



# Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

## Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Εκπαιδευτική Εφαρμογή Ιστορίας Μαθηματικών</b> <b>Educational Application Regarding Mathematics History</b>
Όνοματεπώνυμο Φοιτητή	<b>Χαλκιάς Ευστάθιος</b>
Πατρώνυμο	<b>Νικόλαος</b>
Αριθμός Μητρώου	<b>ΜΠΠΛ18069</b>
Επιβλέπων	<b>Βίρβου Μαρία, Καθηγήτρια</b>

Ημερομηνία Παράδοσης **Ιούνιος 2022**

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Μαρία Βίββου  
Καθηγήτρια

Ευάγγελος  
Σακκόπουλος  
Επίκουρος Καθηγητής

Κωνσταντίνα  
Χρυσafiάδη  
Ε.Δι.Π. Τμήματος  
Πληροφορικής

## Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή.....	5
1.1 Αντικείμενο πτυχιακής εργασίας (ΠΕ).....	5
1.2 Περιγραφή του προβλήματος .....	5
1.3 Προσδοκόμενα οφέλη από την ενσωμάτωση στο εκπαιδευτικό περιβάλλον .....	6
Κεφάλαιο 2: Σχεδιασμός λογισμικού .....	9
2.1 Σύντομη περιγραφή του λογισμικού που θα αναπτυχθεί .....	9
2.2.1 Ανάλυση απαιτήσεων .....	9
2.2.2 Μορφή ερωτήσεων .....	10
2.2.3 Μορφή ύλης .....	10
2.3 Αρχιτεκτονική των δεδομένων της εφαρμογής .....	11
2.4 Ακτινογραφία της βάσης δεδομένων.....	11
2.5 Διαγράμματα UML.....	16
Κεφάλαιο 3: Τεχνική τεκμηρίωση εφαρμογής σε Visual Studio .....	20
3.1 Δομή project.....	20
3.2 Σταθερές κατά μήκος της εφαρμογής (Constants.cs) .....	20
3.3 Επικοινωνία με τη βάση δεδομένων (DB.cs).....	20
3.4 Φόρμα πρόσβασης (SignForm.cs) .....	29
3.5 Κεντρική φόρμα (IntroForm.cs) .....	30
3.6 Φόρμα εξέτασης (TestForm.cs) .....	37
3.7 Φόρμα αποτελεσμάτων (ResultForm.cs) .....	45
3.8 Δημιουργία εγκατάστασης και φιλοξενία βάσης δεδομένων .....	46
Κεφάλαιο 4: Στιγμιότυπα εκτέλεσης εφαρμογής.....	48
Κεφάλαιο 5: Εγχειρίδιο χρήστη της εφαρμογής «Ιστορία των Μαθηματικών» .....	52
Μελέτη απο αρχείο τύπου «έγγραφο» (pdf) .....	62
Μελέτη απο αρχείο τύπου «Διαδικτυακό.....	63
Βίντεο».....	63



# Κεφάλαιο 1: Εισαγωγή

## 1.1 Αντικείμενο πτυχιακής εργασίας (ΠΕ)

Με την παρούσα ΠΕ πραγματοποιείται η υλοποίηση μιας εκπαιδευτικής εφαρμογής στο μάθημα της Ιστορίας των Μαθηματικών και με την ανάλυση τόσο της εφαρμογής όσο και της σημαντικότητας της διδασκαλίας του εν λόγω μαθήματος γίνεται φανερός ο λόγος συγγραφής της ΠΕ που έχει ως σκοπό την αξιοποίηση της εφαρμογής στην εκπαιδευτική διαδικασία. Το πρόγραμμα σε υπολογιστή, το οποίο κατά τη συγγραφή της εργασίας έχει ολοκληρωθεί είναι ένας πρακτικός και άμεσος τρόπος για την ενσωμάτωση του μαθήματος στην εκπαίδευση και η τεχνική τεκμηρίωση από την πλευρά του σπουδαστή αλλά και η χρήση του από το μαθητή θα λάβει το πρωτεύοντα χώρο στο παρόν έγγραφο.

## 1.2 Περιγραφή του προβλήματος

Με βάση την υφιστάμενη βιβλιογραφική έρευνα για την ιστορία των μαθηματικών αναφορικά με την εφαρμογή της στην εκπαιδευτική διαδικασία, υπάρχουν τόσο φιλοσοφικού όσο και πρακτικού/διδακτικού χαρακτήρα λόγοι που αποτρέπουν από αυτή να πραγματοποιηθεί. Σχετικά με τη φύση των μαθηματικών, η Ιστορία δεν είναι Μαθηματικά. Αν πρέπει να διδαχθεί η ιστορία, τότε θα πρέπει πρώτα να διδαχθούν τα ίδια τα μαθηματικά: δηλαδή πρώτα πρέπει να διδάσκεται το αντικείμενο, και μετά η ιστορία του. Ταυτόχρονα, η ιστορία μπορεί να είναι περίπλοκη και να προκαλεί σύγχυση στους μαθητές, παρά να τους βοηθήσει στην καλύτερη κατανόηση. Είναι ένα πρόβλημα εντονότερο όταν στατιστικά το μάθημα της ιστορίας είναι αντιπαθητικό και βαρετό προς τους διδασκόμενους.

