



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Διαδραστική Εφαρμογή Ιστού Εκπαιδευτικού Περιεχομένου Interactive Web Application of Educational Content
Όνοματεπώνυμο Φοιτητή	Ευάγγελος Μαυρόπουλος
Πατρώνυμο	Κωνσταντίνος
Αριθμός Μητρώου	ΜΠΠΛ 16012
Επιβλέπων	Ευθύμιος Αλέπης, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Ιανουάριος 2019**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

(υπογραφή)

Μαρία Βίρβου
Καθηγήτρια

(υπογραφή)

Κων/νος Πατσάκης
Επίκουρος Καθηγητής

ΠΕΡΙΛΗΨΗ

Στη παρούσα μεταπτυχιακή διατριβή αναπτύχθηκε εφαρμογή ιστού προηγμένου λογισμικού, διαδραστικού εκπαιδευτικού περιεχομένου. Αποτελείτε από ενότητες θεωρίας και διαγωνισμάτων και αναφέρεται κατά κύριο λόγο σε μαθητές γυμνάσιου, λυκείου ή και σε μεγαλύτερες ηλικίες που θέλουν να εμπλουτίσουν τις γνώσεις στα μαθηματικά και συγκεκριμένα στην στατιστική και στις πιθανότητες. Ο κάθε χρήστης αφού κάνει εγγραφή έχει τη δυνατότητα να κάνει διαγωνίσματα από πέντε διαφορετικά επίπεδα και να δει τη βαθμολογία του. Επίσης υπάρχει η δυνατότητα να βλέπει το ιστορικό όλων των βαθμολογιών του ανά επίπεδο διαγωνισμάτων σε μορφή γραφημάτων με βάση βαθμολογία και ημερομηνία. Επιπρόσθετα σε περίπτωση που ολοκληρώσει όλα τα διαγωνίσματα από τα διαφορετικά επίπεδα με βαθμολογία πάνω από 50% εμφανίζεται το όνομα του χρήστη και ημερομηνία ολοκλήρωσης της εφαρμογής. Αρχικά, παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής αυτής καθώς και η αρχιτεκτονική σχεδίαση. Επίσης και ο τρόπος με βάση αυτές τις τεχνολογίες που αναπτύχθηκε η εφαρμογή. Στη συνέχεια τα χαρακτηριστικά λειτουργίας, εγχειρίδιο χρήστη με επεξηγηματικές εικόνες σε κάθε ενότητα της εφαρμογής. Τέλος έχουμε μια συνοπτική αναφορά στο εκπαιδευτικό περιεχόμενο της εφαρμογής

ABSTRACT

In this postgraduate dissertation, a web-based application of advanced software has been developed for interactive educational content. It consists of theory modules and exams, and refers primarily to high school students, or later in life that they want to enrich their knowledge in mathematics and in particular statistical and probability. Each user has the ability to make competitions from five different levels and see his score. He also has the ability to view the history of all his grades by grade level in graphs based on scoring and date. Additionally, if all competitions are completed from different levels with a score of more than 50%, the user name and date of completion of the application are displayed. Initially, the technologies used to implement this application and the architectural design are presented. Also, the way based on these technologies developed the application. Then the operating characteristics, how it works, a user manual with explanatory pictures in each section of the application. Finally, we have a brief reference to the educational content of the application

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ.....	iii
ABSTRACT	iv
Κεφάλαιο 1 Εισαγωγή.....	1
1.1 Στόχος της διατριβής	1
Κεφάλαιο 2 Τεχνολογίες και Αρχιτεκτονική Σχεδίαση	2
2.1 Περιβάλλον Ανάπτυξης	2
2.2 Εισαγωγή ASP.NET.....	3
2.3 Γλώσσα προγραμματισμού C#.....	3
2.4 Σύνταξη Razor.....	4
2.5 Μοντέλο υλοποίησης MVC (Model – View – Controller)	4
2.5.1 Βασικά πλεονεκτήματα	5
2.6 Δομή Εφαρμογής (Project Structure).....	6
2.6.1 MVC Controller	7
2.6.2 MVC Model	8
2.6.3 MVC View	9
2.6.4 MVC Δρομολόγηση(Routing).....	11
2.7 Τεχνολογίες διεπαφής χρήστη (front-end).....	11
2.7.1 HTML	11
2.7.2 CSS	16
2.7.3 JavaScript	18
2.7.4 jQuery βιβλιοθήκη	21
2.8 Σχεδιασμός και υλοποίηση Βάσης Δεδομένων	22
2.8.1 Βάση Δεδομένων Microsoft SQL Server	24
2.8.2 Πρόσβαση και διαχείριση δεδομένων Entity Framework	25
2.8.3 Αποθήκευση και ανάκτηση δεδομένων LINQ	26
2.8.4 Εμφάνιση δεδομένων σε γραφική μορφή	29
Κεφάλαιο 3 Χαρακτηριστικά λειτουργίας εφαρμογής	30
3.1 Αρχική σελίδα.....	30
3.2 Εγγραφή χρήστη	31
3.3 Είσοδος χρήστη	32
3.4 Ενότητα Θεωρίας	33
3.5 Ενότητα Διαγωνισμάτων	35
3.5.1 Κεντρικό μενού.....	35
3.5.2 Διαγωνίσματα.....	36

3.6 Έλεγχος επιδόσεων χρήστη.....	37
3.7 Χρήστες που έχουν ολοκληρώσει όλα τα διαγωνίσματα επιτυχώς	38
Κεφάλαιο 4 Εκπαιδευτικό περιεχόμενο εφαρμογής	39
4.1 Βασικά στοιχεία Στατιστικής – Μέτρα Θέσης και Διασποράς	39
4.2 Εισαγωγή στη θεωρία πιθανοτήτων	40
4.3 Πράξεις με ενδεχόμενα	41
4.3.1 Συμπληρωματικό ενδεχόμενο A'	41
4.3.2 A ένωση B , $A \cup B$	42
4.3.3 A τομή B , $A \cap B$	42
4.4 Ασυμβίβαστα ή Ξένα ενδεχόμενα μεταξύ τους	42
ΒΙΒΛΙΟΓΡΑΦΙΑ	43

Κεφάλαιο 1 Εισαγωγή

1.1 Στόχος της διατριβής

Στη παρούσα μεταπτυχιακή διατριβή παρουσιάζεται η υλοποίηση διαδραστικής εφαρμογής εκπαιδευτικού περιεχομένου. Τα τελευταία χρόνια η ταχεία ανάπτυξη της τεχνολογίας είναι συνυφασμένη με όλες τις πτυχές της καθημερινότητας μας. Συνεπώς γίνεται κατανοητό ότι αυτό έχει αποτέλεσμα και σχεδόν σε όλους τους επαγγελματικούς κλάδους και στον εκπαιδευτικό χώρο. Πλέον η διδασκαλία των μαθητών αλλά γενικότερα ο τρόπος μετάδοσης έχει αλλάξει από το παραδοσιακό τρόπο σε ένα τελείως διαφορετικό μοντέλο ανάπτυξης με τη χρήση των υπολογιστών. Η εφαρμογή αυτή είναι ένα πολύ χρήσιμο εργαλείο στο τομέα της εκπαίδευσης και οι βασικοί στόχοι δημιουργίας είναι:

- Ανάπτυξη πλατφόρμας με σύγχρονες τεχνολογίες που εξασφαλίζουν ασφάλεια καταχώρησης δεδομένων και ανάκτησης για εκπαιδευτικούς σκοπούς
- Ενότητα θεωρίας, που εισάγει βασικές έννοιες στο αντικείμενο Στατιστικής και Θεωρίας Πιθανοτήτων
- Διαγωνίσματα πολλαπλών επιπέδων βαθμού δυσκολίας
- Καταγραφή όλων των αποτελεσμάτων στα διαγωνίσματα με βάση την ημερομηνία, που σε αυτά έχει πρόσβαση και ο χρήστης έτσι ώστε να ελέγχει την επίδοση του ανά ημερομηνία και να διατηρείτε ιστορικό
- Χρήση υπολογιστή μέτρων θέσης και διασποράς, όπου ο χρήστης μπορεί να εισάγει τυχαίες τιμές και να λάβει δυναμικά τα αποτελέσματα. Είναι αρκετά χρήσιμο για εξάσκηση καθώς και για επαλήθευση αποτελεσμάτων
- Κριτήρια ολοκλήρωσης της εφαρμογής με βάση της βαθμολογίες. Ο κάθε χρήστης θεωρείτε πιστοποιημένος εφόσον έχει ολοκληρώσει επιτυχώς όλες τα επίπεδα δυσκολίας των διαγωνισμάτων
- Γραφική απεικόνιση των αποτελεσμάτων όλων των χρηστών για την εξαγωγή ποιοτικών συμπερασμάτων με βάση το κάθε χρήστη αλλά και συνολική
- Συμμετοχή όσο το δυνατών περισσότερων χρηστών για την αξιολόγηση της εφαρμογής

Συμπερασματικά η εφαρμογή αυτή περιλαμβάνει τα κυριότερα βήματα μάθησης που είναι μελέτη, αξιολόγηση, επίπεδο δυσκολίας, ολοκλήρωση εκμάθησης. Επιπλέον η εφαρμογή μπορεί εύκολα να τροποποιηθεί σε οπουδήποτε αντικείμενο μάθησης αλλά και να εξελιχθεί

Κεφάλαιο 2 Τεχνολογίες και Αρχιτεκτονική Σχεδίαση

2.1 Περιβάλλον Ανάπτυξης

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε το Microsoft Visual Studio 2015 Enterprise. Το Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft. Χρησιμοποιείται για την ανάπτυξη προγραμμάτων ηλεκτρονικών υπολογιστών, καθώς και ιστοσελίδων, εφαρμογών ιστού, υπηρεσιών ιστού και εφαρμογών για κινητά. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft όπως τα Windows API, τα Windows Forms, το Windows Presentation Foundation, το Windows Store και το Microsoft Silverlight. Μπορεί να παράγει τόσο εγγενή κώδικα όσο και διαχειριζόμενο κώδικα.

Το Visual Studio περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα το οποίο υποστηρίζει το IntelliSense (το στοιχείο ολοκλήρωσης κώδικα) καθώς και το refactoring κώδικα. Το ολοκληρωμένο πρόγραμμα εντοπισμού σφαλμάτων λειτουργεί τόσο ως πρόγραμμα εντοπισμού σφαλμάτων σε επίπεδο πηγής όσο και ως εργαλείο εντοπισμού σφαλμάτων σε επίπεδο μηχανής. Άλλα ενσωματωμένα εργαλεία περιλαμβάνουν έναν προφίλ κώδικα, σχεδιαστή μορφών για την κατασκευή εφαρμογών GUI, σχεδιαστή ιστοσελίδων, σχεδιαστή τάξεων και σχεδιαστή σχήματος βάσης δεδομένων. Αποδέχεται plug-ins που βελτιώνουν τη λειτουργικότητα σχεδόν σε όλα τα επίπεδα - συμπεριλαμβανομένης της προσθήκης υποστήριξης για συστήματα ελέγχου πηγής (όπως το Subversion και το Git) και την προσθήκη νέων εργαλείων όπως οι εκδότες και οι οπτικοί σχεδιαστές για συγκεκριμένες γλώσσες ή ομάδες εργαλείων για άλλες πτυχές της ανάπτυξης λογισμικού (όπως ο πελάτης Team Foundation Server: Team Explorer).

Το Visual Studio υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού και επιτρέπει στον επεξεργαστή κειμένου και στον εντοπισμό σφαλμάτων να υποστηρίξει (σε διαφορετικούς βαθμούς) σχεδόν οποιαδήποτε γλώσσα προγραμματισμού, υπό την προϋπόθεση ότι υπάρχει μια συγκεκριμένη γλώσσα. Οι ενσωματωμένες γλώσσες περιλαμβάνουν C, C ++, C ++ / CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML και CSS. Υποστήριξη για άλλες γλώσσες όπως Python, Ruby, Node.js και μεταξύ άλλων είναι διαθέσιμη μέσω plug-ins. Τα Java (και J #) υποστηρίχθηκαν στο παρελθόν.

2.2 Εισαγωγή ASP.NET

Το ASP.NET είναι ένα Web framework που σχεδιάστηκε και αναπτύχθηκε από τη Microsoft. Χρησιμοποιείται για την ανάπτυξη ιστοτόπων, εφαρμογών ιστού και υπηρεσιών διαδικτύου. Παρέχει ενσωμάτωση HTML, CSS και JavaScript. Ξεκίνησε για πρώτη φορά τον Ιανουάριο του 2002. Είναι χτισμένο στο Common Language Runtime (CLR) και επιτρέπει στους προγραμματιστές να γράφουν κώδικα χρησιμοποιώντας οποιαδήποτε υποστηριζόμενη γλώσσα .NET

Το ASP σημαίνει Active Server Pages και υποστηρίζει τα εξής μοντέλα ανάπτυξης

- Classic ASP
- ASP.NET Web Forms
- ASP.NET MVC
- ASP.NET Web Pages
- ASP.NET API
- ASP.NET Core

Στη παρούσα διπλωματική διατριβή έχει επιλεγθεί το μοντέλο ανάπτυξης ASP.NET MVC

2.3 Γλώσσα προγραμματισμού C#

Η C# είναι κατά κύριο λόγο αντικειμενοστραφής γλώσσα προγραμματισμού, ασφαλής (type-safe) και αποτελεί μέρος του .NET framework. Οι ρίζες της κατά βάση είναι στη C και παρουσιάζει αρκετές ομοιότητες και με τη Java. Αναπτύχθηκε από τη Microsoft κατά το 2000 και θεωρείται ως μία από τις πιο σε ζήτηση γλώσσες προγραμματισμού. Από τη πρώτη έκδοσή της το 2000 έχει εξελιχθεί αρκετά και έχει ενσωματώσει επιπλέον χαρακτηριστικά.

Το C # έχει ένα ενοποιημένο σύστημα τύπου. Όλοι οι τύποι C #, συμπεριλαμβανομένων των τύπων όπως για παράδειγμα int, float, double, κληρονομούν από το αντικείμενο. Έτσι, όλοι οι τύποι μοιράζονται ένα σύνολο κοινών λειτουργιών και ιδιότητες οποιουδήποτε τύπου μπορούν να αποθηκευτούν, να μεταφερθούν και να λειτουργήσουν με συνεπή τρόπο. Επιπλέον, η C # υποστηρίζει τόσο τους τύπους αναφοράς που έχουν οριστεί από τον χρήστη (by ref) όσο και τους τύπους τιμών (by value), επιτρέποντας τη δυναμική κατανομή αντικειμένων. Για να διασφαλιστεί ότι τα προγράμματα και οι βιβλιοθήκες C # μπορούν να εξελιχθούν με την πάροδο του χρόνου χωρίς προβλήματα, δόθηκε μεγάλη έμφαση στις εκδόσεις της C #. Πολλές γλώσσες προγραμματισμού δεν δίνουν μεγάλη προσοχή σε αυτό το ζήτημα και, ως εκ τούτου, τα προγράμματα γραμμένα σε αυτές τις γλώσσες σπάνε πιο συχνά από ό, τι είναι απαραίτητο όταν εισάγονται νεότερες εκδόσεις από βιβλιοθήκες που εξαρτώνται από προηγούμενες εκδόσεις βιβλιοθηκών. Ο σχεδιασμός των εκδόσεων της C# βασίστηκε πολύ σε αυτό το κομμάτι.

