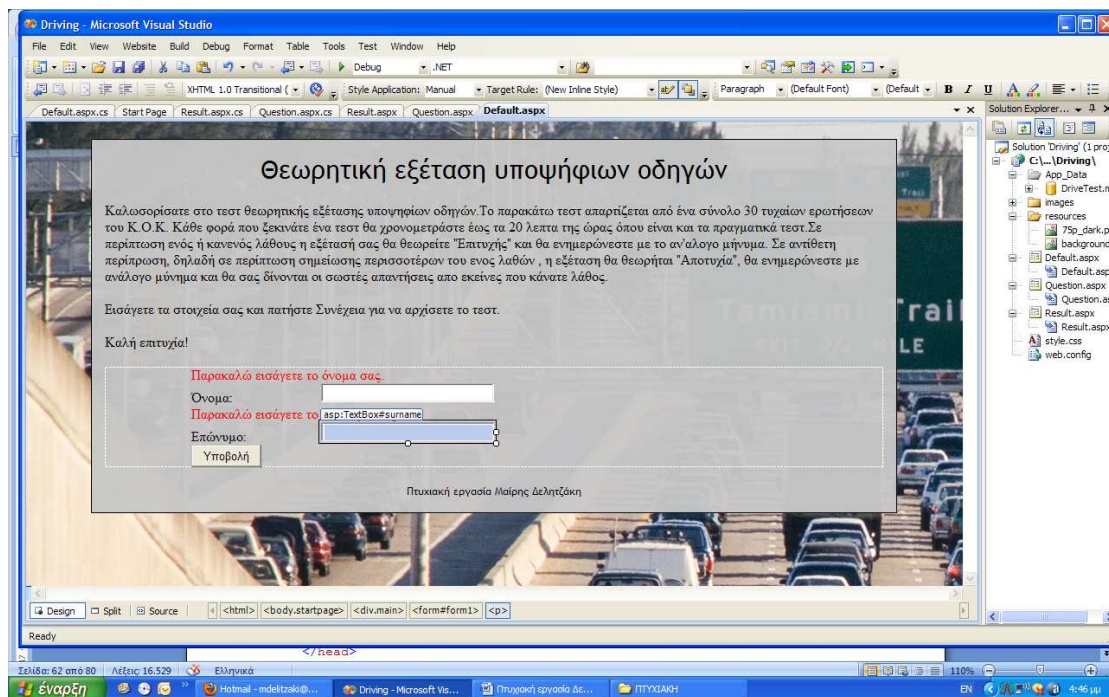
Τα φίλτρα του PhotoShop κάνουν μια λειτουργία ανάλογη μ' αυτή των φωτογραφικών φίλτρων : φιλτράρουν ή διαθλούν το φως, με αποτέλεσμα να τροποποιούν την εικόνα. Όλα τα φίλτρα του PhotoShop βρίσκονται στο μενού Filter και μπορούμε να ακυρώσουμε (αναιρέσουμε) αμέσως τη χρήση κάποιου φίλτρου, αν το αποτέλεσμα δεν μας αρέσει. Μπορούμε να δούμε προκαταβολικά (σε προεπισκόπηση) τα αποτελέσματα των πιο βασικών φίλτρων και μπορούμε να εφαρμόσουμε πολλά φίλτρα στη σειρά ή και να χρησιμοποιήσουμε το ίδιο φίλτρο πολλές φορές. Όταν είναι επιλεγμένο κάποιο τμήμα της εικόνας, το φίλτρο επιδρά μόνο στην επιλεγμένη περιοχή, ενώ όταν δεν είναι επιλεγμένο κάποιο τμήμα της εικόνας, το φίλτρο επιδρά σ' ολόκληρη την εικόνα. Για να δημιουργήσουμε ομαλές μεταβάσεις μεταξύ των φιλτραρισμένων και των αφιλτράριστων περιοχών μιας εικόνας, μπορούμε να θολώσουμε το περίγραμμα επιλογής με την εντολή Feather... του μενού Select. Για να εφαρμόσουμε ξανά το τελευταίο φίλτρο, μπορούμε να επιλέξουμε Last Filter από το μενού Filter. Για να αναμίξουμε μια φιλτραρισμένη εικόνα με την αφιλτράριστη αρχική εικόνα, μπορούμε να χρησιμοποιήσουμε την εντολή Fade... από το μενού Filter. Στο πλαίσιο διαλόγου Fade μπορούμε να ρυθμίσουμε την αδιαφάνεια με τον ροοστάτη (πλαίσιο κειμένου) Opacity και τον τρόπο ανάμιξης από την πτυσσόμενη λίστα Mode. Για να βλέπουμε το αποτέλεσμα της εφαρμογής του φίλτρου σε προεπισκόπηση, πρέπει να είναι επιλεγμένο το πλαίσιο ελέγχου Preview. Η εντολή αυτή πρέπει να χρησιμοποιείται αμέσως μετά την εφαρμογή του φίλτρου. Πρέπει να έχουμε υπόψη μας ότι η πραγματική ομορφιά των φίλτρων του PhotoShop βρίσκεται στον συνδυασμό και την εφαρμογή τους σε μικρά, επιλεγμένα τμήματα της εικόνας.