Αναφορικά με τον πρακτικό χαρακτήρα του προβλήματος, εδώ δεν προκαλεί εντύπωση ότι ο χρόνος που διατίθεται για την εκμάθηση μαθηματικών στην τάξη είναι ήδη περιορισμένος, και μειώνεται ακόμα περισσότερο αν προστεθεί η διδασκαλία της ιστορίας τους. Η έλλειψη πόρων είναι ένα σημαντικότατο πρόβλημα με την έννοια ότι δεν υπάρχει αρκετό υλικό κατάλληλο για να βοηθήσει ακόμα και τους δασκάλους που θα ήθελαν να ενσωματώσουν ιστορικές πληροφορίες.

Επιπρόσθετα, δεν υπάρχει σαφής ή συνεπής τρόπος για την αξιολόγηση των μαθητών σε οποιαδήποτε ιστορική συνιστώσα, και αν δεν αξιολογείται τότε οι μαθητές δεν θα την εκτιμήσουν και δεν θα δώσουν προσοχή σε αυτήν. Η εισαγωγή μιας ιστορικής προσέγγισης στη διδασκαλία έχει εκ φύσεως ποιοτικό χαρακτήρα, και είναι δύσκολο να ενσωματωθεί σε τεστ ή εξετάσεις.

## 1.3 Προσδοκόμενα οφέλη από την ενσωμάτωση στο εκπαιδευτικό

### περιβάλλον

Τα οφέλη που προκύπτουν από τη λύση λογισμικού που προτείνεται χωρίζονται σε 2 είδη: Το πρώτο είδος αφορά τη θεωρητικό περιβάλλον που το συνοδεύει και την υπόδηλη αναγκαιότητα του μαθήματος και το δεύτερο είδος αφορά το τεχνικό πλαίσιο της λύσης. Αναφορικά με το θεωρητικό πλαίσιο του μαθήματος, μπορούμε να δηλώσουμε με μία πρόταση ότι η χρήση της ιστορίας «ως εργαλείο» περιλαμβάνει τα επιχειρήματα που έχουν σχέση με το πώς οι μαθητές μαθαίνουν μαθηματικά. Η ιστορία θα μπορούσε να λειτουργήσει ως ένας παράγοντας κινητοποίησης για την εκμάθηση και μελέτη των μαθηματικών, βοηθώντας, για παράδειγμα, να διατηρηθεί το ενδιαφέρον και ο ενθουσιασμός των μαθητών για το αντικείμενο. Μία ιστορική προσέγγιση θα μπορούσε να δώσει στα μαθηματικά ένα πιο ανθρώπινο πρόσωπο, και έτσι να τα κάνει λιγότερο τρομακτικά. Συχνά κομμάτια της μαθηματικής εξέλιξης, τα οποία έχουν ταλαιπωρήσει μαθηματικούς κατά το παρελθόν, τείνουν να ταλαιπωρούν και τους σημερινούς μαθητές, έτσι οι μαθητές θα μπορούσαν να πάρουν κουράγιο εάν γνωρίζουν ότι μία μαθηματική ιδέα που τους δυσκολεύει στην κατανόηση, είναι στην πραγματικότητα αποτέλεσμα δουλειάς εκατοντάδων χρόνων μέχρι να φτάσει στην τελική της μορφή. Επίσης, η ιστορία θα μπορούσε να έχει το ρόλο ενός γνωστικού εργαλείου υποστήριξης της εκμάθησης των μαθηματικών. Η ιστορία θα μπορούσε να βελτιώσει τη μάθηση και τη διδασκαλία των μαθηματικών, υποδεικνύοντας διαφορετικούς τρόπους παρουσίασης τους. Οι ιστορικές εμπειρικές μελέτες, θα μπορούσαν να βοηθήσουν στην ανάπτυξη μίας υποθετικής τροχιάς μάθησης, και θα μπορούσαν να βοηθήσουν τους εκπαιδευτικούς να δουν μέσα από τα μάτια των μαθητών. Μία ειδική χρήση της ιστορίας ως ένα γνωστικό εργαλείο, εμφανίζεται στην αναγνώριση επιστημολογικών εμποδίων, δηλαδή εμποδίων που κάποιος δεν μπορεί, ούτε θα πρέπει να αποφύγει, εξαιτίας του διαμορφωτικού τους ρόλου στην αναζήτηση γνώσης, και στον τρόπο που θα μπορούσαν αυτά να ξεπεραστούν. Τέλος, πολύ σημαντικά επιχειρήματα της κατηγορίας αυτής, είναι τα εξελικτικά επιχειρήματα, τα οποία υποστηρίζουν ότι δεν γίνεται να υπάρξει διδασκαλία μαθηματικών χωρίς τη χρήση ιστορίας, με πιο ξεκάθαρο το επιχείρημα της ανακεφαλαίωσης, το οποίο αναφέρει ότι «η οντογένεση ανακεφαλαιώνει τη φυλογένεση» (η οντογένεση αφορά την εξέλιξη ενός ατόμου, ενώ η φυλογένεση την εξέλιξη ενός είδους), δηλαδή για να μάθει κάποιος καλά μαθηματικά, πρέπει να ακολουθήσει τα ίδια βήματα που έχουν ακολουθήσει τα μαθηματικά κατά την εξέλιξή τους. Αυτό συχνά σχετίζεται και με ένα άλλο εξελικτικό επιχείρημα, αυτό του ιστορικού παραλληλισμού, το οποίο αφορά την παρατήρηση δυσκολιών και εμποδίων που εμφανίζονται στην ιστορία και επανεμφανίζονται στην. Με όλα τα παραπάνω, είναι εμφανές ότι η χρήση της ιστορίας «ως εργαλείο» σχετίζεται με ζητήματα που αφορούν το εσωτερικό των μαθηματικών.