Με τη βοήθεια της C#, μπορούμε να αναπτύξουμε διαφορετικούς τύπους ασφαλών εφαρμογών

- Εφαρμογές παραθύρων (windows application)
- Εφαρμογές ιστού (windows application)
- Υπηρεσίες διαδικτύου (web services)
- Εφαρμογές βάσεων δεδομένων (Database application)

2.4 Σύνταξη Razor

Το MVC χρησιμοποιεί μηχανή προβολής Razor, ώστε να μπορούμε να γράψουμε επίσης κώδικα πλευράς διακομιστή (Server) σε HTML. Το Razor είναι μια σύνταξη που επιτρέπει να ενσωματωθεί κώδικας βασισμένο σε διακομιστές (Visual Basic και C #) σε ιστοσελίδες.

Ο κώδικας που βασίζεται στον διακομιστή μπορεί να δημιουργήσει δυναμικό περιεχόμενο ιστού ενώ μια ιστοσελίδα διατίθεται στο πρόγραμμα περιήγησης. Όταν καλείται μια ιστοσελίδα, ο διακομιστής εκτελεί τον κώδικα που βασίζεται σε διακομιστή μέσα στη σελίδα πριν επιστρέψει τη σελίδα στο πρόγραμμα περιήγησης. Με την εκτέλεση του διακομιστή, ο κώδικας μπορεί να εκτελεί πολύπλοκες εργασίες, όπως πρόσβαση σε βάσεις δεδομένων.

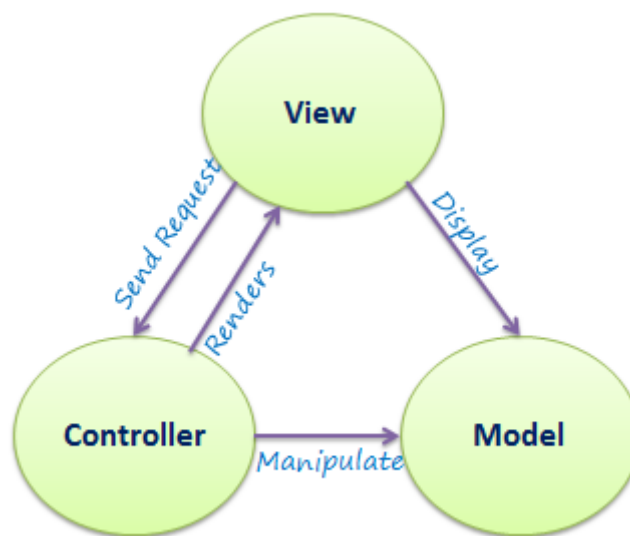
2.5 Μοντέλο υλοποίησης MVC (Model – View – Controller)

Σύμφωνα με τον επίσημο ορισμό, ο Model-View-Controller (MVC) είναι ένα αρχιτεκτονικό μοντέλο λογισμικού για την υλοποίηση διεπαφών χρήστη. Διαχωρίζει μια δεδομένη εφαρμογή λογισμικού σε τρία διασυνδεδεμένα μέρη, έτσι ώστε να διαχωρίζονται οι εσωτερικές αναπαραστάσεις πληροφοριών από τους τρόπους με τους οποίους οι πληροφορίες παρουσιάζονται ή γίνονται αποδεκτές από τον χρήστη. Το MVC είναι ένα αρχιτεκτονικό μοτίβο που χωρίζει μια εφαρμογή σε τρία βασικά λογικά στοιχεία: το μοντέλο, την προβολή και τον ελεγκτή. Κάθε ένα από αυτά τα στοιχεία είναι κατασκευασμένα για να χειρίζονται συγκεκριμένες πτυχές ανάπτυξης μιας εφαρμογής. Το MVC είναι ένα από τα πιο συχνά και σύγχρονα χρησιμοποιούμενα πρότυπα ανάπτυξης ιστοσελίδων

- Model
Το στοιχείο Μοντέλο αντιστοιχεί σε όλες τις λογικές που σχετίζονται με τα δεδομένα με τα οποία συνεργάζεται ο χρήστης. Αυτό μπορεί να αντιπροσωπεύει είτε τα δεδομένα που μεταφέρονται μεταξύ των στοιχείων προβολής και ελεγκτή είτε οποιαδήποτε άλλα δεδομένα που σχετίζονται με επιχειρηματική λογική. Για παράδειγμα, ένα αντικείμενο πελάτη θα ανακτήσει τις πληροφορίες του πελάτη από τη βάση δεδομένων, θα το χειριστεί και θα το ενημερώσει ξανά στη βάση δεδομένων ή θα το χρησιμοποιήσει για την απόδοση δεδομένων.
- View
Το στοιχείο Προβολή χρησιμοποιείται για όλη τη λογική του UI της εφαρμογής. Για παράδειγμα, η προβολή πελατών θα περιλαμβάνει όλα τα στοιχεία UI, όπως κουτιά κειμένου, αναπτυσσόμενα μενού. Με τα οποία αλληλοεπιδρά ο τελικός χρήστης.

- Controller

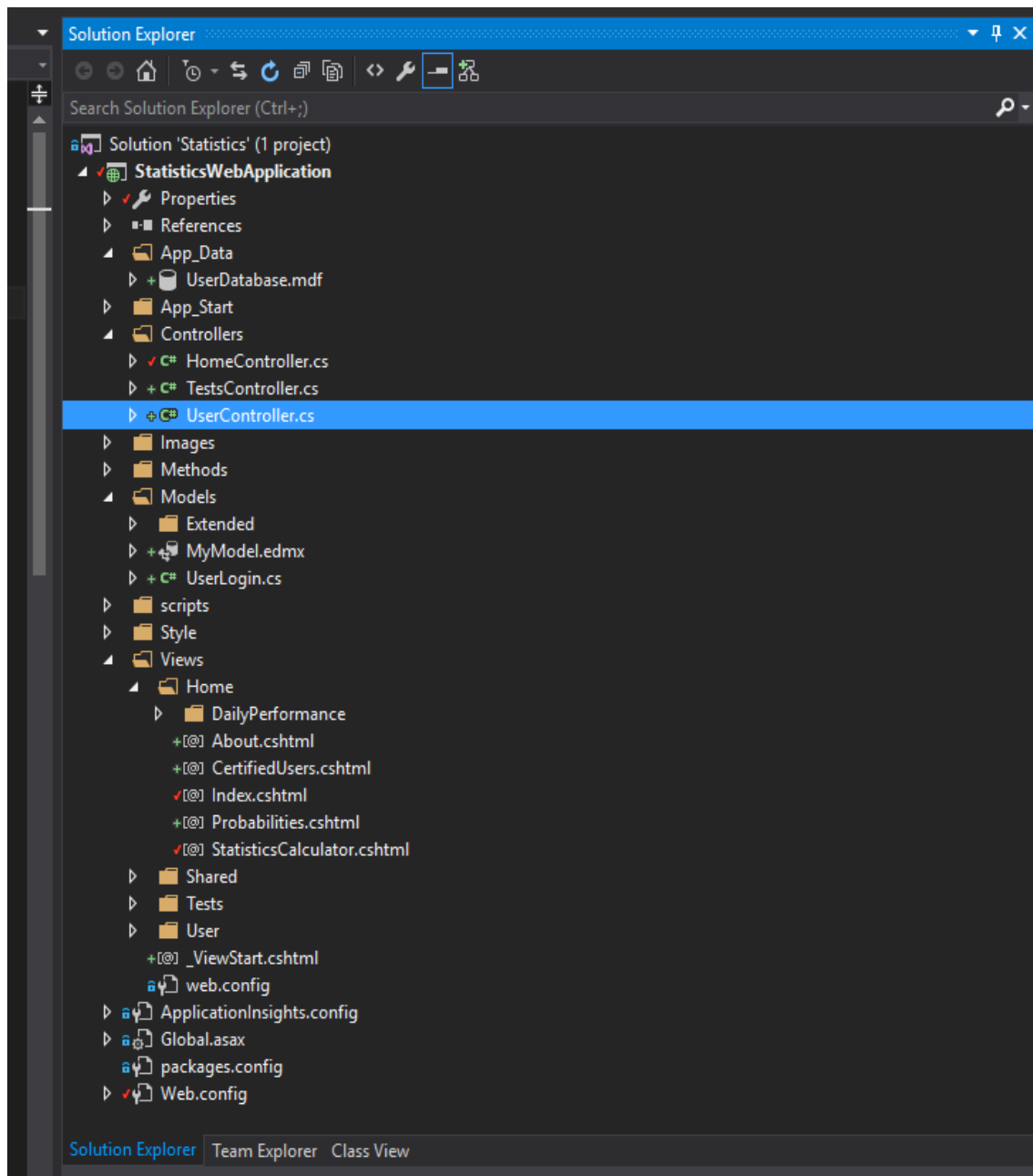
Οι ελεγκτές λειτουργούν ως διεπαφή μεταξύ των στοιχείων του μοντέλου και της προβολής για να επεξεργάζονται όλες τις επιχειρησιακές λογικές και εισερχόμενες αιτήσεις, να χειρίζονται δεδομένα χρησιμοποιώντας το στοιχείο μοντέλο και να αλληλοεπιδρούν με τις προβολές για να καταστήσουν την τελική έξοδο. Για παράδειγμα, ο ελεγκτής πελάτη θα χειριστεί όλες τις αλληλεπιδράσεις και τις εισροές από την προβολή πελάτη και θα ενημερώσει τη βάση δεδομένων χρησιμοποιώντας το μοντέλο πελάτη. Ο ίδιος ελεγκτής θα χρησιμοποιηθεί για την προβολή των δεδομένων του πελάτη



2.5.1 Βασικά πλεονεκτήματα

- Ο διαχωρισμός του προγράμματος. Αυτό σημαίνει ότι μπορούμε να χωρίσουμε σε τρία μέρη της εφαρμογής όπως το Μοντέλο, Προβολή και Ελεγκτής
- Χειρίζεται εύκολα τον κώδικα λόγω του διαχωρισμού
- Ταυτόχρονα, μπορούμε να χωρίσουμε τη δουλειά πολλών προγραμματιστών ταυτόχρονα. Δεν θα επηρεάσει το έργο ενός προγραμματιστή στο έργο ενός άλλου προγραμματιστή.
- Υποστηρίζει TTD (test-driven development). Δημιουργία σεναρίων για τον έλεγχο της εφαρμογής
- Η τελευταία έκδοση του MVC υποστηρίζει προεπιλεγμένες ανταποκρίσιμες ιστοσελίδες και κινητά πρότυπα.

2.6 Δομή Εφαρμογής (Project Structure)



Όπως βλέπουμε, περιέχει τρεις φακέλους που ονομάζονται Μοντέλο, Προβολή και Ελεγκτής. Το HomeController είναι προεπιλεγμένος ελεγκτής για την εφαρμογή. Ο φάκελος App_Data περιέχει τη βάση δεδομένων. Ο φάκελος Controllers ως διεπαφή μεταξύ των στοιχείων του μοντέλου και της προβολής. Φάκελος Models που αποτελεί ουσιαστικά σε κώδικα της λειτουργία της βάσης δεδομένων γνωστό και ως entity framework όπου θα αναλυθεί η λειτουργία του σε επόμενη ενότητα. Επίσης οι φάκελοι scripts που περιέχει κώδικα σε JavaScript και ο φάκελος Style με κώδικα css που και τα δυο αφορούν λειτουργία του UI της εφαρμογής. Τέλος ο φάκελος Views το στοιχείο Προβολή χρησιμοποιείται για όλη τη λογική του UI της εφαρμογής. Οπού με τη βοήθεια του controller είναι ουσιαστικά αυτό που εμφανίζεται στο χρήστη με τα οποία αλληλοεπιδρά ο τελικός χρήστης. Μέσα σε κάθε φάκελο μπορούμε να έχουμε και υπό φακέλους για να οργανώσουμε καλύτερα το κώδικα της εφαρμογής. Παράδειγμα στο φάκελο Views/Home εκτός από της σελίδες βλέπουμε υπάρχει και ο φάκελος DailyPerformance. όπου το περιεχόμενο τους αναλύεται στο επόμενο κεφάλαιο.

2.6.1 MVC Controller

Ο ελεγκτής είναι μια κλάση που χειρίζεται τα αιτήματα των χρηστών. Ανακτά δεδομένα από το μοντέλο και αποδίδει την προβολή ως απόκριση.

Το πλαίσιο ASV.NET MVC αντιστοιχεί σε ζητούμενες διευθύνσεις URL στις κλάσεις που αναφέρονται ως ελεγκτές. Ο ελεγκτής επεξεργάζεται τα εισερχόμενα αιτήματα, χειρίζεται τις εισροές και τις αλληλεπιδράσεις των χρηστών και εκτελεί αυτό που ζητάμε να επιστρέψει.