## 6. Κεφάλαιο – Η εκπαιδευτική εφαρμογή

### 6.1 Τα μέρη της εφαρμογής αναλυτικά.

Στην εφαρμογή μας υπάρχουν τα βασικά μέρη τα οποία την απαρτίζουν. Αρχικά υπάρχει η αρχική σελίδα της εφαρμογής όπου ο κάθε χρήστης θα μπορεί να διαβάσει κάποια ενημερωτικά θέματα σχετικά με τον Κώδικα Οδικής Κυκλοφορίας (Κ.Ο.Κ.) . Όπως και κάποια βασικά στοιχεία για το πώς λειτουργεί η συγκεκριμένη εφαρμογή. Ουσιαστικά υπάρχει μια εικόνα φόντου και μπροστά από αυτή υπάρχει το κείμενο που εξηγεί αναλυτικά όλη την λειτουργία της εφαρμογής. Στο κάτω μέρος της αρχικής σελίδας υπάρχει κουμπί "Υποβολή" που θα οδηγήσει τον χρήστη στην επόμενη σελίδα όπου είναι και το τεστ.

**Στο περιβάλλον της Visual Studio 2008 απεικονίζεται καθαρά η αρχική σελίδα της εφαρμογής.**



**Υπάρχει ένα κείμενο που εξηγεί την λειτουργία της εφαρμογής , δυο Textboxes για να εισάγει κάθε χρήστης τα στοιχεία του και ένα κουμπί που ξεκινά την εφαρμογή.**

## Ο κώδικας που χρειάστηκε

```
@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta http-equiv="content-type" content="text/html; charset=utf-
8" />
    <title>Τεστ οδήγησης</title>
    <link href="style.css" media="all" type="text/css"
rel="stylesheet" />
</head>
<body class="startpage">
    <div class="main">
        <h1>Θεωρητική εξέταση υποψηφίων οδηγών</h1>
        <p>Καλωσορίσατε στο τεστ θεωρητικής εξέτασης υποψηφίων οδηγών. Το
παρακάτω τεστ απαρτίζεται
        από ένα σύνολο 30 τυχαίων ερωτήσεων του Κ.Ο.Κ. Κάθε φορά που
ξεκινάτε ένα τεστ θα χρονομετράστε
        έως τα 20 λεπτα της ώρας όπου είναι και τα πραγματικά τεστ. Σε
περίπτωση ενός ή κανενός λάθους
        η εξέτασή σας θα θεωρείτε "Επιτυχής" και θα ενημερώνεστε με το
αν'αλογο μήνυμα. Σε αντίθετη
        περίπτωση, δηλαδή σε περίπτωση σημείωσης περισσότερων του ενός
λαθών , η εξέταση θα θεωρήται "Αποτυχία",
        θα ενημερώνεστε με ανάλογο μήνυμα και θα σας δίνονται οι σωστές
απαντήσεις απο εκείνες που κάνατε λάθος.
        </p>
        <p>
        Εισάγετε τα στοιχεία σας και πατήστε Συνέχεια για να αρχίσετε το
τεστ.</p>
        <p>Καλή επιτυχία!</p>

        <form id="form1" runat="server">
            <p>
                <asp:RequiredFieldValidator runat="server" ID="nameVal"
ControlToValidate="name" ErrorMessage="Παρακαλώ εισάγετε το όνομα
σας. " Display="Dynamic"/>
            </p>
            <p>
                <asp:Label id="lblName" Text="Όνομα:" runat="server"
Width="150px"></asp:Label>
                <asp:TextBox id="name" runat="server" Width="200px"
CausesValidation="True" />
            </p>
            <p>
                <asp:RequiredFieldValidator runat="server" ID="surnameVal"
ControlToValidate="surname" ErrorMessage="Παρακαλώ εισάγετε το
επώνυμό σας." Display="Dynamic"/>
            </p>
        </form>
    </div>
</body>
</html>
```

```
</p>
<p>
    <asp:Label id="lblSurname" Text="Επώνυμο:" runat="server"
Width="150px"></asp:Label>
    <asp:TextBox id="surname" Width="200px" runat="server"
CausesValidation="True" />
</p>
<p>
    <asp:Button id="btnSubmit" Text="Υποβολή" runat="server"
onclick="btnSubmit_Click" />
</p>
</form>

<div class="footer">
    <p>Πτυχιακή εργασία Μαίρης Δελητζάκη</p>
</div>

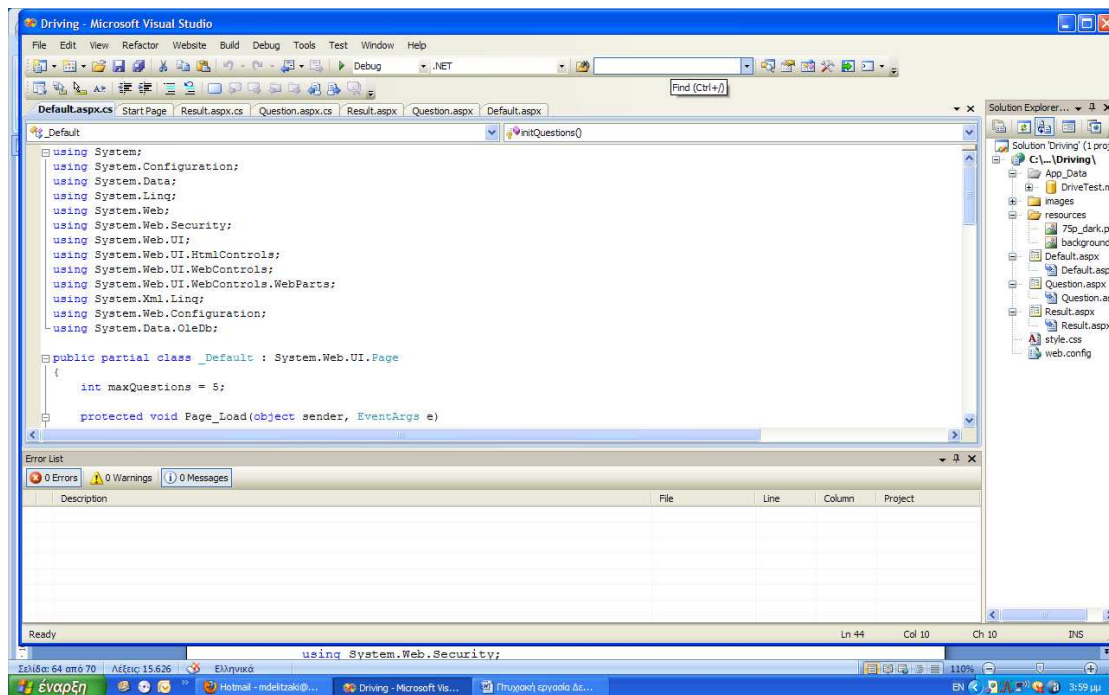
</div>
</body>
</html>
```