Η εκμάθηση πτυχών της ιστορίας των μαθηματικών αποτελεί από μόνη της έναν σκοπό, χωρίς όμως αυτό να σημαίνει ότι η εκμάθηση της ιστορίας των μαθηματικών αντιμετωπίζεται ως ανεξάρτητο μάθημα, αλλά θα πρέπει να εστιάζει στις αναπτυξιακές και εξελικτικές πτυχές των

μαθηματικών ως αντικείμενο. Υπό αυτήν την έννοια, κάποιοι από τους στόχους της ιστορίας, είναι να δείξει στους μαθητές, ότι τα μαθηματικά υπάρχουν και εξελίσσονται στο χρόνο και στο χώρο, ότι είναι ένα επιστημονικό πεδίο που έχει εξελιχθεί και όχι κάτι που ήρθε από το πουθενά, ότι οι άνθρωποι έχουν πάρει μέρος σε αυτήν την εξέλιξη, ότι η εξέλιξη των μαθηματικών 19 οφείλεται σε πολλούς διαφορετικούς πολιτισμούς καθ' όλη τη διάρκεια της ιστορίας, ότι και η διαμόρφωση των μαθηματικών έχει επηρεαστεί από αυτούς τους πολιτισμούς, αλλά και οι πολιτισμοί έχουν επηρεαστεί από την εξέλιξη των μαθηματικών, καθώς και ότι η εξέλιξη οδηγείται τόσο από εσωτερικές όσο και από εξωτερικές δυνάμεις. Με όλα τα παραπάνω, είναι εμφανές ότι η χρήση της ιστορίας «ως στόχο» σχετίζεται με θέματα των μαθηματικών, όπως με την φύση, τον ρόλο και την σημασία τους, που θα μπορούσαν να ονομαστούν επιγραμματικά μετα-ζητήματα.

Συνεχίζοντας την επιχειρηματολογία για τα οφέλη της ενσωμάτωσης του μαθήματος στην πρακτική διάσταση, η εργασία αποσκοπεί στο να δώσει μία πρόταση στο τι θεωρεί αποτελεσματική λειτουργία μιας εκπαιδευτικής εφαρμογής στο μάθημα της ιστορίας των μαθηματικών αλλά και να επιτρέψει τον εύκολο χειρισμό από την πλευρά του μαθητή. Η αποτελεσματική διαχείριση και η ευχρηστία της εφαρμογής έχει άμεσα ευεργετικά αποτελέσματα για το μαθητή, καθώς ο τελευταίος επικεντρώνεται στο περιεχόμενο της εφαρμογής, δηλαδή στην ύλη του μαθήματος και τις ασκήσεις παρά στον τρόπο χρήσης της που πιθανόν να ήταν προβληματικός σε περίπλοκες και «ογκώδεις» εφαρμογές.

Για να προσκομιστεί όφελος στη διαδικασία, σημαντικός είναι ο σχεδιασμός ενοτήτων που ενδείκνυνται να διδαχθούν βάσει μίας ιστορικής προσέγγισης, και ανάλογη προσαρμογή των αναλυτικών προγραμμάτων. Προφανώς, μία διδακτική προσέγγιση βασισμένη στην ιστορία για όλη τη διδακτέα ύλη είναι ανέφικτη και περιττή, και αρκεί η εφαρμογή της σε ένα μικρό μέρος της ύλης για να προσελκύσει το ενδιαφέρον των μαθητών και να τους κινητοποιήσει.

Το λογισμικό που αναπτύχθηκε είναι ένα πρώτο βήμα προς την επιτυχή ενσωμάτωση στη μαθητική διαδικασία. Παρατίθενται οι τρόποι με τους οποίους θα μπορούσε να επιτευχθεί η ενσωμάτωση αυτή:

1. η εκμάθηση της ιστορίας, με την άμεση παροχή ιστορικών πληροφοριών,
2. η εκμάθηση μαθηματικών θεμάτων, ακολουθώντας μία προσέγγιση διδασκαλίας και μάθησης εμπνευσμένη από την ιστορία,
3. η ανάπτυξη βαθύτερης συνείδησης/επίγνωσης, τόσο για τα ίδια τα μαθηματικά, όσο και για το κοινωνικό και πολιτιστικό πλαίσιο στο οποίο υπάρχουν.