Παράδειγμα χρήσης controller

```
namespace Statistics.Controllers
{
    public class HomeController : Controller
    {
        // GET: Home
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult StatisticsCalculator()
        {
            return View();
        }

        public ActionResult Probabilities()
        {
            return View();
        }
    }
}
```

Για να αποκτήσετε πρόσβαση σε αυτόν τον ελεγκτή στο πρόγραμμα περιήγησης, προσθέτουμε ένα αρχείο μέσα στον φάκελο προβολής (View) με κατάληξη .cshtml. Το αρχείο αυτό μπορεί να περιέχει κώδικα html, css ,JavaScript ή C# με σύνταξη Razor

2.6.2 MVC Model

Τώρα, μπορούμε να προσθέσουμε οποιοδήποτε αριθμό ιδιοτήτων και μεθόδων στο μοντέλο. Αυτά είναι χρήσιμα για να καταστεί η MVC μια καθαρή προσέγγιση όπως περιεγράφηκε. Δημιουργούμε μερικές ιδιότητες και πώς επικοινωνούν αυτές με τη βάση δεδομένων όπου θα αναλυθεί παρακάτω η λογική σχεδίασης

Παράδειγμα κλάσης User και Level_1 όπου συνδέονται με μοναδικό τρόπο χρησιμοποιώντας το constructor User. Επίσης έχουμε τα instant fields κάθε κλάσης με τις ιδιότητες get-set για κάθε ένα αντικείμενο

```
public partial class User
{
    public User()
    {
        this.Level_1 = new HashSet<Level_1>();
        this.Level_2 = new HashSet<Level_2>();
        this.Level_3 = new HashSet<Level_3>();
        this.Level_4 = new HashSet<Level_4>();
        this.Level_5 = new HashSet<Level_5>();
    }

    public int UserId { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string Email { get; set; }
    public System.DateTime DateOfBirth { get; set; }
    public string Password { get; set; }
    public virtual ICollection<Level_1> Level_1 { get; set; }
    public virtual ICollection<Level_2> Level_2 { get; set; }
    public virtual ICollection<Level_3> Level_3 { get; set; }
    public virtual ICollection<Level_4> Level_4 { get; set; }
    public virtual ICollection<Level_5> Level_5 { get; set; }
}
}
```

```
public partial class Level_1
{
    public int Id { get; set; }
    public int UserId { get; set; }
    public int Score { get; set; }
    public bool Completed { get; set; }
    public System.DateTime Date { get; set; }

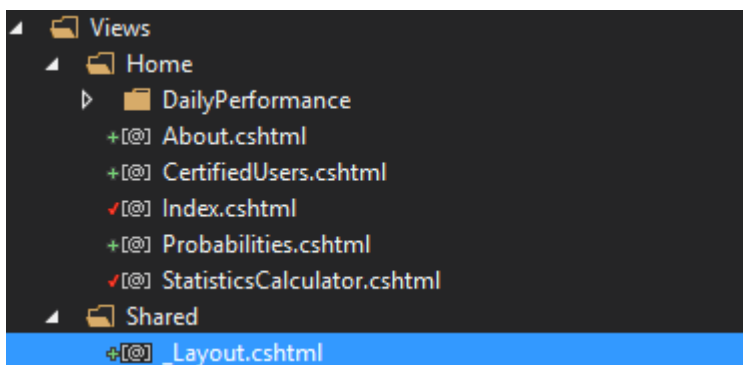
    public virtual User User { get; set; }
}
}
```

2.6.3 MVC View

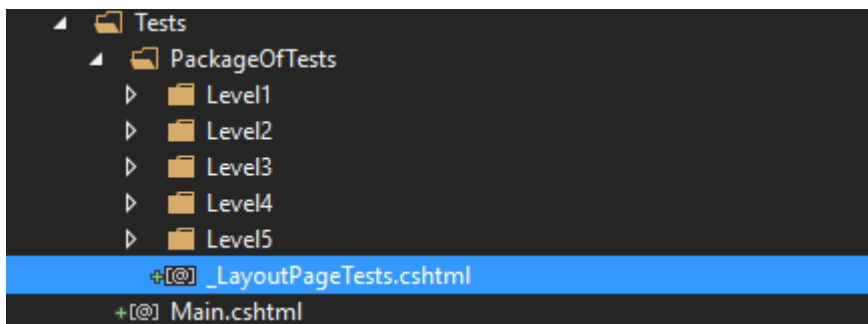
Η προβολή MVC είναι μια τυπική σελίδα HTML που μπορεί να περιέχει δέσμες ενεργειών. Χρησιμοποιείται για τη δημιουργία ιστοσελίδων για την εφαρμογή. Σε αντίθεση με τις ιστοσελίδες ASP.NET, οι προβολές MVC αντιστοιχίζονται στη δράση και έπειτα ο ελεγκτής μετατρέπει την προβολή στο πρόγραμμα περιήγησης. Η MVC έχει ορισμένους περιορισμούς. Το αρχείο προβολής θα πρέπει να βρίσκεται στον υποκατάλογο του φακέλου Προβολή.

Μια εφαρμογή μπορεί να περιέχει κοινά τμήματα στο περιβάλλον χρήστη, τα οποία παραμένουν τα ίδια σε όλη την εφαρμογή, όπως το λογότυπο, η κεφαλίδα, η αριστερή γραμμή πλοήγησης, το δεξί τμήμα ή το υποσέλιδο. Το ASP.NET MVC εισήγαγε μια προβολή Layout που περιέχει αυτά τα κοινά τμήματα UI, έτσι ώστε να μην χρειάζεται να γράφουμε τον ίδιο κώδικα σε κάθε σελίδα. Η προβολή διάταξης είναι ίδια με την κύρια σελίδα της εφαρμογής ιστού ASP.NET

Η προβολή διάταξης έχει την ίδια επέκταση με άλλες προβολές, .cshtml . Οι προβολές διάταξης μοιράζονται με πολλαπλές προβολές, οπότε πρέπει να αποθηκεύονται στον κοινόχρηστο φάκελο. Για παράδειγμα, το _Layout.cshtml στο φάκελο κοινόχρηστο όπως φαίνεται παρακάτω.



Πρακτικά αυτό σημαίνει ότι δεν χρειάζεται να ξαναγράψουμε κώδικα σε σελίδες που έχουν κοινά τμήματα UI (User Interface). Επίσης μπορούμε να έχουμε διαφορετικά layout σε που να αναφέρονται σε διαφορετικές σελίδες. Στη συγκεκριμένη εφαρμογή επιλέχθηκε για τα Τεστ να υπάρχει διαφορετικό layout σε σχέση με τις υπόλοιπες σελίδες



Επιπλέον άλλο ένα σημαντικό κομμάτι όσο αναφορά τη λειτουργία της εφαρμογής είναι τα διαγωνίσματα όσο το δυνατό να είναι διαφορετικά κάθε φορά.

Όπως βλέπουμε στη παραπάνω εικόνα ο φάκελος PackageOfTests έχει 5 διαφορετικούς φακέλους Level1, Level2, Level3, Level4, Level5. Ο καθένας φάκελος περιέχει μέσα διαγωνίσματα για κάθε επίπεδο

Ας υποθέσουμε ότι ο χρήστης επιλέγει Level2, από αυτό το επίπεδο θα λάβει από το Controller ένα τυχαίο διαγώνισμα αυτού του επιπέδου. Ο λόγος προφανώς είναι να μην επαναλαμβάνονται όσο δυνατόν τα διαγωνίσματα

```
public ActionResult TestLevel2()
{
    Random rand = new Random();
    int level2 = rand.Next(1, 3);
    var userId = Session["UserId"].ToString();
    var completed = user.Level_1.Where(a =>
a.UserId.ToString() == userId && a.Completed ==
true).FirstOrDefault();
    if (completed == null)
    {
        TempData["msg"] = "<script>alert('Please first complete previous
levels');</script>";
        return RedirectToAction("Main", "Tests");
    }
    return View("~/Views/Tests/PackageOfTests/Level2/Test" + level2 +
".cshtml");
}
```

Ο συγκεκριμένος κώδικας ουσιαστικά ελέγχει αν ο χρήστης έχει ολοκληρώσει το προηγούμενο level έτσι ώστε να πάει στο επόμενο και ο Controller να μας φέρει ένα τυχαίο διαγώνισμα από το level2 το οποίο είναι τυχαίο μεν, αλλά έχει βαθμό δυσκολίας που αντιστοιχεί στο level2. Δηλαδή από το φάκελο Tests/PackageOfTests/Level2 μας επιστρέφει ένα από τα διαγωνίσματα.

Επίσης παρατηρούμε ότι μέσα στη κλάση του controller και μέσα στη μέθοδο μπορούμε να χρησιμοποιήσουμε JavaScript στο συγκεκριμένο παράδειγμα σε περίπτωση που ο χρήστης δεν έχει ολοκληρώσει το προηγούμενο level του εμφανίζεται ένα pop-up παράθυρο ενημερώνοντας τον ότι για να λάβει διαγωνίσματα από το level2 θα πρέπει πρώτα να ολοκληρώσει το προηγούμενο level

2.6.4 MVC Δρομολόγηση(Routing)

Στο MVC, η δρομολόγηση είναι μια διαδικασία χαρτογράφησης του αιτήματος του προγράμματος περιήγησης στη δράση του ελεγκτή και την επιστροφή απάντησης. Κάθε εφαρμογή MVC έχει προεπιλεγμένη δρομολόγηση για τον προεπιλεγμένο HomeController. Μπορούμε να ορίσουμε προσαρμοσμένη δρομολόγηση για νέο ελεγκτή.

Το αρχείο RouteConfig.cs χρησιμοποιείται για τον ορισμό της δρομολόγησης για την εφαρμογή.

```
namespace Statistics
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");

            routes.MapRoute(
                name: "Default",
                url: "{controller}/{action}/{id}",
                defaults: new { controller = "Home", action = "Index", id =
                    UrlParameter.Optional }
            );
        }
    }
}
```

2.7 Τεχνολογίες διεπαφής χρήστη (front-end)

Οι τεχνολογίες όσο αναφορά τη λειτουργία του UI (User Interface)

Είναι HTML5 Hypertext Markup Language, CSS3 Cascading Style Sheets και JavaScript

2.7.1 HTML

Στα τέλη της δεκαετίας του 1980, ο φυσικός Tim Berners-Lee, ο οποίος εργαζότανε στο CERN, πρότεινε ένα σύστημα εγγράφων για διαμοιρασμό και κοινή χρήση για τους ερευνητές του CERN. Το 1989, έγραψε ένα σημείωμα που πρότεινε ένα διαδικτυακό σύστημα υπερκειμένου.

Διαδραστική Εφαρμογή Ιστού Εκπαιδευτικού Περιεχομένου

Ο Tim Berners-Lee είναι γνωστός ως ο πατέρας του HTML. Η πρώτη διαθέσιμη περιγραφή της HTML ήταν ένα έγγραφο που ονομάζεται "Ετικέτες HTML" που προτάθηκε από τον Tim στα τέλη του 1991. Η τελευταία έκδοση του HTML είναι HTML5. Το HTML σημαίνει Hyper Text Markup Language, το οποίο χρησιμοποιείται για τη δημιουργία ιστοσελίδων και εφαρμογών ιστού. Γλώσσα Σήμανσης Υπερκειμένου είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

- Hyper Text: Το HyperText σημαίνει απλώς "Κείμενο μέσα στο κείμενο". Ένα κείμενο έχει έναν σύνδεσμο μέσα του, είναι ένα υπερκείμενο. Κάθε φορά που κάνετε κλικ σε έναν σύνδεσμο που σας μεταφέρει σε μια νέα ιστοσελίδα, έχετε κάνει κλικ σε ένα υπερκείμενο. Το HyperText είναι ένας τρόπος για να συνδέσετε δύο ή περισσότερες ιστοσελίδες (έγγραφα HTML) μεταξύ τους.
- Markup Language: Μια γλώσσα σήμανσης είναι μια γλώσσα υπολογιστή που χρησιμοποιείται για την εφαρμογή συμβάσεων διαμόρφωσης και μορφοποίησης σε ένα έγγραφο κειμένου. Η γλώσσα σήμανσης καθιστά το κείμενο πιο διαδραστικό και δυναμικό. Μπορεί να μετατρέψει το κείμενο σε εικόνες, πίνακες, συνδέσμους κτλ
- Ιστοσελίδα: Μια ιστοσελίδα είναι ένα έγγραφο το οποίο είναι συνήθως γραμμένο σε HTML και μεταφράζεται από ένα πρόγραμμα περιήγησης ιστού. Μια ιστοσελίδα μπορεί να αναγνωριστεί εισάγοντας μια διεύθυνση URL. Μια ιστοσελίδα μπορεί να είναι στατικού ή δυναμικού τύπου. Με τη βοήθεια μόνο της HTML, μπορούμε να δημιουργήσουμε στατικές ιστοσελίδες.

Ως εκ τούτου, η HTML είναι μια γλώσσα σήμανσης που χρησιμοποιείται για τη δημιουργία ελκυστικών ιστοσελίδων με τη βοήθεια της μορφοποίησης και η οποία φαίνεται σε μια ωραία μορφή σε ένα Web browser. Ένα έγγραφο HTML αποτελείται από πολλές ετικέτες HTML και κάθε ετικέτα HTML περιέχει διαφορετικό περιεχόμενο.

Χαρακτηριστικά της HTML

- Είναι μια πολύ εύκολη και απλή γλώσσα. Μπορεί να κατανοηθεί και να τροποποιηθεί εύκολα.
- Είναι πολύ εύκολο να κάνετε μια αποτελεσματική παρουσίαση με HTML επειδή έχει πολλές ετικέτες μορφοποίησης.
- Είναι μια γλώσσα σήμανσης, οπότε παρέχει έναν ευέλικτο τρόπο σχεδιασμού ιστοσελίδων μαζί με το κείμενο.
- Διευκολύνει τους προγραμματιστές να προσθέσουν έναν σύνδεσμο στις ιστοσελίδες.
- Είναι ανεξάρτητο από πλατφόρμα επειδή μπορεί να εμφανίζεται σε οποιαδήποτε πλατφόρμα, όπως Windows, Linux και Macintosh
- Διευκολύνει τον προγραμματιστή να προσθέσει γραφικά, βίντεο και ήχο στις ιστοσελίδες που το κάνει πιο ελκυστικό και διαδραστικό.
- Η HTML μπορούμε να χρησιμοποιήσουμε ετικέτες είτε με κεφαλαία είτε με πεζά γράμματα.

Τα έγγραφα HTML πρέπει να αρχίζουν με μια δήλωση τύπου εγγράφου (Document Type Declaration). Αυτή η δήλωση βοηθά τους browser να καταλάβουν πώς πρέπει να διαβάσουν το περιεχόμενο του εγγράφου και πώς να το παρουσιάσουν μετά.