### Το αρχείο aspx.cs

Είναι το αρχείο που έχει την λειτουργία της εφαρμογής , κάνει την σύνδεση με την βάση δεδομένων την τυχαία επιλογή των ερωτήσεων των εικόνων των πολλαπλών επιλογων και τις εμφανίζει σε ένα πίνακα διαδοχικά με την επίδραση του χρήστη και με την επιλογή του κουμπιού "επομένη".... Ακόμη παρέχεται και η δυνατότητα να ξαναδεί την προηγούμενη ερώτηση πατώντας το κουμπί "προηγούμενη" και επίσης εμφανίζει το χρονόμετρο καθ' όλη την διάρκεια του τεστ. Και το ονοματεπώνυμο του χρήστη.

Στην παρακάτω εικόνα απεικονίζεται το source τμήμα στο περιβάλλον της Visual Studio 2008





## Ο κώδικας που χρειάστηκε

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

public partial class _Default : System.Web.UI.Page
{
    int maxQuestions = 5;

    protected void Page_Load(object sender, EventArgs e)
    {
    }
}
```

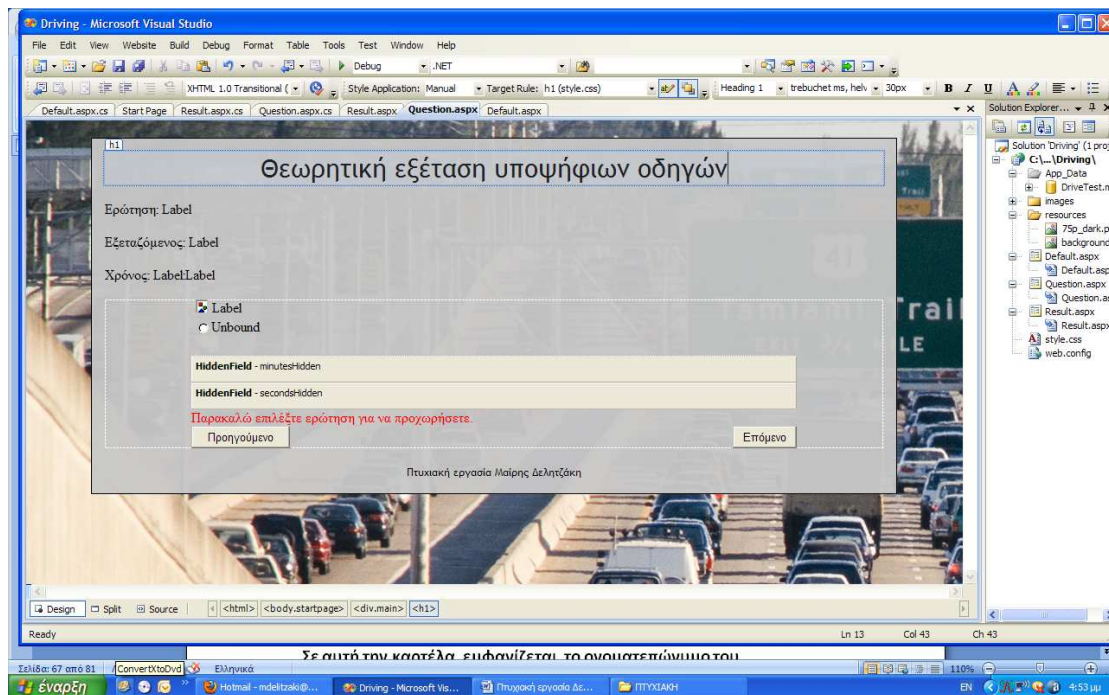
```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    Session["userName"] = name.Text;
    Session["userSurname"] = surname.Text;
    initQuestions();
}

protected void initQuestions()
{
    Configuration rootWebConfig =
WebConfigurationManager.OpenWebConfiguration("/Driving");
   ConnectionStringSettings connectionString =
rootWebConfig.ConnectionStrings.ConnectionStrings["DrivingDB"];
    OleDbConnection connection = new
OleDbConnection(connectionString.ToString());
    connection.Open();
    OleDbCommand command = new OleDbCommand("SELECT TOP " +
maxQuestions + " id FROM `erwtiseis` ORDER BY RND(INT(NOW*ID)-
NOW*ID)", connection);

    OleDbDataReader myReader = command.ExecuteReader();
    int i=0;
    Object[,] questionArray = new Object[maxQuestions,2];

    while(myReader.Read())
    {
        questionArray[i,0] = myReader["id"];
        questionArray[i, 1] = -1;
        i++;
    }
}
```

### Η καρτέλα Questions.aspx



Σε αυτή την καρτέλα εμφανίζεται το ονοματεπώνυμο του εξεταζομένου, ο χρόνος που απομένει, η ερώτηση, η εικόνα που την περιγράφει, το radiobuttonlist με τις πολλαπλές επιλογές που καλείτε ο χρήστης να απαντήσει και τα δυο κουμπιά πλοήγησης στην επόμενη και την προηγούμενη ερώτηση.

## Ο κώδικας που χρειάστηκε