Εφόσον γίνεται σωστά η σχεδίαση της ύλης του μαθήματος, αυτό μπορεί να ενσωματωθεί σε μια εφαρμογή υπολογιστή και μάλιστα με διαδραστικό τρόπο (όπως βίντεο) με σκοπό την κινητοποίηση του διδασκόμενου. Η προβολή της προόδου του μαθητή σε πραγματικό χρόνο είναι ένα χαρακτηριστικό που μπορεί να τον βοηθήσει και να τον κάνει να καλύψει γρηγορότερα τα κενά του.

Τέλος, κατά τη διαδικασία υλοποίησης της εφαρμογής είναι η εκμάθηση και εξοικείωση από την πλευρά του σπουδαστή τις με γλώσσες προγραμματισμού που απαιτούνται για τη δημιουργία ολοκληρωμένων συστημάτων και διασύνδεση με περιφερειακές πλην σαφώς απαραίτητων τεχνολογιών που προορίζονται για χρήση μέσα από μια εφαρμογή παραθυρικού περιβάλλοντος που αλληλεπιδρά με το χρήστη.



## Κεφάλαιο 2: Σχεδιασμός λογισμικού

### 2.1 Σύντομη περιγραφή του λογισμικού που θα αναπτυχθεί

Η εφαρμογή θα είναι ένα πρόγραμμα παραθυρικού περιβάλλοντος (desktop application) και ο κώδικας που θα γραφεί θα είναι στη γλώσσα C# στο περιβάλλον Visual Studio 2022 κάνοντας χρήση του .NET framework 4.8 της Microsoft. Συγκεκριμένα θα δημιουργηθεί ένα νέο έργο (project) με το περιβάλλον διεπαφής που παρέχουν τα Windows Forms, που σημαίνει ότι η εφαρμογή θα εκτελείται σε συστήματα Windows.

Μέσω της εφαρμογής θα επιτρέπεται στο χρήστη (μαθητή) να ταυτοποιείται και να διαβάσει την ύλη του μαθήματος, η οποία διαιρείται σε ενότητες και υποδιαιρείται σε κεφάλαια. Η πλειοψηφία της ύλης υπάρχει σε έντυπη μορφή (PDF αρχεία και διαδικτυακοί HTML πόροι), ενώ υπάρχουν και βίντεο (προβολή μέσα από την υπηρεσία Youtube). Ο μαθητής μπορεί να προχωρήσει στην εξάσκηση στην ενότητα που θα επιλέξει έτσι ώστε να διαπιστώσει την ποσότητα και την ποιότητα των γνώσεων που απορρόφησε. Υπάρχει δυνατότητα ο μαθητής να προβάλλει τη σωστή απάντηση όπως και να πληροφορηθεί για το κεφάλαιο στο οποίο ανήκει η ερώτηση έτσι ώστε να ανατρέξει στο συγκεκριμένο σημείο στην ύλη για δεύτερη ανάγνωση. Όταν ο μαθητής ολοκληρώσει την εξάσκηση, τότε ξεκλειδώνει το διαγώνισμα για την εκάστοτε ενότητα και ο μαθητής μπορεί να το κάνει για να λάβει βαθμολογία. Σημειώνεται ότι οι ερωτήσεις για εξασκήσεις και διαγωνίσματα εμφανίζονται με τυχαίο αλγόριθμο, ούτως ώστε να διασφαλιστεί μια ακεραιότητα στη βαθμολόγηση του μαθητή αλλά και στην αφομοίωση των γνώσεων που έλαβε κατά την ανάγνωση της ύλης της ενότητας. Μετά την ολοκλήρωση ενός διαγωνίσματος ή εξάσκησης, ο μαθητής πληροφορείται αναλυτικά για τη βαθμολογία που έλαβε. Τέλος, εμφανίζονται στατιστικά στοιχεία για τη χρήση του προγράμματος και για τη συνολική βαθμολογία του μαθητή, τα οποία θα περιγραφούν σε επόμενες ενότητες.

#### 2.2.1 Ανάλυση απαιτήσεων

Παρακάτω παρατίθενται οι προδιαγραφές και η ανάλυση απαιτήσεων του προγράμματος που θα υλοποιηθεί:

1. Εγγραφή χρήστη
2. Είσοδος (ταυτοποίηση) χρήστη
3. Εμφάνιση κεφαλαίων και οργάνωσή τους σε ενότητες
4. Ανάγνωση της ύλης για κάθε κεφάλαιο
5. Δυνατότητα διενέργειας εξέτασης για συγκεκριμένη ενότητα
6. Δυνατότητα διενέργειας διαγωνίσματος όταν έχει ήδη πραγματοποιηθεί εξάσκηση για τη συγκεκριμένη ενότητας

7. Προβολή αναλυτικών αποτελεσμάτων κατά την ολοκλήρωση εξέτασης ή διαγωνίσματος
8. Προβολή στατιστικών στοιχείων (συνολικές αναγνώσεις κεφαλαίων, συνολικός αριθμός διενεργηθέντων εξασκήσεων και διαγωνισμάτων και τελευταίος βαθμός)

## 2.2.2 Μορφή ερωτήσεων

Οι ερωτήσεις που εμφανίζονται κατά τη διενέργεια εξετάσεων (και εξασκήσεων) είναι 4 ειδών:

1. Συμπλήρωση σωστής απάντησης. Εδώ παρουσιάζεται στο μαθητή μία πρόταση με ένα κενό σε ένα σημείο αυτής και καλείται να συμπληρώσει το κενό με μία λέξη για να θεωρείται η πρόταση σωστή.
2. Ερώτηση σωστού/λάθους. Παρουσιάζεται στο μαθητή μια πρόταση και καλείται να επιλέξει εάν αυτή είναι σωστή ή όχι πατώντας το αντίστοιχο κουμπί στην εφαρμογή.
3. Ερώτηση πολλαπλής επιλογής: Παρουσιάζεται στο μαθητή μια πρόταση ή μια ερώτηση και καλείται ο μαθητής να επιλέξει την απάντηση έτσι ώστε είτε να κάνει την πρόταση σωστή είτε να απαντήσει σωστά στην ερώτηση είτε να συμπληρώσει το κενό που έχει η πρόταση.
4. Ερώτηση αντιστοίχισης: Παρουσιάζονται 2 στήλες στο μαθητή και καλείται να αντιστοιχίσει τα στοιχεία της πρώτης στήλης (αριστερά) με τα στοιχεία της δεύτερης στήλης (δεξιά). Τα στοιχεία ανάμεσα στις στήλες ομοιάζουν σε σημασία ή συμπληρώνουν το ένα το άλλο και ο μαθητής επιλέγει ένα στοιχείο της στήλης A και ένα στοιχείο της στήλης B για να τα συνδέσει μεταξύ τους.

## 2.2.3 Μορφή ύλης

Η ύλη που αναγιγνώσκει ο μαθητής ανήκει σε 3 είδη:

1. Τοπικό αρχείο PDF: Όταν επιλέξει ο μαθητής ένα κεφάλαιο, του οποίου η ύλη έχει ένα αρχείο PDF, τότε αυτό εμφανίζεται (ανοίγει) ενσωματωμένο στην εφαρμογή.
2. Διαδικτυακή σελίδα: Εδώ απαιτείται η ενεργή σύνδεση στο Διαδίκτυο στη συσκευή του μαθητή για να μπορέσει να προβληθεί online περιεχόμενο που προέρχεται από ιστοσελίδα ενσωματωμένο στην εφαρμογή. Χρησιμοποιήθηκαν διαδικτυακοί πόροι από την ιστοσελίδα της Wikipedia.
3. Διαδικτυακό βίντεο στο YouTube: Απαιτείται και σε αυτή την περίπτωση ενεργή σύνδεση στο Internet καθώς το YouTube βίντεο αναπαράγεται ενσωματωμένο στην εφαρμογή.

## 2.3 Αρχιτεκτονική των δεδομένων της εφαρμογής

Το σύστημα αποθήκευσης δεδομένων της εφαρμογής θα βασίζεται στο σχεσιακό σύστημα βάσεων δεδομένων κάνοντας χρήση της τεχνολογίας MySQL. Η χρήση βάσης δεδομένων κρίθηκε απαραίτητη για την εν λόγω εφαρμογή καθώς ένα σχεσιακό σύστημα (RDMS) επιτρέπει την καλύτερη και αποδοτικότερη διαχείριση των δεδομένων σε σχέση με τη χρήση επίπεδων απλών αρχείων (flat files). Μέσω αυτού του σχεσιακού συστήματος τα δεδομένα είναι εύκολα επεξεργάσιμα, ταχύτερα επιλέξιμα με ταυτόχρονη κατανάλωση των ελάχιστων δυνατών πόρων της μηχανής ενώ παράλληλα συγκρατείται η ακεραιότητά τους και αποφεύγεται ο κατακερματισμός τους που θα γινόταν άλλοτε με την αποθήκευση επίπεδων αρχείων. Η χρήση ενός RDMS είναι μάλιστα αναγκαία σήμερα καθώς επιτρέπει την ανέξοδη και ταχεία μετάβαση των δεδομένων της εφαρμογής σε διακομιστή Ιστού (server) από το τοπικό μηχάνημα καθώς υπάρχει μεγάλη τάση που θέλει τις εφαρμογές να είναι στο cloud. Επομένως, έχει προβλεφθεί η ενημέρωση και προβολή των δεδομένων της εφαρμογής απομακρυσμένα καθώς η τοπική βάση δεν εξυπηρετεί τους σκοπούς μιας εκπαιδευτικής εφαρμογής, η οποία εάν λάβει ευρεία αποδοχή θα χρειαστεί οργάνωση της διδασκαλίας και συνεπώς ενημέρωση των δεδομένων που κάνει χρήση η εφαρμογή.