Οι περισσότερες ιδιότητες των στοιχείων είναι ζεύγη ονομάτων και τιμών, τα οποία διαχωρίζονται με ένα «=» και γράφονται μέσα στην ετικέτα εκκίνησης ενός στοιχείου, μετά το

όνομα του στοιχείου. Η τιμή μπορεί να περικλείεται σε μονά ή διπλά εισαγωγικά, παρότι τιμές που αποτελούνται από συγκεκριμένους χαρακτήρες μπορούν να γράφονται χωρίς εισαγωγικά στην HTML, αλλά όχι στην XHTML. Το να μένουν οι τιμές των ιδιοτήτων χωρίς εισαγωγικά θεωρείται ανασφαλές. Εκτός από τις ιδιότητες που γράφονται ως ζεύγη ονομάτων και τιμών, υπάρχουν και μερικές οι οποίες επηρεάζουν το στοιχείο απλά με την παρουσία τους μέσα στην ετικέτα εκκίνησης. Υπάρχουν και μερικές κοινές ιδιότητες οι οποίες εμφανίζονται σε πολλά στοιχεία:

- Η ιδιότητα `id` παρέχει ένα αναγνωριστικό για ένα στοιχείο το οποίο είναι μοναδικό σε ολόκληρο το έγγραφο. Χρησιμοποιείται για να ταυτοποιεί το στοιχείο ώστε τα CSS να μπορούν να αλλάξουν τον τρόπο που αυτό εμφανίζεται, καθώς και τα σενάρια μπορούν να αλλάξουν, να μετακινήσουν ή να διαγράψουν τα περιεχόμενα ή την εμφάνισή του. Ακόμα, αν ένα `id` προστεθεί στο URL μιας σελίδας, παρέχει ένα μοναδικό αναγνωριστικό για ένα τμήμα της σελίδας.
- Η ιδιότητα `class` παρέχει τη δυνατότητα ταξινόμησης παρόμοιων αντικειμένων στην ίδια κλάση. Μπορεί να χρησιμοποιηθεί για να αποδώσει κάποια σημασία στο στοιχείο, ή για σκοπούς εμφάνισης. Ουσιαστικά μορφοποιεί το κείμενο με βάση κάποια κοινά χαρακτηριστικά οπου σε αυτό μπορούν να προστεθούν κάποιες κοινές ιδιότητες. Για παράδειγμα, ένα έγγραφο HTML μπορεί να χρησιμοποιεί την επισήμανση `class="test"` σε μερικά στοιχεία για να ξεχωρίσει από το υπόλοιπο κείμενο του εγγράφου. Κατά την εμφάνιση του εγγράφου, αυτά τα στοιχεία μπορεί για παράδειγμα το κείμενο που είναι μέσα σε αυτή τη κλάση να είναι με έντονο χρώμα. Επίσης οι ιδιότητες `class` χρησιμοποιούνται σημασιολογικά στα `microformat`. Ένα στοιχείο μπορεί να έχει πολλαπλές κλάσεις, για παράδειγμα το `class="test level"` βάζει το στοιχείο τόσο στην κλάση `test` όσο και στην `level` ή ακόμα να έχουμε κλάσεις μέσα σε άλλες δηλαδή ως υποκατηγορίες.
- Η ιδιότητα `style` εφαρμόζει στυλ εμφάνισης σε συγκεκριμένα στοιχεία. Θεωρείται καλύτερη τακτική να χρησιμοποιούνται οι ιδιότητες `id` ή `class` ώστε να επιλέγεται το στοιχείο μέσα σε ένα CSS, αλλά μερικές φορές μπορεί να είναι πιο απλό να ανατεθούν `style` κατευθείαν στο στοιχείο χωρίς να χρειασθεί να γράψουμε επιπλέον κώδικα. Γενικά καλή τακτική είναι πάντως να διαχωρίζουμε το `style` και μάλιστα σε ξεχωριστό αρχείο
- Η ιδιότητα `title` προσθέτει μια εξήγηση στο στοιχείο στο οποίο εφαρμόζεται. Στους περισσότερους φυλλομετρητές αυτή η ιδιότητα εμφανίζεται ως αναδυόμενο παράθυρο βοήθειας.

Παράδειγμα file Test2.shtml

```
<form id="Test2" method="post" runat="server">
  <div class="questions">

    <div class="question1">
      <br><b>
        A card is chosen at random from a pack of 52 playing cards.
        What is the probability of a King or a Queen?
```

Διαδραστική Εφαρμογή Ιστού Εκπαιδευτικού Περιεχομένου

```

        </b>
        <div class="answers1">
          <div class="answer1">
            <input type="radio" name="question-1-answers-A" id="1-A"
value="A" />
            <label for="question-1-answers-A">2/13</label>
          </div>
          <div class="answer2">
            <input type="radio" name="question-1-answers-B" id="1-B"
value="B" />
            <label for="question-1-answers-B">1/2</label>
          </div>
          <div class="answer3">
            <input type="radio" name="question-1-answers-C" id="1-C"
value="C" />
            <label for="question-1-answers-C">1/52</label>
          </div>
          <div class="answer4">
            <input type="radio" name="question-1-answers-D" id="1-D"
value="D" />
            <label for="question-1-answers-D">1/26</label>
          </div>
        </div>
      </div>

      <div class="question2">
        <br><b>Which of the following pairs of events is mutually
exclusive?</b>
        <div class="answers2">
          <div class="answer1">
            <input type="radio" name="question-2-answers-A" id="2-A"
value="A" />
            <label for="question-2-answers-A">Run backwards and
forwards at the same time</label>
          </div>
          <div class="answer2">
            <input type="radio" name="question-2-answers-B" id="2-B"
value="B" />
            <label for="question-2-answers-B">Two dice: Odd and
even</label>
          </div>
          <div class="answer3">
            <input type="radio" name="question-2-answers-C" id="2-C"
value="C" />
            <label for="question-2-answers-C">Sit down and listen
music</label>
          </div>
          <div class="answer4">
            <input type="radio" name="question-2-answers-D" id="2-D"
value="D" />
            <label for="question-2-answers-D">Cards: Aces and
Spades</label>
          </div>
        </div>
      </div>
    </div>

```

```

    <div class="question3">
      <br><b>A die is thrown once. What is the probability that the
score is odd but not 3?</b>
      <div class="answers3">
        <div class="answer1">
          <input type="radio" name="question-3-answers-A" id="3-A"
value="A" />
          <label for="question-3-answers-A">1/3</label>
        </div>
        <div class="answer2">
          <input type="radio" name="question-3-answers-B" id="3-B"
value="B" />
          <label for="question-3-answers-B">1/2</label>
        </div>
        <div class="answer3">
          <input type="radio" name="question-3-answers-C" id="3-C"
value="C" />
          <label for="question-3-answers-C">2/3</label>
        </div>
        <div class="answer4">
          <input type="radio" name="question-3-answers-D" id="3-D"
value="D" />
          <label for="question-3-answers-D">0</label>
        </div>
      </div>
    </div>

    <div class="question4">
      <br><b>If P(A)= 4/5.What is the value of P(A')?</b>
      <div class="answers4">
        <div class="answer1">
          <input type="radio" name="question-4-answers-A" id="4-A"
value="A" />
          <label for="question-4-answers-A">1/5</label>
        </div>
        <div class="answer2">
          <input type="radio" name="question-4-answers-B" id="4-B"
value="B" />
          <label for="question-4-answers-B">2/5</label>
        </div>
        <div class="answer3">
          <input type="radio" name="question-4-answers-C" id="4-C"
value="C" />
          <label for="question-4-answers-C">1/3</label>
        </div>
        <div class="answer4">
          <input type="radio" name="question-4-answers-D" id="4-D"
value="D" />
          <label for="question-4-answers-D">1/4</label>
        </div>
      </div>
    </div>

```

```

        <div class="question5">
            <br><b>What is the population standard deviation for the numbers:
11,18,235,192,-10,-102?</b>
            <div class="answers5">
                <div class="answer1">
                    <input type="radio" name="question-5-answers-A" id="5-A"
value="A" />
                    <label for="question-5-answers-A">117.84</label>
                </div>
                <div class="answer2">
                    <input type="radio" name="question-5-answers-B" id="5-B"
value="B" />
                    <label for="question-5-answers-B">171.78</label>
                </div>
                <div class="answer3">
                    <input type="radio" name="question-5-answers-C" id="5-C"
value="C" />
                    <label for="question-5-answers-C">-104.71</label>
                </div>
                <div class="answer4">
                    <input type="radio" name="question-5-answers-D" id="5-D"
value="D" />
                    <label for="question-5-answers-D">181.2</label>
                </div>
            </div>
        </div>
        <br>
        <br>
        <input type="submit" onclick="validateTest();" value="submit"
name="submit" class="button"/>
        <input type="reset" value="Reset" class="button"/>

```

2.7.2 CSS

Η CSS σημαίνει Cascading Style Sheets. Πρόκειται για μια γλώσσα φύλλου στυλ που χρησιμοποιείται για να περιγράψει την εμφάνιση και τη μορφοποίηση ενός εγγράφου γραμμένο σε γλώσσα σήμανσης. Παρέχει μια πρόσθετη δυνατότητα στην HTML. Χρησιμοποιείται γενικά με HTML για να αλλάξει το στυλ των ιστοσελίδων και των διεπαφών χρήστη. Μπορεί επίσης να χρησιμοποιηθεί με οποιοδήποτε είδος εγγράφων XML. Το CSS χρησιμοποιείται μαζί με HTML και JavaScript στις περισσότερες ιστοσελίδες για τη δημιουργία εφαρμογών ιστού και για εφαρμογές κινητής τηλεφωνίας. Οι βασικές λειτουργία για το CSS είναι μορφοποιεί και να αλλάζει την εικόνα του html αρχείου. Τα τρία βασικά πλεονεκτήματα χρήσης CSS

- Πριν από το CSS, ετικέτες όπως γραμματοσειρά, χρώμα, στυλ φόντου, ευθυγραμμίσεις στοιχείων, σύνορα και μέγεθος χρειάστηκε να επαναληφθούν σε κάθε ιστοσελίδα. Αυτή ήταν μια πολύ μεγάλη διαδικασία. Για παράδειγμα: Εάν αναπτύσσετε μια μεγάλη ιστοσελίδα όπου οι γραμματοσειρές και οι πληροφορίες χρώματος προστίθενται σε κάθε σελίδα, θα γίνει μια μακρά και δαπανηρή διαδικασία. Το CSS δημιουργήθηκε για να λύσει αυτό το πρόβλημα.
- Οι ορισμοί στυλ CSS αποθηκεύονται σε εξωτερικά αρχεία CSS, ώστε να είναι δυνατή η αλλαγή ολόκληρου του ιστοτόπου αλλάζοντας μόνο ένα αρχείο.

- Το CSS παρέχει επιπλέον λειτουργίες και χαρακτηριστικά από το απλό HTML για να καθορίσει την εμφάνιση και την αίσθηση του ιστοτόπου.

Παράδειγμα file Tests.css

```
.circle {
  display: table-cell;
  height: 120px;
  width: 120px;
  cursor: pointer;
  text-align: center;
  vertical-align: middle;
  border-radius: 50%;
  background: green;
  color: #fff;
  font: 18px "josefin sans", arial;
}

#MainTest {
  position: relative;
  height: 100vh;
}

#level1 {
  position: absolute;
  top: 20%; /*[wherever you want it]*/
  left: 50%;
}

#level2 {
  position: absolute;
  top: 60%;
  left: 35%;
}

#level3 {
  position: absolute;
  top: 60%;
  left: 65%;
}

#level4 {
  position: absolute;
  top: 100%;
  left: 35%;
}

#level5 {
  position: absolute;
  top: 100%;
  left: 65%;
}
```

```
.Certificate {  
    position: absolute;  
    left: 50%;  
    top: 140%;  
    color: cornflowerblue;  
}  
  
.button {  
    background-color: white;  
    color: black;  
    border: 2px solid #555555;  
    padding: 10px;  
    width: 100px;  
}  
  
.button:hover {  
    background-color: #3B5998;  
    color: white;  
}
```

2.7.3 JavaScript

Η γλώσσα προγραμματισμού JavaScript δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, η Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξή της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον Σεπτέμβριο του 1995. Η LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3

Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. Η JScript πρόσθεσε νέους μεθόδους για να διορθώσει τα Y2K-προβλήματα στην JavaScript, οι οποίοι βασίστηκαν στην java.util.Date τάξη της Java.JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996. Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μία οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript. Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (μαζί με άλλους λόγους). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των παισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web. Τον Ιανουάριο του 2009, το έργο

CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. server-side). Η JavaScript είναι μια γλώσσα προγραμματισμού που βασίζεται σε αντικείμενα, η οποία είναι ελαφριά και cross-platform. Η JavaScript δεν είναι γλώσσα που χρειάζεται compiler, αλλά είναι μια μεταφρασμένη γλώσσα. Ο Μεταφραστής JavaScript (ενσωματωμένος στο πρόγραμμα περιήγησης) είναι υπεύθυνος για τη μετάφραση του κώδικα JavaScript για το πρόγραμμα περιήγησης ιστού. Η αρχική έκδοση της JavaScript έχει αρκετά στοιχεία από τη σύνταξη της γλώσσας προγραμματισμού C, ωστόσο κατά την εξέλιξη της έχει προσθέσει αρκετά στοιχεία από σύγχρονες γλώσσες προγραμματισμού.

Έτσι λοιπόν, μπορούμε να δημιουργήσουμε διαδραστικά web site, όπου κυρίως χρησιμοποιείτε :

- Υπολογισμούς με δυναμικό τρόπο
- Drop-down menus
- Εμφάνιση ώρας-ημερομηνίας
- Εμφάνιση διαφόρων pop-up windows κτλ.

Ωστόσο δεν είναι μόνο ιστοσελίδες που χρησιμοποιούν JavaScript. Υπάρχουν πολλά προγράμματα από τη πλευρά του πελάτη (server programs) που τη χρησιμοποιούν. Node.js είναι η πιο γνωστή. Επιπλέον κάποιες βάσεις δεδομένων όπως MongoDB, τη χρησιμοποιούν ως γλώσσα προγραμματισμού.

Παράδειγμα υπολογισμού μέτρων θέσης και διασποράς με χρήση JavaScript

```
function Statistics() {  
  
    var x = $("#data").val().split(',').map(Number);  
  
    var avg;  
    var med;  
    var st_d;  
    var v;  
  
    //invalid input...  
    for (var i in x) {  
        if (Number.isNaN(x[i]) || (x[i] === "")) {  
            alert("Incorrect input data");  
            exit();  
        }  
    }  
  
    //average  
    var sum = 0  
    for (var i in x)  
    {  
        sum = sum + x[i];  
    }  
    avg = sum / x.length;  
  
    //median  
    x = x.sort();
```

```
if (x.length % 2 == 0) {
    med = (x[(x.length) / 2 - 1] + x[(x.length) / 2]) / 2.0;
}
else {
    med = x[parseInt(x.length / 2)];
}

//variance
var sumS = 0;
for (var j in x) {
    sumS = sumS + Math.pow(x[j] - avg, 2);
}

v = sumS / (x.length);

//standard_deviation
st_d = Math.sqrt(v);

//modes
var modes = [], count = [], i, number, maxIndex = 0;

for (i = 0; i < x.length; i += 1) {
    number = x[i];
    count[number] = (count[number] || 0) + 1;
    if (count[number] > maxIndex) {
        maxIndex = count[number];
    }
}

for (i in count)
    if (count.hasOwnProperty(i) && count[i] === maxIndex) {
        modes.push(Number(i));
    }
}
```

Παράδειγμα υπολογισμού σωστών απαντήσεων στα διαγωνίσματα

Όπως είδαμε πιο πάνω κάθε μια ερώτηση και πιθανή απάντηση έχει μοναδικό id. Επιπλέον από τη προηγούμενη μέθοδο οι ερωτήσεις και οι απαντήσεις ανακατεύονται κάθε φορά. Ένας απλός τρόπος υπολογισμού των σωστών απαντήσεων που δίνει ο χρήστης είναι κάθε φορά η ερώτηση με το πρώτο id να είναι σωστή. Στην πραγματικότητα εφόσον κάθε φορά οι ερωτήσεις και οι απαντήσεις αλλάζουν είναι αδύνατο κάθε φορά να είναι σωστή απάντηση η πρώτη. Η απλή αυτή μέθοδος λειτουργεί για όλα τα διαγωνίσματα σε όλα τα επίπεδα

```
function validateTest() {
    var count = 0;
    for(var i =1;i<6;i++){
        if (document.getElementById(i+"-A").checked) { count++; }
    }
    alert("Score: "+count+"/5");
}
```

2.7.4 jQuery βιβλιοθήκη

jQuery είναι μια βιβλιοθήκη της JavaScript και κατά κύριο λόγο δημιουργήθηκε για να απλοποιεί της διάφορες εντολές της JavaScript χρησιμοποιείτε σχεδόν σε όλους τις μεγάλες ιστοσελίδες και είναι ελεύθερο λογισμικό