```
<%@ Page Language="C#" Debug="true" AutoEventWireup="true"
CodeFile="Question.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
```

```

    <meta http-equiv="content-type" content="text/html; charset=utf-
8" />
    <title>Τεστ οδήγησης</title>
    <link href="style.css" media="all" type="text/css"
rel="stylesheet" />
</head>
<body class="startpage">
    <div class="main">
        <h1>Θεωρητική εξέταση υποψήφιων οδηγών</h1>

        <p>
Ερώτηση: <asp:Label ID="qNoLbl" runat="server" Text="Label"
/>
        </p>
        <p>
Εξεταζόμενος: <asp:Label ID="nameLbl" runat="server"
Text="Label" />
        </p>
        <p>
Χρόνος: <asp:Label ID="minutesLbl" runat="server"
Text="Label" />:<asp:Label ID="secondsLbl" runat="server"
Text="Label" />
        </p>
        <form id="form1" runat="server">

        <p>
            <asp:Image ID="questionImg" runat="server" CssClass="left" />
            <asp:Label ID="questionLbl" runat="server" Text="Label" />
        </p>
        <p>
            <asp:RadioButtonList ID="questionRadio" runat="server"
CausesValidation="true"/>

            <br class="clear" />
        </p>

        <p>
            <asp:HiddenField ID="minutesHidden" runat="server" />
            <asp:HiddenField ID="secondsHidden" runat="server" />
        </p>

        <p>
            <asp:RequiredFieldValidator runat="server"
ID="questionRadioVal" ControlToValidate="questionRadio"
ErrorMessage="Παρακαλώ επιλέξτε ερώτηση για να προχωρήσετε."
Display="Dynamic"/>
        </p>

        <p>
            <asp:Button ID="PrevButton" runat="server" Text="Προηγούμενο"
CssClass="left"

```

```
        onclick="PrevButton_Click" />
        <asp:Button ID="NextButton" runat="server" Text="Επόμενο"
        CssClass="right"
        onclick="NextButton_Click" />
        <br class="clear" />
    </p>

</form>

<div class="footer">
    <p>Πτυχιακή εργασία Μαίρης Δελητζάκη</p>
</div>

</div>

<script type="text/javascript">
<!--
//
var minutesHolder = document.getElementById("minutesLbl");
var secondsHolder = document.getElementById("secondsLbl");

var minutesHidden = document.getElementById("minutesHidden");
var secondsHidden = document.getElementById("secondsHidden");

var minutes = parseInt(minutesHolder.innerHTML);
var seconds = parseInt(secondsHolder.innerHTML);

function display(){
    var secStr;

    if(seconds < 10){
        secStr = "0" + seconds;
    }else{
        secStr = seconds;
    }
    minutesHolder.innerHTML = minutes;
    secondsHolder.innerHTML = secStr;
}

function tick(){
    seconds--;

    if (seconds<0){
        seconds = 59;
        minutes--;
    }

    minutesHidden.value = minutes;
    secondsHidden.value = seconds;
    display();

    if(minutes <=0 && seconds <=0){
        alert("0 χρόνος τελείωσε.");
    }
}
-->
</script>
```

```
        window.location.href = "http://" +
window.location.host + "/Result.aspx";

        }else{
            setTimeout("tick()",1000);
        }

    }

    display();
    tick() ;
    -->
</script>

</body>
</html>
```

### Το αρχείο questions.aspx.cs

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

public partial class _Default : System.Web.UI.Page
{
    String username;
    Object[,] questionsArray;
    int questionNo, minutes, seconds;

    protected void Page_Load(object sender, EventArgs e)
    {
        username = Session["userName"] + " " +
Session["userSurname"];
        questionsArray = (Object[,]) Session["questionArray"];
        questionNo = Convert.ToInt32(Session["currentQuestion"]);
        minutes = Convert.ToInt32(Session["minutes"]);
        seconds = Convert.ToInt32(Session["seconds"]);
    }
}
```

```

        nameLbl.Text = username;
        qNoLbl.Text = (questionNo + 1) + "/" +
(questionArray.Length/2);
        minutesLbl.Text = Convert.ToString(minutes);
        secondsLbl.Text = Convert.ToString(seconds);
        minutesHidden.Value = Convert.ToString(minutes);
        secondsHidden.Value = Convert.ToString(seconds);

        initRadio( Convert.ToInt32(questionArray[questionNo,0]) );
        if (Convert.ToInt32(questionArray[questionNo, 1]) != -1)
            questionRadio.SelectedValue =
Convert.ToString(questionArray[questionNo, 1]);
    }

    protected void initRadio(int questionId)
    {
        Configuration rootWebConfig =
WebConfigurationManager.OpenWebConfiguration("/Driving");
       ConnectionStringSettings connectionString =
rootWebConfig.ConnectionStrings.ConnectionStrings["DrivingDB"];
        OleDbConnection connection = new
OleDbConnection(connectionString.ToString());

        connection.Open();
        OleDbCommand command = new OleDbCommand("SELECT ervtish FROM
erwtiseis WHERE id="+questionId, connection);
        OleDbDataReader myReader = command.ExecuteReader();
        myReader.Read();
        questionLbl.Text = (String)myReader["ervtish"];
        questionImg.ImageUrl = "./images/" + questionId + ".jpg";

        myReader.Close();

        if (questionRadio.Items.Count == 0)
        {
            command = new OleDbCommand("SELECT id, apantisi FROM
epiloges WHERE erotisi_id=" + questionId, connection);
            myReader = command.ExecuteReader();

            while (myReader.Read())
            {
                ListItem li = new ListItem();
                li.Text = (String)myReader["apantisi"];
                li.Value = Convert.ToString(myReader["id"]);
                questionRadio.Items.Add(li);
            }

            myReader.Close();
        }
        connection.Close();
    }

    protected void NextButton_Click(object sender, EventArgs e)
    {