## 2.4 Ακτινογραφία της βάσης δεδομένων

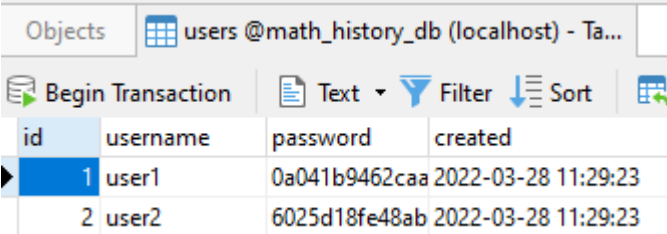
Το πιο σημαντικό συστατικό μιας εφαρμογής δεν είναι τόσο η εφαρμογή αυτή καθαυτή, αλλά η βάση δεδομένων, καθώς αυτή αποτελεί την περιουσία του συστήματος αφού τα δεδομένα των χρηστών είναι πιο σημαντικά από τον κώδικα και πρέπει να υπάρχει εγγύηση ότι αυτά δε θα κατακερματιστούν ούτε να αλλοιωθούν ενώ παράλληλα πρέπει να υπάρχει διασφάλιση ότι τα δεδομένα είναι πραγματικά, ακέραια και ότι αποθηκεύονται με το βέλτιστο τρόπο με την ελάχιστη δυνατή χρήση πόρων χωρίς να επηρεάζουν αρνητικά τη λειτουργία και την απόδοση της εφαρμογής. Δημιουργήθηκαν 5 πίνακες και αναλύεται η δομή τους παρακάτω:

### Πίνακας χρηστών (users):

Ο εν λόγω πίνακας αποθηκεύει τους χρήστες που θα χρησιμοποιούν την εφαρμογή. Αποτελείται από τη στήλη id, η οποία είναι το αναγνωριστικό κάθε εγγραφής ή με απλά λόγια είναι η ταυτότητα της κάθε γραμμής σε αυτόν τον πίνακα όπως και τους υπόλοιπους που θα δούμε. Είναι πρωτεύον κλειδί, ακέραιος 11 ψηφίων αύξουσας αρίθμησης, που σημαίνει ότι κάθε νέα εγγραφή που εισάγεται στον πίνακα λαμβάνει αυτόματα (από τη MySQL) έναν αριθμό με βάση την τιμή id της προηγούμενης εγγραφής και αυξημένη κατά 1 μονάδα. Η στήλη username είναι τύπου varchar 255 χαρακτήρων, δηλαδή συμβολοσειρά με μέγιστο μήκος 255 χαρακτήρες. Παρομοίως και η στήλη password και αποτελούν μαζί τη βάση για την ταυτοποίηση των χρηστών, δηλαδή απαιτείται από το χρήστη να παράσχει σωστά και τα 2 πεδία για την επιτυχή αυθεντικοποίησή του. Η στήλη

created είναι τύπου χρονοσφραγίδας (timestamp) και συμπληρώνεται αυτομάτως από τη MySQL σε κάθε νέα εγγραφή και έτσι γνωρίζουμε την ημερομηνία και ώρα που εγγράφηκε ο χρήστης στην εφαρμογή.

## Εγγραφές



id	username	password	created
1	user1	0a041b9462caa	2022-03-28 11:29:23
2	user2	6025d18fe48ab	2022-03-28 11:29:23

Εικόνα 1 Εγγραφές πίνακα users

## Πίνακας κεφαλαίων (chapters):

Εδώ θα βρούμε και πάλι τη στήλη id και έχει την ίδια ιδιότητα με την ομώνυμη στήλη στον πίνακα users. Η στήλη section μάς πληροφορεί για την ενότητα που ανήκει το κεφάλαιο και είναι τύπου varchar (συμβολοσειρά). Η στήλη title αφορά τον τίτλο του κεφαλαίου και είναι τύπος text, δηλαδή συμβολοσειρά μεγάλου μήκους, εντός της οποίας μπορούμε να αποθηκεύσουμε αρκετά megabytes κειμένου. Παρομοίως και η στήλη description, η οποία αποτελεί μια περιγραφή του κεφαλαίου. Η στήλη type είναι συμβολοσειρά και αποτελεί τον τύπο του κεφαλαίου, δηλαδή το είδος του πόρου (ή αρχείου) που διαβάζει ο μαθητής. Έχει προαναφερθεί ότι έχουμε 3 τύπους ύλης και έχουν τις κωδικές ονομασίες: pdf, online\_page, online\_video. Η στήλη file είναι τύπος text και αποθηκεύει είτε το όνομα του αρχείου PDF (εάν πρόκειται για τοπικό αρχείο) είτε τη διεύθυνση της ιστοσελίδας (URL) εάν πρόκειται για διαδικτυακό πόρο εάν πρόκειται για ιστοσελίδα ή βίντεο.