Μερικά από τα χαρακτηριστικά της που αποτελούνε και πλεονεκτήματα

- Είναι cross-platform δηλαδή μπορεί να χρησιμοποιηθεί σε όλα τα λειτουργικά
- Απλοποιεί αρκετά κλήσεις από τη πλευρά του server με AJAX
- Έχει αρκετά επιπλέον events
- Υπάρχει η δυνατότητα να επεκταθεί μέσω plug-ins
- Υποστηρίζεται από τους περισσότερους browser

Η jQuery χρησιμοποιήθηκε στην εφαρμογή αυτή σε κάποιες περιπτώσεις ενδεικτικά

Ο συγκεκριμένος κώδικας αφού προηγουμένως έχει υπολογισθεί με JavaScript μετρά θέση και διασποράς η jQuery χρησιμοποιείτε για να παρουσιάσει τα αποτελέσματα με βάση το id

```
$("#max").html("<b>Maximum:</b> " + Math.max.apply(Math, x));
$("#min").html("<b>Minimum:</b> " + Math.min.apply(Math, x));
$("#count").html("<b>Count:</b> " + x.length);
$("#sum").html("<b>Sum:</b> " + sum);
$("#average").html("<b>Average:</b> " + Math.round(avg * 100) / 100);
$("#median").html("<b>Median:</b> " + Math.round(med * 100) / 100);
$("#mode").html("<b>Modes:</b>" + modes);
$("#variance").html("<b>Variance:</b> " + Math.round(v * 100) / 100);
$("#standard_deviation").html("<b>Standard Deviation:</b> " + Math.round(st_d
* 100) / 100);
```

Η επόμενη συνάρτηση μας δίνει σε τυχαία σειρά από κάθε διαγώνισμα τόσο τις ερωτήσεις όσο και τις απαντήσεις

```
$(document).ready(function () {
    for (i = 1; i <= 5; i++) {
        var parent = $(".answers" + i);
        var divs = parent.children();
        while (divs.length) {
            parent.append(divs.splice(Math.floor(Math.random() * divs.length),
1)[0]);
        }
        var parent = $(".questions");
        var divs = parent.children();
        while (divs.length) {
            parent.append(divs.splice(Math.floor(Math.random() * divs.length),
1)[0]);
        }
    }
});
```

Στη παρακάτω μέθοδο με χρήση jQuery όταν ο χρήστης έχει τη δυνατότητα να διαλέξει μια μόνο από τις πιθανές απαντήσεις στην ενότητα των διαγωνισμάτων.

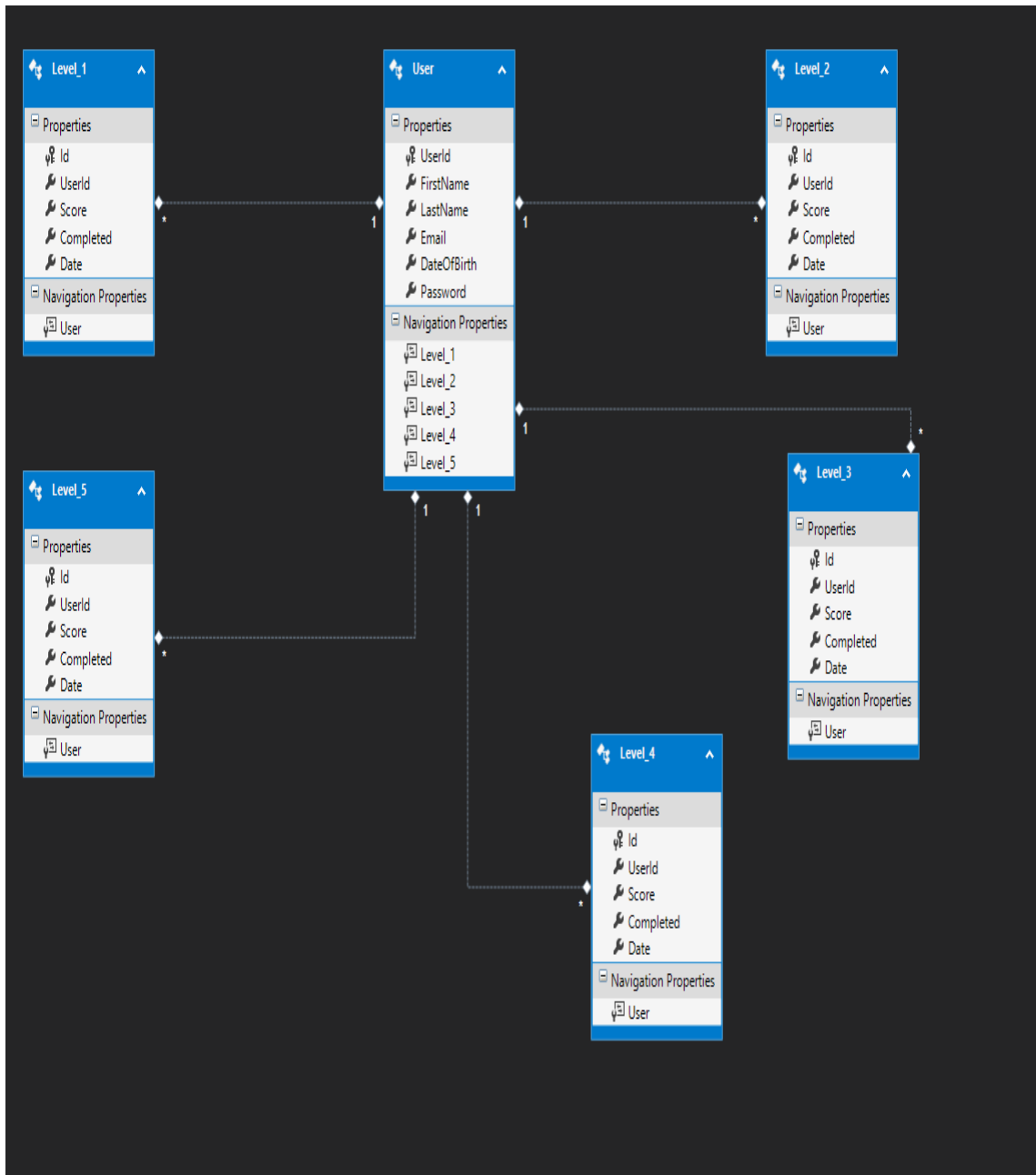
```
$(document).ready(function () {
    $('input[type=radio]').change(function () {
        if (this.checked) {
            $(this).closest('.question')
                .find('input[type=radio]').not(this)
                .prop('checked', false);
        }
    });
});
```

2.8 Σχεδιασμός και υλοποίηση Βάσης Δεδομένων

Στο παρακάτω σχήμα βλέπουμε το σχεδιασμό της Βάσης Δεδομένων. Ο κάθε χρήστης έχει ένα μοναδικό αναγνωριστικό (UserId). Επίσης όνομα, επίθετο, email, ημερομηνία γέννησης και κωδικό. Οπού ο πίνακας αυτό συνδέεται με τους υπολοίπους πίνακες με βάση το UserId . Στους πίνακες Level_1, Level_2, Level_3, Level_4, Level_5

Υπάρχουν οι εγγραφές

- Id. Μοναδικό αναγνωριστικό (Πρωτεύον κλειδί)
- UserId. το κλειδί που αντιστοιχεί από το πίνακα User(Ξένο κλειδί)
- Score. Το αποτέλεσμα που έλαβε ο χρήστης από το τεστ
- Completed. Αν η βαθμολογία ήταν πάνω από 50%
- Date. Η ημερομηνία που συμμετείχε στο τεστ



2.8.1 Βάση Δεδομένων Microsoft SQL Server

SQL Server είναι λογισμικό (Σχεσιακό Σύστημα Διαχείρισης Βάσεων Δεδομένων) που αναπτύχθηκε από τη Microsoft. Ονομάζεται επίσης MS SQL Server. Εφαρμόζεται από τις προδιαγραφές των σχεσιακών βάσεων δεδομένων που οι πίνακες συνδέονται με πρωτεύοντα (μοναδικά) και ξένα κλειδιά δηλαδή κλειδιά που αναφέρονται σε πρωτεύοντα. Οι σχεσιακές βάσεις δεδομένων αναφέρονται ως RDBMS (Relational Database Management System)

Χαρακτηριστικά του MS SQL Server

- υλοποιείται από τις προδιαγραφές του RDBMS.
- είναι επεκτάσιμος.
- ανεξάρτητος από την πλατφόρμα.
- υποστηρίζει τη γλώσσα SQL

Υλοποίηση πινάκων User, Level_1, Level_2, Level_3, Level_4, Level_5

```
CREATE TABLE [dbo].[User] (
    [UserId] INT IDENTITY (1, 1) NOT NULL,
    [FirstName] NVARCHAR (50) NOT NULL,
    [LastName] NVARCHAR (50) NOT NULL,
    [Email] NVARCHAR (50) NOT NULL,
    [DateOfBirth] DATE NOT NULL,
    [Password] NVARCHAR (MAX) NOT NULL,
    PRIMARY KEY CLUSTERED ([UserId] ASC)
);
```

```
CREATE TABLE [dbo].[Level 1] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
    [UserId] INT NOT NULL,
    [Score] INT DEFAULT ((0)) NOT NULL,
    [Completed] BIT DEFAULT ((0)) NOT NULL,
    [Date] DATE NOT NULL,
    PRIMARY KEY CLUSTERED ([Id] ASC),
    FOREIGN KEY ([UserId]) REFERENCES [dbo].[User] ([UserId]) ON DELETE CASCADE
ON UPDATE CASCADE
);
```

```
CREATE TABLE [dbo].[Level 2] (
    [Id] INT IDENTITY (1, 1) NOT NULL,
```

```

[UserId]    INT NOT NULL,
[Score]     INT DEFAULT ((0)) NOT NULL,
[Completed] BIT DEFAULT ((0)) NOT NULL,
[Date]      DATE NOT NULL,
PRIMARY KEY CLUSTERED ([Id] ASC),
FOREIGN KEY ([UserId]) REFERENCES [dbo].[User] ([UserId]) ON DELETE CASCADE
ON UPDATE CASCADE
);

```

```

CREATE TABLE [dbo].[Level 3] (
[Id]        INT IDENTITY (1, 1) NOT NULL,
[UserId]    INT NOT NULL,
[Score]     INT DEFAULT ((0)) NOT NULL,
[Completed] BIT DEFAULT ((0)) NOT NULL,
[Date]      DATE NOT NULL,
PRIMARY KEY CLUSTERED ([Id] ASC),
FOREIGN KEY ([UserId]) REFERENCES [dbo].[User] ([UserId]) ON DELETE CASCADE
ON UPDATE CASCADE
);

```

```

CREATE TABLE [dbo].[Level 4] (
[Id]        INT IDENTITY (1, 1) NOT NULL,
[UserId]    INT NOT NULL,
[Score]     INT DEFAULT ((0)) NOT NULL,
[Completed] BIT DEFAULT ((0)) NOT NULL,
[Date]      DATE NOT NULL,
PRIMARY KEY CLUSTERED ([Id] ASC),
FOREIGN KEY ([UserId]) REFERENCES [dbo].[User] ([UserId]) ON DELETE CASCADE
ON UPDATE CASCADE
);

```

```

CREATE TABLE [dbo].[Level 5] (
[Id]        INT IDENTITY (1, 1) NOT NULL,
[UserId]    INT NOT NULL,
[Score]     INT DEFAULT ((0)) NOT NULL,
[Completed] BIT DEFAULT ((0)) NOT NULL,
[Date]      DATE NOT NULL,
PRIMARY KEY CLUSTERED ([Id] ASC),
FOREIGN KEY ([UserId]) REFERENCES [dbo].[User] ([UserId]) ON DELETE CASCADE
ON UPDATE CASCADE
);

```

2.8.2 Πρόσβαση και διαχείριση δεδομένων Entity Framework

Η τεχνολογία που επιλέχθηκε είναι το Entity Framework έκδοση 6.2.0. Entity Framework είναι ένα πλαίσιο ORM (object-relational mapping) όπου πρακτικά σημαίνει της μετατροπή των δεδομένων χρησιμοποιώντας τεχνικές αντικειμενοστρεφών γλωσσών. Χρησιμοποιείτε για εφαρμογές .NET που υποστηρίζονται από τη Microsoft. Επιτρέπει στους προγραμματιστές να δουλεύουν με δεδομένα χρησιμοποιώντας αντικείμενα κατηγοριών συγκεκριμένων τομέων χωρίς να εστιάζουν στους υποκείμενους πίνακες βάσης δεδομένων και στις στήλες όπου αποθηκεύονται αυτά τα δεδομένα. Με το Πλαίσιο Οντοτήτων, οι προγραμματιστές μπορούν να

εργαστούν σε υψηλότερο επίπεδο αφαίρεσης όταν ασχολούνται με δεδομένα και μπορούν να δημιουργούν και να διατηρούν εφαρμογές προσανατολισμένες σε δεδομένα με λιγότερους κώδικες σε σύγκριση με τις παραδοσιακές εφαρμογές.

Επίσημος ορισμός από Microsoft: Το Entity Framework είναι ένας αντικειμενικός σχεσιακός χαρτογράφος (O / RM) ο οποίος επιτρέπει στους προγραμματιστές .NET να συνεργάζονται με μια βάση δεδομένων χρησιμοποιώντας αντικείμενα .NET. Εξαλείφει την ανάγκη για το μεγαλύτερο μέρος του κώδικα πρόσβασης δεδομένων που συνήθως πρέπει να γράψουν οι προγραμματιστές.

Χαρακτηριστικά:

- Cross-platform: Το entity framework μπορεί να λειτουργήσει σε Windows, Linux και Mac.
- Μοντελοποίηση: Το EF (Entity Framework) δημιουργεί ένα EDM (Μοντέλο δεδομένων οντοτήτων) που βασίζεται σε οντότητες POCO (Plain Old CLR Object) με ιδιότητες get / set διαφορετικών τύπων δεδομένων. Χρησιμοποιεί αυτό το μοντέλο όταν ερωτά ή αποθηκεύει δεδομένα οντότητας στην υποκείμενη βάση δεδομένων.
- Πρόσβαση: Το EF επιτρέπει τη χρήση ερωτημάτων LINQ (C # / VB.NET) για την ανάκτηση δεδομένων από την υποκείμενη βάση δεδομένων. Ο πάροχος βάσης δεδομένων θα μεταφράσει αυτά τα ερωτήματα LINQ στη γλώσσα ερωτήματος που αφορά τη συγκεκριμένη βάση δεδομένων (π.χ. SQL για μια σχεσιακή βάση δεδομένων). Το EF μας επιτρέπει επίσης να εκτελούμε ερωτήσεις SQL χωρίς να το κάνουμε απευθείας στη βάση δεδομένων.
- Αποθήκευση: Το EF εκτελεί εντολές INSERT, UPDATE και DELETE στη βάση δεδομένων με βάση τις αλλαγές που έγιναν στις οντότητες σας όταν καλείτε τη μέθοδο SaveChanges (). Το EF παρέχει επίσης τη μέθοδο ασύγχρονης μεθόδου SaveChangesAsync ().