```

```
questionsArray[questionNo, 1] = questionRadio.SelectedValue;
Session["questionArray"] = questionsArray;

questionNo++;
Session["currentQuestion"] = questionNo;

if (questionsArray.Length / 2 <= questionNo)
{
    Server.Transfer("Result.aspx");
}

Session["minutes"] = Request["minutesHidden"];
Session["seconds"] = Request["secondsHidden"];

Server.Transfer("Question.aspx");
}
protected void PrevButton_Click(object sender, EventArgs e)
{
    if (questionNo > 0)
    {
        questionsArray[questionNo, 1] =
questionRadio.SelectedValue;
        Session["questionArray"] = questionsArray;

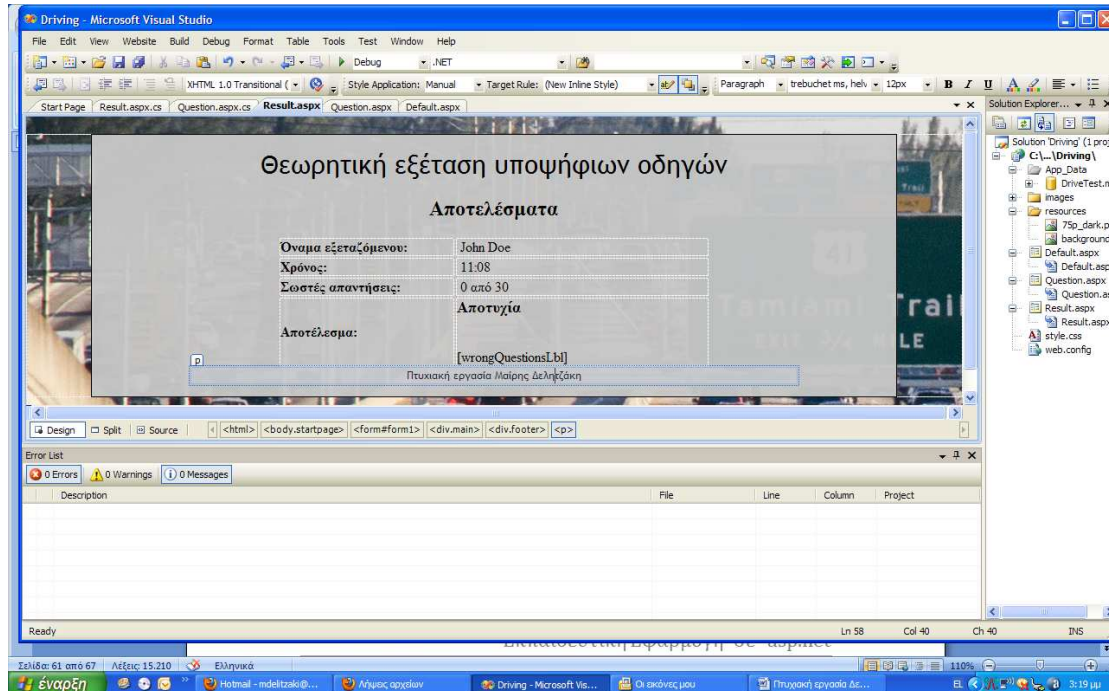
        questionNo--;
        Session["currentQuestion"] = questionNo;

        Session["minutes"] = Request["minutesHidden"];
        Session["seconds"] = Request["secondsHidden"];

        Server.Transfer("Question.aspx");
    }
}
}
```



Τα αποτελέσματα κάθε τεστ στην καρτέλα Result.aspx



Σε αυτή την καρτέλα υπάρχει ένας δυναμικός πίνακας που δέχεται το ονοματεπώνυμο του χρήστη ο οποίος εκτελεί την προκειμένη στιγμή το τεστ, τον χρόνο στο οποίο έχει τελειώσει το τεστ τον αριθμό των απαντήσεων που απαντήθηκαν σωστά και τις σωστές απαντήσεις από τις λάθος απαντήσεις που έδωσε ο χρήστης.

Ο κώδικας που χρειάστηκε

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Result.aspx.cs" Inherits="_Default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <meta http-equiv="content-type" content="text/html; charset=utf-
8" />
  <title>Τεστ οδήγησης</title>
```

```

    <link href="style.css" media="all" type="text/css"
rel="stylesheet" />
</head>
<body class="startpage">
    <form id="form1" runat="server">
        <div class="main">
            <h1>Θεωρητική εξέταση υποψήφιων οδηγών</h1>
            <h2>Αποτελέσματα</h2>

            <table class="result">
                <tr>
                    <th>
                        Όνομα εξεταζόμενου:
                    </th>
                    <td>
                        &nbsp;<asp:Label ID="nameLbl" runat="server"
Text="John Doe"></asp:Label>
                    </td>
                </tr>

                <tr>
                    <th>
                        Χρόνος:
                    </th>
                    <td>
                        &nbsp;<asp:Label ID="timeLbl" runat="server"
Text="11:08"></asp:Label>
                    </td>
                </tr>

                <tr>
                    <th>
                        Σωστές απαντήσεις:
                    </th>
                    <td>
                        &nbsp;<asp:Label ID="correctLbl" runat="server"
Text="0"></asp:Label> &nbsp;<asp:Label ID="wrongQuestionsLbl" runat="server"
Text="30"></asp:Label>
                    </td>
                </tr>