## Εγγραφές

id	section	title	description	type	file
1	1	Εισαγωγή στην ιστορία των μαθηματικών	(Null)	pdf	Presentation22_02_12.pdf
2	1	Τα Μαθηματικά των αρχαίων Αιγυπτίων, συστήματα αριθμησης	(Null)	pdf	Presentation29_02_12.pdf
3	1	Τα Μαθηματικών των Βαβυλωνίων	(Null)	pdf	Presentation1_03_12.pdf
4	1	Τι είναι απόδειξη?	(Null)	pdf	Presentation7_03_12.pdf
5	2	Θαλής, Πυθαγόρας, Πλατωνικά Στερεά, άρρητα μεγέθη, παράδοξα του Ζήνωνα,	(Null)	pdf	Presentation8_03_12.pdf
6	2	Εύδοξος	(Null)	pdf	Presentation14_03_12.pdf
7	2	Τα Στοιχεία του Ευκλείδη, το πέμπτο αίτημα και οι μη Ευκλείδειες γεωμετρίες	(Null)	pdf	Presentation15_03_12.pdf
8	2	Αρχιμήδης: Ψαμμίτης, το βοεκό πρόβλημα και το εμβαδόν ανάμεσα στη παραβ	(Null)	pdf	Presentation21_03_12.pdf
9	2	Αρχιμήδης: Η μέθοδος: κύβος και κύλινδρος, Διόφαντος	(Null)	pdf	Presentation22_03_12.pdf
10	3	Τι είναι Άλγεβρα? Δευτεροβάθμια εξίσωση από το 2000 π.Χ. -250 μ.Χ.	(Null)	pdf	Presentation28_03_12.pdf
11	3	Αραβικά μαθηματικά. Liber abaci. Ιταλοί και τριτοβάθμια εξίσωση	(Null)	pdf	Presentation29_03_12.pdf
12	3	Cardano και λύση της τριτοβάθμιας	(Null)	pdf	Presentation4_04_12.pdf
13	3	Τεταρτοβάθμια, Bombelli, Viete: οι μαγικοί αριθμοί είναι χρήσιμοι	(Null)	pdf	Presentation5_04_12.pdf
14	3	Ο καιρός των λογαρίθμων	(Null)	pdf	Presentation25_04_12.pdf
15	4	Σκέφτομαι άρα υπάρχει...Μέθοδοι παραγωγής και τετραγωνισμού: Descartes,	(Null)	pdf	Presentation26_04_12.pdf
16	4	Στο δρόμο για το Θεμελιώδες Θεώρημα του Λογισμού: Newton και το Θεώρημα	(Null)	pdf	Presentation02_05_12.pdf
17	4	Newton και το Θεμελιώδες Θεώρημα του Λογισμού, Leibniz	(Null)	pdf	Presentation03_05_12.pdf
18	4	Σύγκριση των Λογισμών του Newton και Leibniz, Κάποια βασικά ερωτήματα στ	(Null)	pdf	Presentation09_05_12.pdf
19	5	Το Θεμελιώδες Θεώρημα της Άλγεβρας και απαντήσεις σε κάποια βασικά ερωτή	(Null)	pdf	Presentation10_05_12.pdf
20	5	Peacock και συμβολική Άλγεβρα, Euler, Gauss. Στο δρόμο για την Αφηρημένη/Mc	(Null)	pdf	Presentation17_05_12.pdf
21	5	Μία σύντομη Εισαγωγή στην Ιστορία της Πιθανότητας (διάλεξη: Καθηγ. Π. Μωυ	(Null)	pdf	Presentation30_03_12.pdf
22	5	Αφηρημένη Άλγεβρα: Θεωρία Ομάδων στο έργο των Gauss, Galois, Μη αντιμετ	(Null)	pdf	Presentation24_05_12.pdf
1000	6	Ερατοσθένης ο Κυρηναίος	(Null)	online_page	<a href="https://el.wikipedia.org/wiki/">https://el.wikipedia.org/wiki/</a>
1007	6	Πυθαγόρας	(Null)	online_page	<a href="https://el.wikipedia.org/wiki/">https://el.wikipedia.org/wiki/</a>
1008	6	Το πείραμα του Ερατοσθένη	(Null)	online_video	<a href="https://www.youtube.com/e">https://www.youtube.com/e</a>
1009	7	Ευκλείδης	(Null)	online_page	<a href="https://el.wikipedia.org/wiki/">https://el.wikipedia.org/wiki/</a>
1010	7	Αρχιμήδης	(Null)	online_page	<a href="https://el.wikipedia.org/wiki/">https://el.wikipedia.org/wiki/</a>
1011	7	Παράδοξα του Ζήνωνα	(Null)	online_page	<a href="https://el.wikipedia.org/wiki/">https://el.wikipedia.org/wiki/</a>