2.8.3 Αποθήκευση και ανάκτηση δεδομένων LINQ

Η τεχνολογία που επιλέχθηκε είναι το LINQ. Το Integrated Query Language (LINQ) είναι μια ισχυρή γλώσσα ερωτημάτων που ξεκίνησε με το .Net 3.5 & Visual Studio 2008. Το LINQ μπορεί να χρησιμοποιηθεί με C # ή Visual Basic για την αναζήτηση διαφορετικών πηγών δεδομένων. Το LINQ (Integrated Query Language) είναι ομοιόμορφη σύνταξη ερωτήματος στη C # και VB.NET που χρησιμοποιείται για την αποθήκευση και την ανάκτηση δεδομένων από διαφορετικές πηγές. Είναι ενσωματωμένο στο C # ή VB, εξαλείφοντας έτσι την αναντιστοιχία μεταξύ των γλωσσών προγραμματισμού και των βάσεων δεδομένων, καθώς και παρέχοντας μια ενιαία διεπαφή ερωτήματος για διαφορετικούς τύπους πηγών δεδομένων.

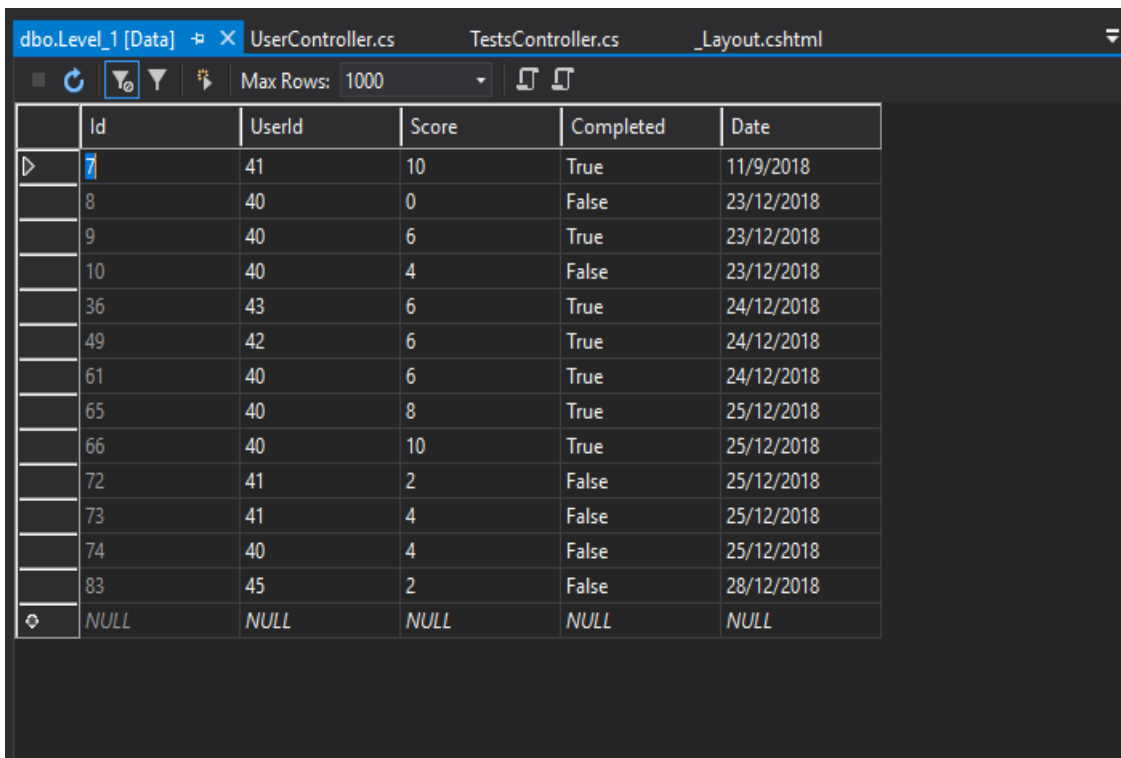
Για παράδειγμα, η SQL είναι μια δομημένη γλώσσα ερωτήματος που χρησιμοποιείται για την αποθήκευση και την ανάκτηση δεδομένων από μια βάση δεδομένων. Με τον ίδιο τρόπο, το LINQ είναι μια δομημένη σύνταξη ερωτήματος ενσωματωμένη στη C # και στη VB.NET που χρησιμοποιείται για την αποθήκευση και την ανάκτηση δεδομένων από διαφορετικούς τύπους πηγών δεδομένων όπως συλλογή αντικειμένων με λίστες, βάση δεδομένων SQL Server, XML, Web Services κτλ. Το LINQ λειτουργεί πάντα με αντικείμενα, ώστε να μπορείτε να χρησιμοποιείτε τα ίδια βασικά πρότυπα κωδικοποίησης για την ερώτηση και τη μετατροπή δεδομένων σε έγγραφο XML, βάσεις δεδομένων SQL, συλλογές δεδομένων, .NET και οποιαδήποτε άλλη μορφή για την οποία είναι διαθέσιμος ένας παροχέας LINQ.

Μπορεί να υλοποιηθεί με σύνταξη μεθόδου ή τη σύνταξη ερωτήματος για πρόσβαση στην Βάση Δεδομένων (Method Syntax or Query Syntax)

Παραδείγματα χρήσης LINQ

- Αποθήκευση βαθμολογία χρήστη μετά την ολοκλήρωση ενός διαγωνίσματος

```
var author1 = new Statistics.Models.Level_1 { Score = 2 * count,
Completed = isCompleted, UserId = (int)Session["UserId"], Date =
DateTime.UtcNow.Date };
context.Level_1.Add(author1);
context.SaveChanges();
```



Id	UserId	Score	Completed	Date
7	41	10	True	11/9/2018
8	40	0	False	23/12/2018
9	40	6	True	23/12/2018
10	40	4	False	23/12/2018
36	43	6	True	24/12/2018
49	42	6	True	24/12/2018
61	40	6	True	24/12/2018
65	40	8	True	25/12/2018
66	40	10	True	25/12/2018
72	41	2	False	25/12/2018
73	41	4	False	25/12/2018
74	40	4	False	25/12/2018
83	45	2	False	28/12/2018
NULL	NULL	NULL	NULL	NULL

Στο παραπάνω πίνακα στη πρώτη στήλη είναι το μοναδικό Id του χρήστη (πρωτεύον κλειδί), το UserId που αντιστοιχεί στο πίνακα με τους χρήστες (είναι ξένο κλειδί). Το Score που έλαβε ο χρήστης, Completed αν ολοκλήρωσε επιτυχώς το διαγώνισμα και τέλος Date η ημερομηνία που συμμετείχε στο διαγώνισμα.

- Ερώτημα προς τη βάση αν ο συνδεδεμένος χρήστης έχει ολοκληρώσει ευτυχώς το Level 1

```
Statistics.Models.UserDatabaseEntities user = new
Statistics.Models.UserDatabaseEntities();
    var userId = Session["UserId"].ToString();
    var completedLevel1 = user.Level_1.Where(a=>
a.UserId.ToString() == userId && a.Completed ==
true).FirstOrDefault();
```

- Ερώτημα προς βάση για χρήστες που έχουν ολοκληρώσει όλα τα επίπεδα διαγωνισμάτων με βαθμολογία πάνω από 50%. Εμφάνιση όνομα, επίθετο, email και ημερομηνία ολοκλήρωσης των διαγωνισμάτων κατά φθίνουσα σειρά

```
Statistics.Models.UserDatabaseEntities certifiedUsers = new
Statistics.Models.UserDatabaseEntities();

var res = from t1 in certifiedUsers.Level_5
join t2 in certifiedUsers.User on t1.UserId equals t2.UserId
where (t1.Completed.ToString() == "True")
group t1 by new
{
    UserId = t1.UserId,
    FirstName = t2.FirstName,
    LastName = t2.LastName,
    Email = t2.Email,
    Date = t1.Date
}
into g
orderby g.Key.Date descending
select new
{
    FirstName = g.Key.FirstName,
    LastName = g.Key.LastName,
    Email = g.Key.Email,
    Date = g.Key.Date
};
```

- Επίσης μπορούμε να χρησιμοποιήσουμε LINQ μέσα στο Controller για να ελέγξουμε παράδειγμα αν ένας χρήστης που κάνει login δίνει σωστά το email και το password. Αναζητούμε λοιπόν στη βάση αν το email (το οποίο είναι μοναδικό) και ο κωδικός μάς έδωσε ο χρήστης υπάρχει στη βάση

```
[ValidateAntiForgeryToken]
public ActionResult LoginForm(UserLogin login)
{
    using (UserDatabaseEntities dc = new UserDatabaseEntities())
    {
        var v = dc.User.Where(a => a.Email == login.Email && a.Password
        == login.Password).FirstOrDefault();
        if(v != null) {
            Session["UserId"] = v.UserId;
            Session["FirstName"] = v.FirstName;
            Session["Email"] = v.Email;

            return RedirectToAction("Index", "Home");
        }
        ModelState.AddModelError("", "Invalid credentials");
        return View(login); }
}
```

Γίνεται σαφές πόσο χρήσιμο εργαλείο είναι το LINQ για πρόσβαση σε δεδομένα πίνακα και επίσης μπορεί να χρησιμοποιηθεί για αρκετά σύνθετα ερωτήματα με σχετικά απλό τρόπο

2.8.4 Εμφάνιση δεδομένων σε γραφική μορφή

Αυτή η ενότητα εξηγεί τον τρόπο εμφάνισης των δεδομένων σε γραφική μορφή, χρησιμοποιώντας τον "Βοηθό πίνακα" Chart Helper

Ο "Βοηθός Χαρτών" μπορεί να δημιουργήσει εικόνες διαγραμμάτων διαφορετικών τύπων με πολλές επιλογές μορφοποίησης και ετικέτες. Μπορεί να δημιουργήσει πρότυπα διαγράμματα, όπως διαγράμματα περιοχής, γραφήματα γραμμών, διαγράμματα στήλης, διαγράμματα γραμμών καθώς και πιο εξειδικευμένα γραφήματα, όπως οι χάρτες μετοχών.

Σε συνδυασμό με LINQ μπορούμε να κάνουμε κάποιο ερώτημα στη βάση και μετά να έχουμε μια γραφική απεικόνιση του αποτελέσματος

Για παράδειγμα θέλουμε για κάθε χρήστη να έχουμε τη βαθμολογία που έχει ανά ημερομηνία έτσι ώστε να βλέπουμε τη πρόοδο του μαθητή

```
Statistics.Models.UserDatabaseEntities showPerformance = new
Statistics.Models.UserDatabaseEntities();
var userId = Session["UserId"].ToString();
var performance = showPerformance.Level_1.Where(a => a.UserId.ToString() ==
userId)
                                .Select(s => new
                                {
                                    score = s.Score,
                                    date = s.Date
                                });

var myChart = new Chart(width: 800, height: 600)
.AddTitle("")
.AddSeries(chartType: "Column", // or Pie
    xValue: performance, xField: "date",
    yValues: performance, yFields: "score")
.AddLegend("Daily Performnce Level1")
.SetYAxis("score", 0, 10)
.Write();
```

Κεφάλαιο 3 Χαρακτηριστικά λειτουργίας εφαρμογής

3.1 Αρχική σελίδα

Στην αρχική σελίδα ο χρήστης έχει το μενού επιλογών, μπορεί να πλοηγηθεί στις ενότητες που περιλαμβάνουν θεωρία (Basic Statistics, Probability Theory) καθώς και στον υπολογιστή στατιστικών. Ωστόσο δεν έχει τη δυνατότητα να κάνει διαγωνίσματα ή να δει με γραφήματα την απόδοση του αν δεν έχει κάνει εγγραφή και είσοδο. Στην παρακάτω εικόνα βλέπουμε τη ενότητα θεωρία Basic Statistics

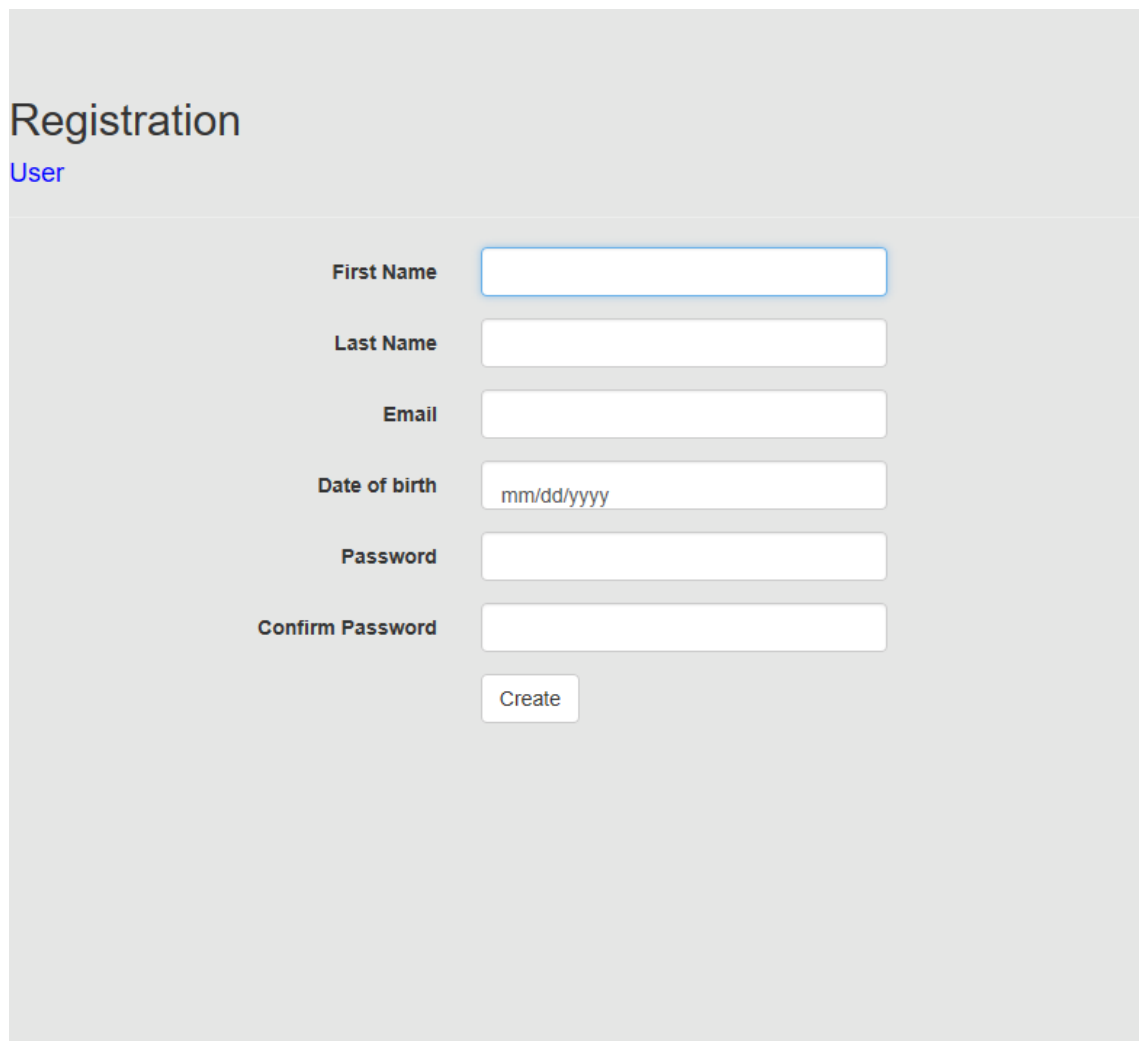
The screenshot displays a web application titled "Statistics Calculator" with a navigation menu including "Basic Statistics", "Probability Theory", "Tests", "My Daily Performance", and "Certified Users". The main content area is divided into sections for different statistical calculations:

- Minimum:** Described as "The smallest value in a sample data set". Example: "Assume that we have values separated by commas such as 3, 1, 5, 9, 7,82, 2, 4, 5". Result: "Minimum value is 1".
- Maximum:** Described as "The largest value in a sample data set: 5,8,7,12,22,3,2". Result: "Maximum value is 22".
- Sum:** Described as "The total of all data values". Formula: $\text{sum} = (X_1 + X_2 + X_3 + \dots + X_n)$. Example: "Data set: 5,7,9,2,-1,2". Result: "Sum = 5+7+9+2+(-1)+2 = 24".
- Count:** Described as "The total number of data values in a data set". Example: "Data set: 4,4,2,11,1". Result: "Count = 5".
- Mean:** Described as "The sum of all of the data divided by the count". Example: "Data set: 3,2,-1,-3,9". Result: "Sum = 10, Count = 5", "Mean = 2". Formula: $\mu = \frac{\sum x_i}{n}, i \in \mathbb{N}$.