                <tr>
                    <th>
                        Αποτέλεσμα:
                    </th>
                    <td>
                        <h3><asp:Label ID="resultLbl" runat="server"
Text="Αποτυχία"></asp:Label></h3>
                        <br />
                        <asp:Label ID="wrongQuestionsLbl" runat="server"
Text="30"></asp:Label>
                    </td>
                </tr>
            </table>

            <div class="footer">

```

```
<p>Πτυχιακή εργασία Μαίρης Δελητζάκη</p>
</div>

</div>
</form>
</body>
</html>
```

## Το αρχείο Result.aspx.cs και ο έλεγχος

```
public partial class _Default : System.Web.UI.Page
{
    String username;
    Object[,] questionsArray;
    int questionNo, minutes, seconds;

    int correctAnswers = 0;
    String wrongAnswers = "";

    protected void Page_Load(object sender, EventArgs e)
    {
        username = Session["userName"] + " " +
Session["userSurname"];
        questionsArray = (Object[,])Session["questionArray"];
        questionNo = Convert.ToInt32(Session["currentQuestion"]);
        minutes = Convert.ToInt32(Session["minutes"]);
        seconds = Convert.ToInt32(Session["seconds"]);

        String secondsString = "";
        if (seconds < 10)
            secondsString = "0" + seconds;
        else
            secondsString = "" + seconds;

        nameLbl.Text = username;
        timeLbl.Text = minutes + ":" + secondsString;

        checkAnswers();
    }

    protected void checkAnswers()
    {
        Configuration rootWebConfig =
WebConfigurationManager.OpenWebConfiguration("/Driving");
       ConnectionStringSettings connectionString =
rootWebConfig.ConnectionStrings.ConnectionStrings["DrivingDB"];
        OleDbConnection connection = new
OleDbConnection(connectionString.ToString());
        OleDbCommand command;
        connection.Open();

        for (int i = 0; i < questionsArray.Length / 2; i++)
```

```

        {
            int currentQuestionId = Convert.ToInt32(questionsArray[i,
0]);
            int currentAnswerId = Convert.ToInt32(questionsArray[i,
1]);
            if (currentAnswerId == -1)
                continue;

            command = new OleDbCommand("SELECT id, apantisi FROM
epiloges WHERE erotisi_id=" + currentQuestionId + " AND susti=true",
connection);
            OleDbDataReader myReader = command.ExecuteReader();
            myReader.Read();

            int correctAnswerId = Convert.ToInt32(myReader["id"]);

            if (currentAnswerId == correctAnswerId)
            {
                correctAnswers++;
            }
            else
            {
                String correctAnswer =
Convert.ToString(myReader["apantisi"]);
                myReader.Close();

                command = new OleDbCommand("SELECT ervtish FROM
erwtiseis WHERE id=" + currentQuestionId, connection);
                myReader = command.ExecuteReader();
                myReader.Read();

                String currentQuestion =
Convert.ToString(myReader["ervtish"]);

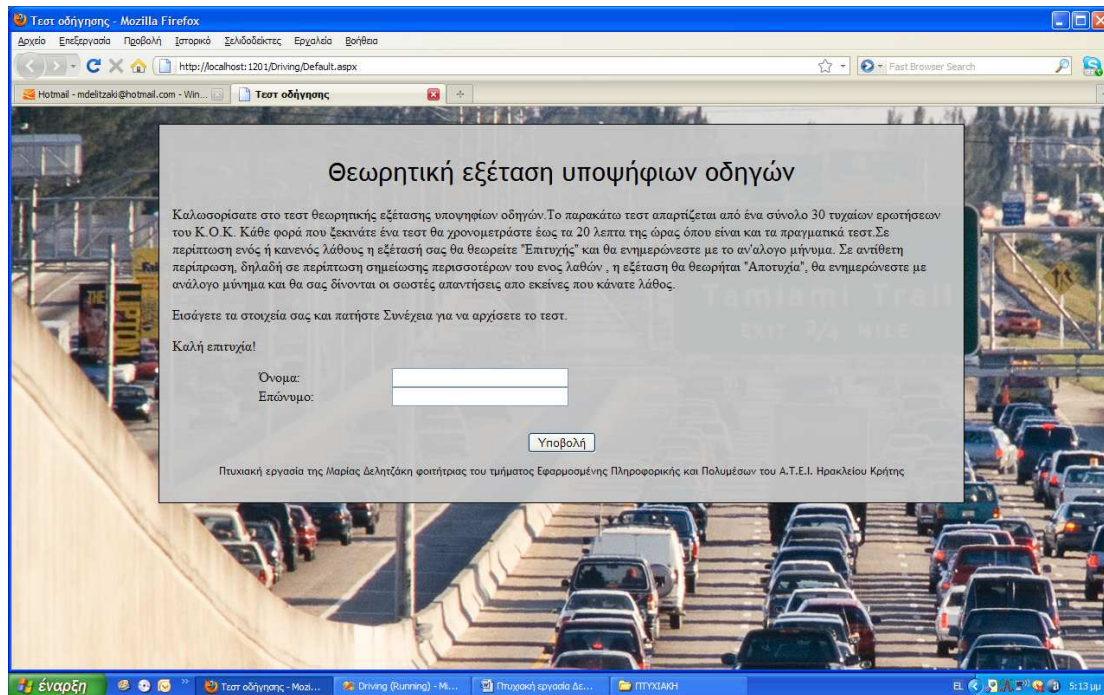
                wrongAnswers += "<p><em>" + currentQuestion +
"</em><br />" + correctAnswer + "</p>";
            }

            myReader.Close();
        }
        connection.Close();