A URL bar at the bottom shows "http://localhost:45554/Home/StatisticsCalculator".

3.2 Εγγραφή χρήστη

Στην ενότητα εγγραφής θα πρέπει να συμπληρωθούν όνομα, επίθετο, ημερομηνία γέννησης καθώς και κωδικός χρήστη και επαλήθευση κωδικού. Ο κωδικός θα πρέπει να περιλαμβάνει τουλάχιστον 6 χαρακτήρες και επίσης το email να είναι έγκυρο. Επιπλέον το email να είναι μοναδικό καθώς ελέγχεται αν υπάρχει ίδιο email καταχωρημένο στη βάση δεδομένων

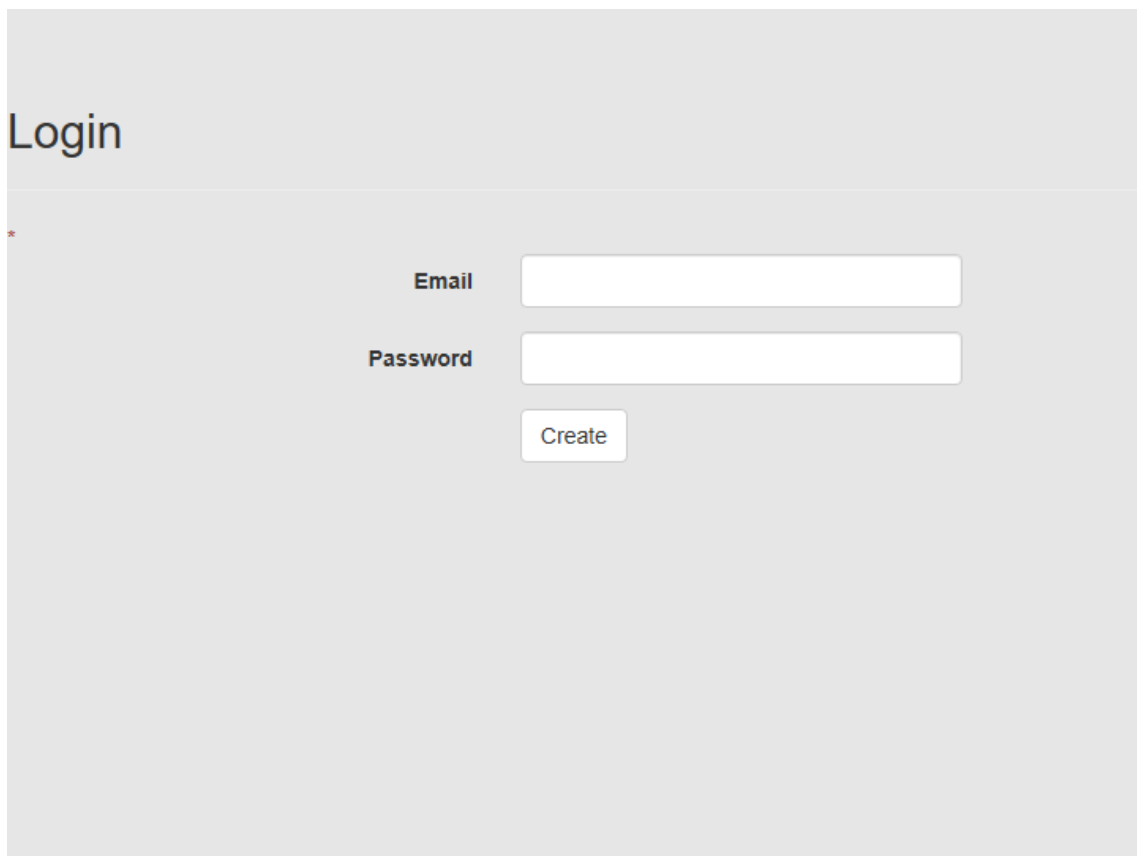


The image shows a registration form titled "Registration" with a sub-section for "User". The form contains the following fields and a button:

- First Name**: A text input field.
- Last Name**: A text input field.
- Email**: A text input field.
- Date of birth**: A text input field with a placeholder "mm/dd/yyyy".
- Password**: A text input field.
- Confirm Password**: A text input field.
- Create**: A button to submit the form.

3.3 Είσοδος χρήστη

Αφού ολοκληρωθεί η εγγραφή πλέον μπορούμε να κάνουμε είσοδο, απλά με το email που δηλώσαμε και το κωδικό. Οπού τα πεδία αυτά πρέπει να αντιστοιχούνε στο χρήστη



The image shows a login form on a light gray background. At the top left, the word "Login" is written in a dark font. Below it, there is a small red asterisk. The form contains two input fields: "Email" and "Password", each with a white rectangular box. Below the "Password" field is a button labeled "Create".

3.4 Ενότητα Θεωρίας

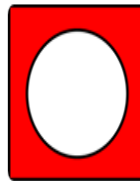
Περιλαμβάνεται στην αρχική σελίδα καθώς και στη ενότητα Probability Theory με το αντίστοιχο υλικό. Ενδεικτικά,

Probability of Two Events

Complement of A and B

Given a probability A, denoted by $P(A)$, it is simple to calculate the complement, or the probability that the event described by $P(A)$ does not occur, $P(A')$. If for example $P(A) = 0.25$ represents the probability that tomorrow is going to rain, we can say that the probability tomorrow not raining is:

$$P(A') = 1 - P(A) = 1 - 0.25 = 0.75$$

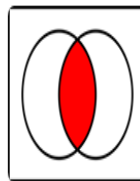


Given this scenario, there is therefore a 75% chance that tomorrow will not rain. Any $P(B')$ would be calculated in the same manner, and it is worth noting that in the calculator above, can be independent, if $P(A) = 0.75$, $P(B)$ does not necessarily have to equal 0.25, it could be everything between 0 and 1

Intersection of A and B

The probability $A \cap B$, read "A-cut B" or "A and B", is performed when A and B are simultaneously executed.

$$P(A \cap B)$$



Mutually exclusive events

In throwing a dice if A is the probability of bringing an even number, and B being able to bring an odd number, we have $A = \{2,4,6\}$ and $B = \{1,3,5\}$. We notice that A and B can not be made at the same time, since they have no common elements. In this case, A and B are said to be incompatible. General: Two contingencies A and B are called incompatible when $A \cap B = \emptyset$

$$P(A \cap B) = 0$$

Independent A and B (Not mutually exclusive events)

In throwing two dices if A is the probability of bringing an even number, and B being able to bring an odd number, we have $A = \{2,4,6\}$ and $B = \{1,3,5\}$. We notice that A and B can be made at the same time. In this case, A and B are said to be independent. $P(A) = 3/6$, $P(B) = 3/6$

$$P(A \cap B) = P(A) \cdot P(B) = 0.5 \cdot 0.5 = 0.25$$

This means that we have probability 25% to bring the same time with one dice even number and with another dice odd number

Conditional probability

3.5 Ενότητα Διαγωνισμάτων

3.5.1 Κεντρικό μενού

Στην ενότητα αυτή ο κάθε χρήστης μπορεί να επιλέξει το επίπεδο που θέλει εφόσον έχει ολοκληρώσει το προηγούμενο με βαθμολογία πάνω από 50%. Σε περίπτωση που δεν έχει κάνει κάποιο διαγώνισμα μπορεί να κάνει από το πρώτο επίπεδο. Δεξιά της εικόνας φαίνεται ότι ο συγκεκριμένος χρήστης έχει ολοκληρώσει όλα τα επίπεδα των διαγωνισμάτων. Σημαντικό επίσης είναι ότι τα διαγωνίσματα ακόμα και στο ίδιο επίπεδο είναι διαφορετικά, έτσι ώστε ο χρήστης να απαντάει με βάση τη πραγματική του γνώση

The screenshot displays a web application interface. At the top, a dark blue navigation bar contains the following links: Basic Statistics, Probability Theory, Statistics Calculator, Tests, My Daily Performance (with a dropdown arrow), Certified Users, Logout, and About. Below the navigation bar, the main content area is divided into two sections. On the left, a central menu consists of six blue oval buttons arranged in a circle, labeled Level 1, Level 2, Level 3, Level 4, Level 5, and Certificate. On the right, a user profile section for 'vaggelis mavropoulos' is shown, with the email address 'vmaurop@gmail.com'. Below the profile, a list of progress indicators shows 'Level1: ✓', 'Level2: ✓', 'Level3: ✓', 'Level4: ✓', and 'Level5: ✓', each with a green checkmark.

3.5.2 Διαγωνίσματα

Τα διαγωνίσματα είναι πολλαπλής επιλογής. Οι ερωτήσεις σε κάθε διαγώνισμα είναι πέντε και μια είναι η σωστή απάντηση. Αφού ο χρήστης ολοκληρώσει το διαγώνισμα ενημερώνετε με ένα pop-up παράθυρο για τη βαθμολογία η οποία καταχωρείτε στη βάση δεδομένων

Test Level 1

If $C = \{1, -3, 15, 9\}$ and $D = \{2, 3, 5, 1, 9\}$, what is $A \cap B$?

- {3,1,15}
- \emptyset
- {9}
- {1,9}

If $A = \{11, 7, 15, 7, 22\}$ and $B = \{22, 3, 5, 7\}$, what is $A \cup B$?

- {3,5}
- {11,7,15,22,3,5}
- {7}
- {11,7}

If $X = \{i, k, l\}$ and $Y = \{o, p, m, n\}$. What is $X \cap Y$?

- 1
- {i, k}
- zero
- \emptyset

A standard deviation of 0 means?

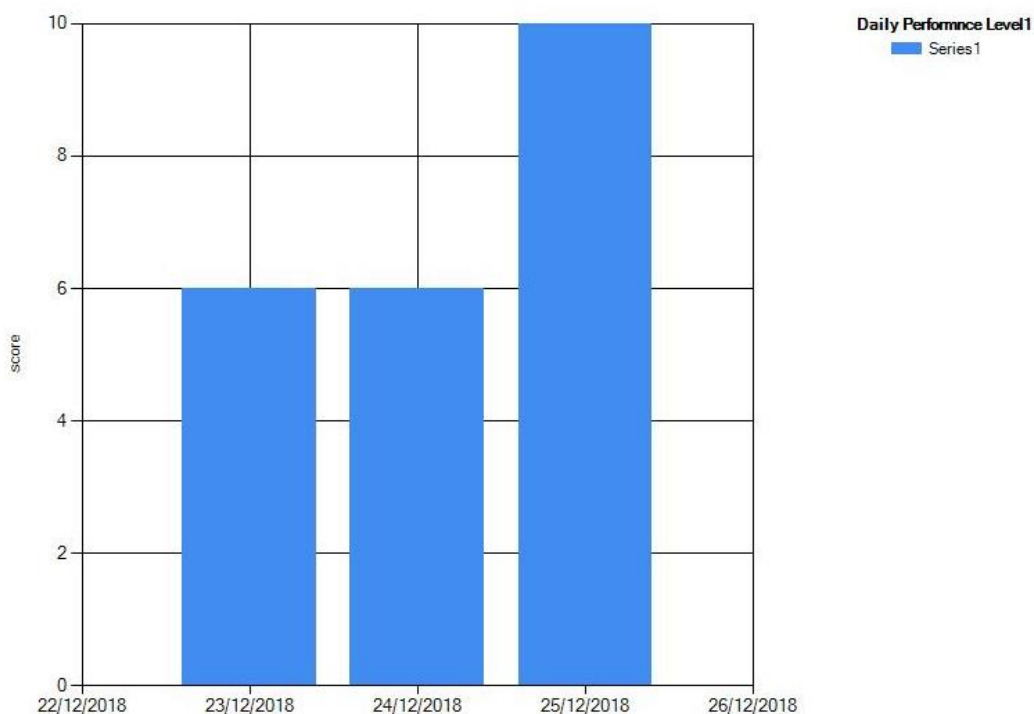
- You did the calculations wrong
- The variance is 1
- You cannot have standard deviation zero
- All values are the same

If you have the variance, how do you get the standard deviation?