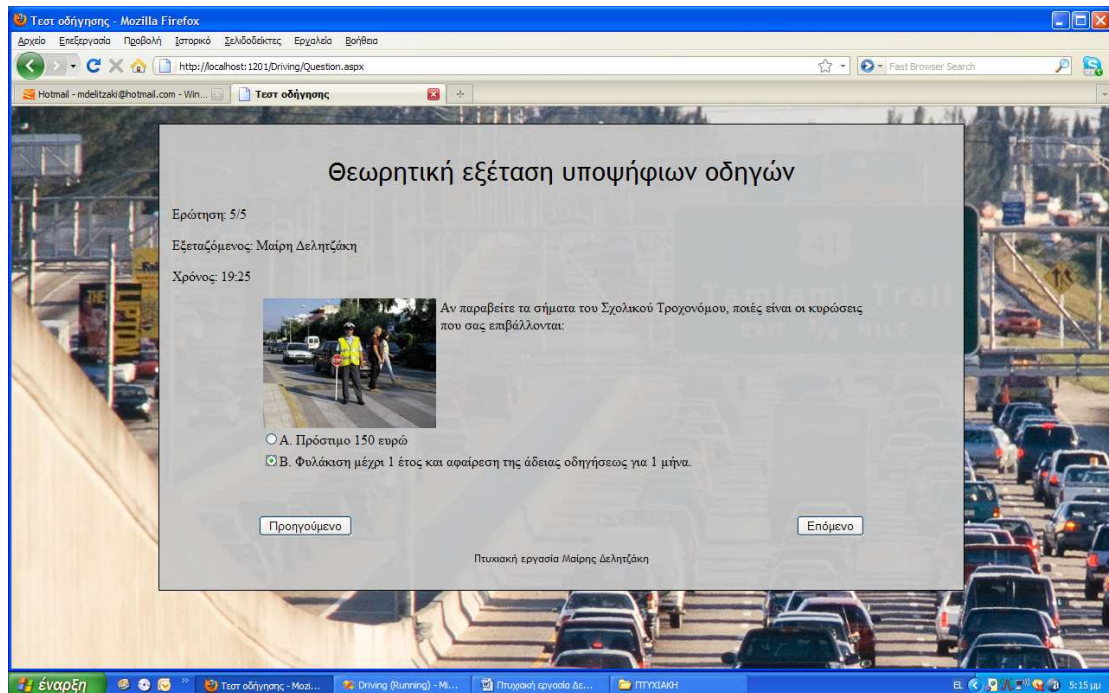
        if ((questionsArray.Length/2)-correctAnswers <= 1)
        {
            resultLbl.Text = "Επιτυχία";
        }
        else
        {
            resultLbl.Text = "Αποτυχία";
        }
        wrongQuestionsLbl.Text = wrongAnswers

```

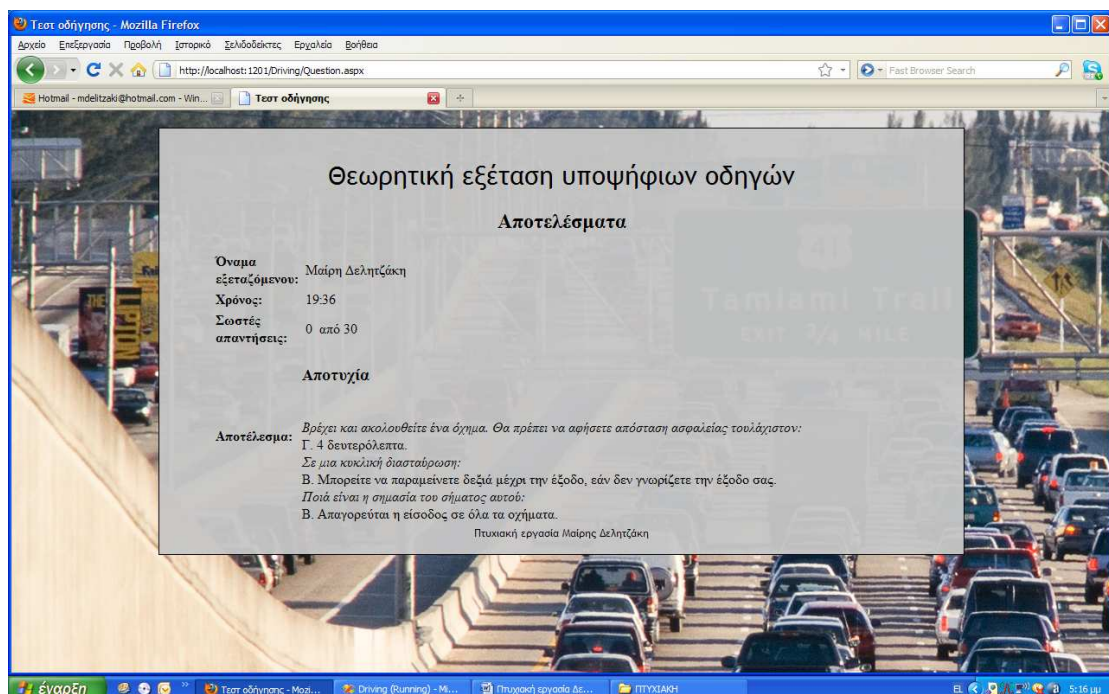
## 6.2 Η εφαρμογή στο ιντερνέτ Η αρχική σελίδα της εφαρμογής



**Οι επόμενες σελίδες που φιλοξενούν τις ερωτήσεις του τεστ εμφανίζονται με αυτή την μορφή**



Στην τελευταία σελίδα εμφανίζονται όλα όσα αναπτύξαμε πιο πάνω στην ανάλυση της εφαρμογής με αυτή την μορφή.



### 6.3 Η λειτουργία

Αυτή η διαδικτυακή εφαρμογή έχει υλοποιηθεί για να εξυπηρετεί κάθε χρήστη του ιντερνετ ο οποίος επιθυμεί να εκπαιδευτεί στο θεωρητικό μέρος του Κ.Ο.Κ. Η εφαρμογή αυτή βοηθάει κάθε υποψήφιο οδηγό εργαζόμενο ή όχι που επιθυμεί να εκπαιδευτεί στα τεστ του κ.ο.κ. οποιαδήποτε ώρα της ημέρας. Τα τεστ είναι απολύτως αξιόπιστα μιας και οι ερωτήσεις είναι αυτούσιες με αυτές που παρέχει το Υπουργείο Μεταφορών και Επικοινωνιών.

Σε αυτή την ιστοσελίδα κάθε επισκέπτης μπορεί να κάνει μια σειρά από τεστ των 30 ερωτήσεων.

Οι 30 ερωτήσεις του κ.ο.κ. που απαρτίζουν κάθε τεστ επιλέγονται τυχαία κάθε φορά που ο υποψήφιος επιλέγει να πραγματοποιήσει ένα νέο τεστ. Για κάθε ερώτηση υπάρχει και μια εικόνα στην οποία αναφέρεται, όπως και ένα πλήθος δύο ή τριών επιλογών από όπου ο χρήστης καλείται να επιλέξει την σωστή απάντηση.

Η εφαρμογή έχει σχεδιαστεί έτσι ώστε να καλύψει όλους τους πιθανούς συνδυασμούς ερωτήσεων που μπορεί να υπάρχουν σε ένα τεστ του κ.ο.κ. Τα τεστ διαρκούν 20 λεπτά της ώρας και για να θεωρηθεί σαν "επιτυχής" πρέπει να γίνουν λιγότερα από ένα λάθη.

Σε διαφορετική περίπτωση δηλαδή σταν ο χρήστης σημειώσει πάνω από ένα λάθη, ή ξεπεράσει τα 20 λεπτά της ώρας, ενημερώνεται με μήνυμα αποτυχίας και μπορεί να δει τις σωστές απαντήσεις των λαθνασμένων επιλογών του.

## 7. Κεφάλαιο – Επίλογος

Ο σκοπός για τον οποίο υλοποιήθηκε η παραπάνω διαδικτυακή εφαρμογή είναι εκπαιδευτικός! Αποτελεί έναν απλό, γρήγορο και αξιόπιστο τρόπο για να μπορέσει κάθε χρήστης να καλύψει όλες τις ερωτήσεις του Κ.Ο.Κ. χωρίς να υποβάλλεται στην κουραστική και βαρετή πολλές φορές διαδικασία της επανάληψης προκαθορισμένων τεστ. Με την εναλλακτική αυτή λύση πετυχαίνουμε να έχουμε κάθε φορά ένα νέο τεστ με διαφορετικές ερωτήσεις που κινούν το ενδιαφέρον του χρήστη και που εξαλείφουν την τάση να άπαντα "μηχανικά" σε τεστ προκαθορισμένα. Ακόμη πετυχαίνουμε και την εξομοίωση με τα πραγματικά τεστ μιας και ο χρήστης διαγωνίζεται σε πραγματικό που χρόνο που το κάνει πιο ρεαλιστικό. Τέλος η δυναμικότητα αυτής της web εφαρμογής ανάγεται στην τυχαία επιλογή ερωτήσεων μέσα από ένα σύνολο, στην αλληλεπίδραση της

εφαρμογής με τις απαντήσεις που σημειώνει ο κάθε χρήστης και από την δυνατότητα το αποτελέσματος ανάλογα με την τήρηση του χρόνου και το πλήθος των σωστών απαντήσεων.

### 8. Κεφάλαιο– Βιβλιογραφία Πηγές

- ASP.NET , Ryan, Danny , John Wiley and Sons Ltd 2001  
Programming ASP.NET, Liberty, Jesse, Hurwitz, Dan , O'Reilly 2003
- Asp.Net 2.0 Revealed, Lorenz, Patrick , Springer-Verlag Berlin and Heidelberg GmbH & Co. KG 2003
- Mastering ASP.NET with VB.Net , Jones, A.Russell , John Wiley and Sons Ltd 2002  
Building Secure Microsoft ASP.NET Applications , Microsoft Press , Microsoft Press,U.S. 2003
- ASP.NET Data Web Controls , Mitchell, Scott , Macmillan Computer Publishing (a Pearson Education company) 2003
- Professional ASP.NET 2.0 , Evjen, Bill, Hanselman, Scott, Muhammad, Farhan , John Wiley and Sons Ltd 2005
- Beginning ASP.Net 2.0 Databases , Kauffman, John, Thangarathinam, Thiru , John Wiley and Sons Ltd , 2005
- Programming Microsoft ASP.NET 2.0 Applications , Esposito, D. , Microsoft Press,U.S. 2006
- Beginning ASP.NET 1.1 E-Commerce , Darie and Watson , Apress , 2004
- ASP.NET E-Commerce Programming : Problem – Design – Solution , Hoffman , Wrox , 2003



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

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

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

Πλήρες Εγχειρίδιο της Visual Basic .NET , Πετρούτσος , Μ. Γκιούρδας , 2005

### **Ιστοσελίδες:**

[www.asp.net](http://www.asp.net)

[www.msdn.microsoft.com](http://www.msdn.microsoft.com)

[Access 2007 Tutorial « Annxyd's Blog»](#)