- Square it
- Add the mean
- Subtract the mean
- Divide it by the sample size

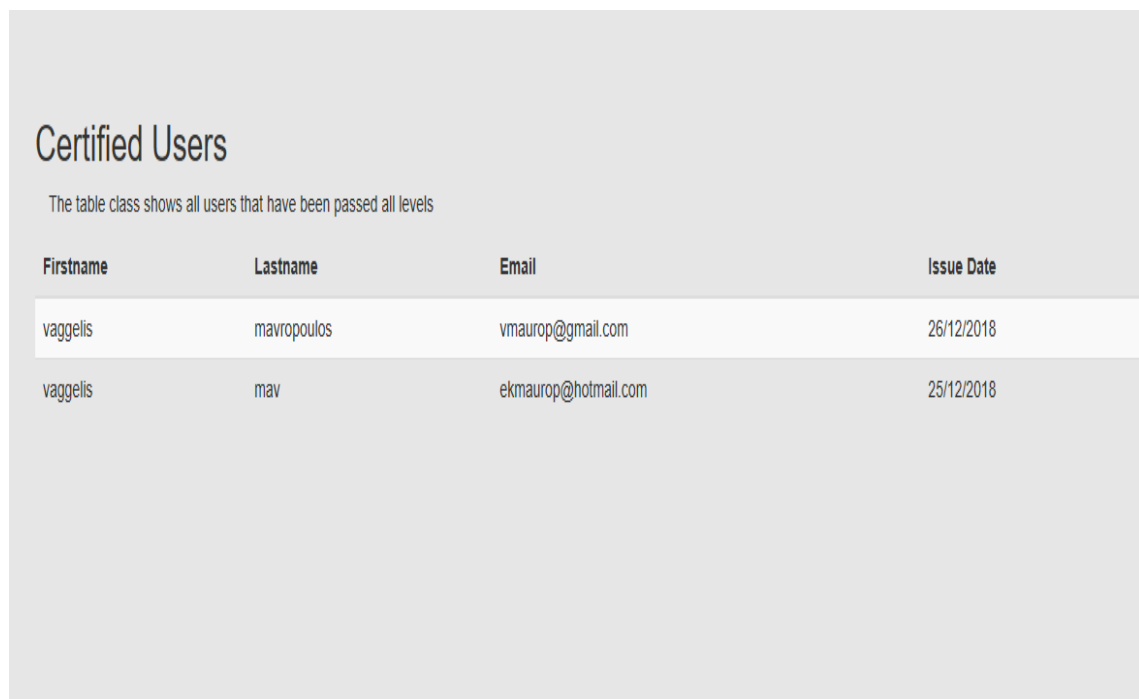
3.6 Έλεγχος επιδόσεων χρήστη

Ο κάθε χρήστης έχει τη δυνατότητα να βλέπει το ιστορικό των επιδόσεων του σε κάθε επίπεδο με μορφή γραφημάτων ημερομηνία-βαθμολογία. Στη παρακάτω εικόνα βλέπουμε το διάγραμμα με βάση βαθμολογία χρήστη ο κάθετος άξονας και ημερομηνία βαθμολογίας ο οριζόντιος άξονας. Τα γραφήματα αυτά είναι χωρισμένα ανά ενότητες, από το κάθε επίπεδο διαγωνισμάτων για το κάθε χρήστη έχουμε γράφημα επιδόσεων. Σε περίπτωση που ο χρήστης δεν έχει συμμετοχή σε κάποια από τα επίπεδα διαγωνισμάτων δεν υπάρχει γράφημα στο συγκεκριμένο επίπεδο



3.7 Χρήστες που έχουν ολοκληρώσει όλα τα διαγωνίσματα επιτυχώς

Όσοι χρήστες έχουν ολοκληρώσει όλες τις ενότητες των διαφορετικών επιπέδων δυσκολίας με βαθμολογία άνω του 50% είναι πιστοποιημένοι χρήστες και μπορούν να δούνε το όνομα τους email και ημερομηνία του επιτεύγματος αυτού



The screenshot displays a web interface with a heading "Certified Users" and a descriptive text: "The table class shows all users that have been passed all levels". Below this is a table with four columns: "Firstname", "Lastname", "Email", and "Issue Date". The table contains two rows of data.

Firstname	Lastname	Email	Issue Date
vaggelis	mavropoulos	vmaurop@gmail.com	26/12/2018
vaggelis	mav	ekmaurop@hotmail.com	25/12/2018

Κεφάλαιο 4 Εκπαιδευτικό περιεχόμενο εφαρμογής

4.1 Βασικά στοιχεία Στατιστικής – Μέτρα Θέσης και Διασποράς

Ο όρος “Στατιστική” ενδεχομένως να προέρχεται από τη λατινική λέξη “status” (πολιτεία, κράτος) η οποία, χρησιμοποιήθηκε αρχικά για το χαρακτηρισμό αριθμητικών δεδομένων που αναφέρονται κυρίως στον πληθυσμό μιας χώρας.

Βασικές έννοιες της Στατιστικής έχουν εισχωρήσει και ενσωματωθεί σε όλες τις επιστήμες. Η ανάλυση στατιστικών ερευνών είναι το κυριότερο εργαλείο έρευνας σε ένα μεγάλο φάσμα εφαρμογών των επιστημών κατά την επεξεργασία δεδομένων χρειαζόμαστε κάποια μέτρα ώστε να μπορούμε να συγκρίνουμε διαφορετικούς πληθυσμούς ή δείγματα μεταξύ τους. Τη δυνατότητα αυτή μας τη δίνουν σε μεγάλο βαθμό τα μέτρα θέσης και τα μέτρα διασποράς. Τα μέτρα θέσης αποτελούν παραμέτρους που δείχνουν γύρω από ποια τιμή τείνουν να συγκεντρωθούν τα δεδομένα μας, δηλαδή δείχνουν την *κεντρική τάση* των δεδομένων μας.

- Μέση τιμή είναι το άθροισμα των στοιχείων του δείγματος δια το πλήθος
- Διάμεσος είναι η μεσαία παρατήρηση του δείγματος εφόσον έχει τοποθετηθεί σε αύξουσα σειρά. Αν το πλήθος των αριθμών είναι άρτιο τότε είναι το ημίαθροισμα των δύο μεσαίων παρατηρήσεων του δείγματος ενώ αν είναι περιττό το πλήθος είναι η μεσαία παρατήρηση
- Διακύμανση είναι το άθροισμά της διαφοράς των τετραγώνων μεταξύ κάθε στοιχείου του δείγματος με τη μέση τιμή διαιρούμενο με το πλήθος του δείγματος
- Τυπική απόκλιση δείγματος είναι το τετράγωνο της διασποράς

4.2 Εισαγωγή στη θεωρία πιθανοτήτων

Το κύριο κίνητρο για την ανάπτυξη της Θεωρίας των Πιθανοτήτων προήλθε από το ενδιαφέρον του ανθρώπου για τα τυχερά παιχνίδια. Σημαντική μάλιστα ώθηση στην ανάπτυξη του κλάδου αυτού των Μαθηματικών αποτέλεσε η γόνιμη αλληλογραφία που αναπτύχθηκε ανάμεσα στους Pascal και Fermat το 17ο αιώνα με αφορμή διάφορα προβλήματα που προέκυψαν από την ενασχόληση του ανθρώπου με τα τυχερά παιχνίδια.

Η ανάπτυξη της Θεωρίας των Πιθανοτήτων οφείλεται επίσης και στις ανάγκες των Φυσικών Επιστημών όπως η εφαρμογή της Θεωρίας Σφαλμάτων σε αστρονομικές παρατηρήσεις. Ιδιαίτερη ανάπτυξη υπήρχε κατά το 18ο αιώνα από τους Μαθηματικούς Bernoulli, De Moivre, Laplace και Gauss. Ιδιαίτερα ο Laplace δεν περιορίζεται μόνο στη μαθηματική ανάλυση των τυχερών παιχνιδιών, αλλά εφαρμόζει τα συμπεράσματά του και σε ένα πλήθος από επιστημονικά και πρακτικά προβλήματα. Αξιοσημείωτο επίσης αποτελεί η κανονική κατανομή Gauss που δεν αφορά μόνο την κατανομή των σφαλμάτων των αστρονομικών παρατηρήσεων αλλά και την κατανομή πολλών βιολογικών, κοινωνικών και φυσικών φαινομένων. Έτσι, στη διάρκεια του 19ου αιώνα γεννιούνται νέοι κλάδοι των εφαρμοσμένων μαθηματικών. Στις μέρες μας η Θεωρία των Πιθανοτήτων με τις εργασίες πολλών διάσημων μαθηματικών, όπως είναι οι Chebyshev, Markov, Von Mises, Kolmogorov κ.ά., έχει σημειώσει αλματώδη πρόοδο. Καινούργια θεωρητικά αποτελέσματα παρέχουν νέες δυνατότητες για τη χρησιμοποίηση των μεθόδων της Θεωρίας των Πιθανοτήτων. Είναι αξιοσημείωτο το γεγονός ότι οι εφαρμογές των Πιθανοτήτων αναφέρονται σε ένα ευρύτατο φάσμα επιστημών όπως η Φυσική, η Χημεία, η Γενετική, η Ψυχολογία, η Οικονομολογία, η Τηλεπικοινωνία, η Μετεωρολογία, η πολιτική κτλ.

Δειγματικός χώρος λέγεται το σύνολο όλων των πιθανών αποτελεσμάτων και συμβολίζεται ως $\Omega = \{\omega_1, \omega_2, \dots, \omega_k\}$. Έτσι λοιπόν αν θέλουμε να υπολογίσουμε για παράδειγμα το δειγματικό χώρο για τη ρίψη ενός ζαριού έχουμε $\Omega = \{1,2,3,4,5,6\}$. Ο δειγματικός χώρος Ω πραγματοποιείται πάντα διότι οποιοδήποτε και αν είναι το αποτέλεσμα θα είναι υποσύνολο του δειγματικού χώρου. Το ενδεχόμενο που δεν πραγματοποιείται συμβολίζεται με \emptyset και λέγεται αδύνατο ενδεχόμενο. Τέλος το πλήθος ενός ενδεχομένου A συμβολίζεται $N(A)$

Ενδεχόμενο είναι ένα υποσύνολο του δειγματικού χώρου για παράδειγμα κατά τη ρίψη ενός ζαριού το ενδεχόμενο A να φέρουμε αριθμό μεγαλύτερο του 3 είναι $A = \{4,5\}$.

Ως κλασσικός ορισμός πιθανότητας είναι το πλήθος των ευνοϊκών ενδεχομένων διαιρούμενο με το συνολικό πλήθος των ενδεχομένων

Αρά με βάση το παράδειγμα έχουμε

$$P(A) = N(A) / N(\Omega)$$

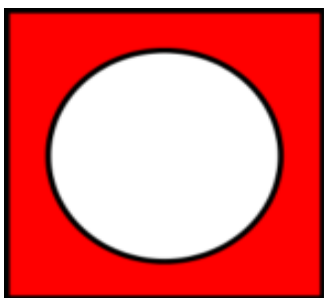
$$\Rightarrow P(A) = 2/6 \text{ ή } P(A) = 1/3$$

Η πιθανότητα είναι ένας αριθμός μεταξύ 0 και 1, το 1 που υποδηλώνει βεβαιότητα και 0 σημαίνει ότι το συμβάν δεν μπορεί να συμβεί. Επομένως, όσο μεγαλύτερη είναι η πιθανότητα ενός συμβάντος, τόσο πιο σίγουρο είναι ότι το συμβάν θα συμβεί. Στην πιο γενική περίπτωση, η πιθανότητα μπορεί να οριστεί αριθμητικά ως ο αριθμός των επιθυμητών αποτελεσμάτων που διαιρούνται με τον συνολικό αριθμό των αποτελεσμάτων.

4.3 Πράξεις με ενδεχόμενα

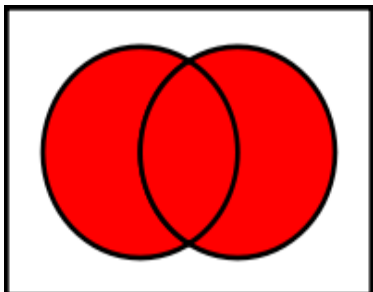
4.3.1 Συμπληρωματικό ενδεχόμενο A'

Το ενδεχόμενο συμβολίζεται με A' και είναι το ενδεχόμενο συμβαίνει όταν δεν πραγματοποιείται το A , αν για παράδειγμα κατά τη ρίψη ενός ζαριού έχουμε το ενδεχόμενο να φέρουμε 4,5, ή 6 $A = \{4,5,6\}$ τότε το ενδεχόμενο να μη πραγματοποιηθεί το ενδεχόμενο αυτό είναι $A' = \{1,2,3\}$



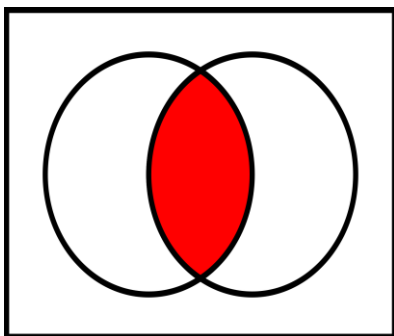
4.3.2 A ένωση B, $A \cup B$

A ένωση B. Το ενδεχόμενο συμβολίζεται με $A \cup B$, και είναι το ενδεχόμενο όταν πραγματοποιείτε ένα τουλάχιστον από τα A, B, αν $A = \{4,5,6\}$, $B = \{1\}$ τότε $A \cup B = \{1,4,5,6\}$



4.3.3 A τομή B, $A \cap B$

A τομή B. Το ενδεχόμενο συμβολίζεται με $A \cap B$ και είναι το ενδεχόμενο όταν πραγματοποιούνται και τα δυο ενδεχόμενα την ίδια στιγμή, αν $A = \{4,5,6\}$, $B = \{3,4\}$ τότε $A \cap B = \{4\}$



4.4 Ασυμβίβαστα ή Ξένα ενδεχόμενα μεταξύ τους

Δυο ενδεχόμενα λέγονται ασυμβίβαστα ή ξένα μεταξύ τους όταν δεν μπορούν να πραγματοποιηθούν ταυτόχρονα. Ένα παράδειγμα είναι κατά τη ρίψη ενός νομίσματος σπού το ενδεχόμενο να φέρω συγχρόνως κορόνα και γράμματα είναι αδύνατο, επομένως τα ενδεχόμενα αυτά είναι ξένα μεταξύ τους, συνεπώς $A \cap B = \emptyset$

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/>
- [2] https://www.tutorialspoint.com/mvc_framework/index.htm
- [3] <https://en.wikipedia.org>
- [4] https://www.w3schools.com/asp/webpages_chart.asp
- [4] <https://stackoverflow.com/>
- [5] <https://www.c-sharpcorner.com/>
- [6] <http://www.entityframeworktutorial.net/querying-entity-graph-in-entity-framework.aspx>
- [7] <https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/database-first-development/changing-the-database>
- [8] <https://dotnet-helpers.com/mvc/retrieve-views-from-different-folders-in-mvc/>
- [9] <http://ebooks.edu.gr/modules/ebook/show.php/DSGL-C100/493/3203,13010/>
- [10] <https://www.mathsisfun.com/data/probability.html>
- [11] <https://www.sololearn.com/>
- [12] <https://codepen.io/bsngr/pen/frDqh>
- [13] <http://www.dotnetawesome.com/2017/04/complete-login-registration-system-asp-mvc.html>
- [14] <https://www.w3schools.com/asp/default.asp>
- [15] <https://www.javatpoint.com/asp-net-tutorial>
- [16] <http://www.tutorialsteacher.com/linq/linq-tutorials>
- [17] <https://docs.microsoft.com/en-us/dotnet/csharp/linq/>
- [18] <http://www.entityframeworktutorial.net/Querying-with-EDM.aspx>
- [19] <https://www.aspsnippets.com/Articles/ASPNet-MVC-Database-Connection-Tutorial-with-example.aspx>

[20]<https://docs.microsoft.com/en-us/aspnet/mvc/overview/getting-started/introduction/creating-a-connection-string>

[21]<https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/index>

[22]<https://www.sololearn.com/Course/CSharp/>