Εικόνα 2 Εγγραφές πίνακα chapters

### Πίνακας ερωτήσεων (questions):

Εδώ θα βρούμε τη στήλη id. Επίσης θα βρούμε τη στήλη section που μας πληροφορεί για την ενότητα στην οποία ανήκει η ερώτηση και είναι τύπου συμβολοσειράς. Η στήλη chapter\_id είναι ακέραιος 11 ψηφίων και είναι νοητό ξένο κλειδί στον πίνακα των κεφαλαίων στη στήλη id. Η στήλη title είναι τύπος text και αποθηκεύει την ερώτηση καθαυτή. Η στήλη correct είναι τύπου text και εδώ αποθηκεύουμε τη σωστή ερώτηση ανάλογα με το είδος της ερώτησης. Για παράδειγμα: Για ερώτηση συμπλήρωσης κενού, η τιμή της εν λόγω στήλης είναι η λέξη που λείπει από την πρόταση της ερώτησης. Για ερώτηση σωστού/λάθους, η τιμή είναι 1 όταν η απάντηση είναι το Σωστό, ενώ η τιμή είναι 0 όταν η απάντηση είναι το Λάθος. Για ερώτηση πολλαπλής επιλογής, η τιμή είναι 1 όταν η σωστή επιλογή είναι η πρώτη από τις διαθέσιμες, η τιμή είναι 2 όταν η σωστή επιλογή είναι η δεύτερη, η τιμή είναι 3 όταν η σωστή επιλογή είναι η τρίτη και η τιμή είναι 4 όταν η σωστή επιλογή είναι η τέταρτη. Για ερώτηση αντιστοίχισης, η τιμή είναι η αντιστοίχιση στοιχείου της πρώτης στήλης με το στοιχείο της δεύτερης στήλης συνενωμένα με παύλα και χωρίζοντας τις αντιστοιχίσεις μεταξύ τους με κόμμα. Για παράδειγμα με την τιμή «1-a,2-c,3-d,4-b» αντιστοιχίζουμε το στοιχείο a της 1<sup>ης</sup> στήλης με το στοιχείο a της 2<sup>ης</sup> στήλης, το στοιχείο 2 από την 1<sup>η</sup> στήλη με το στοιχείο c της 2<sup>ης</sup> στήλης κ.ο.κ. Τέλος, η στήλη type αναφέρεται στο είδος της ερώτησης. Για ερώτηση συμπλήρωσης κενού έχουμε την τιμή fill\_empty, για ερώτηση σωστού/λάθους έχουμε την τιμή correct\_wrong, για ερώτηση πολλαπλής επιλογής έχουμε την τιμή choice και για ερώτηση αντιστοίχισης έχουμε την τιμή map.

id	section	chapter_id	title	correct	type
200	6	1000	Το κόσκινο του Ερατοσθένη είναι ένας τρόπος υπολογισμού των _ αριθμών	πρώτων	fill
201	6	1000	Ο Ερατοσθένης πληροφορήθηκε ότι στη Σύνη (σημερινό Ασουάν) ο ήλιος κατά το μεσημέρι του θερινού κάθεται		fill
211	6	1000	Ένας πρώτος αριθμός είναι ένας φυσικός αριθμός που έχει ακριβώς δύο διαφορετικούς διαιρέτες: το 1 και το	εαυτό	fill
212	6	1000	Η μελέτη των παραστατικών αριθμών οδήγησε τους Πυθαγορείους στην ανακάλυψη της μεθόδου για τριάδων		fill
213	6	1000	Ο αριθμός 25 παριστάνει ένα _	τετράγωνο	fill
214	6	1000	Σύμφωνα με το Πυθαγόρειο Θεώρημα: το τετράγωνο της υποτινούσης (της πλευράς που βρίσκεται α	άθροισμα	fill
215	6	1000	Ισχύει και το αντίστροφο Πυθαγόρειο Θεώρημα: ότι δηλαδή, αν ισχύει η παραπάνω σχέση μεταξύ των ορθογώνιο		fill
216	6	1000	Ο Ερατοσθένης Σπούδασε στην _ και ισχυριζόταν ότι επίσης σπούδασε για κάποια χρόνια στην Αθήνα. Αλεξάνδρεια		fill
300	6	1000	Ο Ερατοσθένης ο Κυρηναίος γεννήθηκε τον 5ο αιώνα π.Χ.	0	co
301	6	1000	Ένα από τα πιο σημαντικά πειράματα που πραγματοποιήθηκε στην ιστορία της ανθρωπότητας ήταν ι	0	co
302	6	1000	Ο Ερατοσθένης εφηύρε τον σφαιρικό αστρολάβο	1	co
303	6	1000	Ο όρος Γεωγραφία αποδίδεται στον Ερατοσθένη	1	co
304	6	1000	Ο Πυθαγόρας έζησε τον 2ο αιώνα π.Χ.	0	co
305	6	1000	Κοντά στον Θαλή ο Πυθαγόρας έλαβε την πρώτη του σοβαρή εκπαίδευση πάνω στα μαθηματικά, τη γ	1	co
306	6	1000	Στην σχολή του ο Πυθαγόρας δεν διδασκε γυναίκες	0	co
307	6	1000	Οι Πυθαγόρειοι φιλόσοφοι είναι μια φιλοσοφική, θρησκευτική και πολιτική σχολή που ιδρύθηκε τον 6ο	1	co
308	6	1000	Η διδασκαλία γινόταν με προφορικό τρόπο και οι προϋποθέσεις για την είσοδο των μαθητών ήταν αυ	1	co
309	6	1000	Οι Πυθαγόρειοι απέδωσαν στην Αριθμητική μέγιστη σημασία, μελετώντας τις ιδιότητές της.	1	co
310	6	1000	Η ανακάλυψη της ασυμμετρίας είναι, σύμφωνα με τον έγκυρο σχολιαστή Πάππο από την Αλεξάνδρεια, 1		co

Εικόνα 3 Εγγραφές πίνακα questions

### Πίνακας αναγνώσεων (read\_actions):

Σε αυτόν τον πίνακα αποθηκεύουμε τις αναγνώσεις του κεφαλαίου εκ μέρους του μαθητή. Έχουμε φυσικά τη στήλη id και τη στήλη user\_id που είναι ακέραιος 11 ψηφίων αποτελώντας νοητό ξένο κλειδί στον πίνακα users επί της στήλης id, δηλαδή η εγγραφή αφορά έναν χρήστη. Η στήλη chapter\_id είναι νοητό ξένο κλειδί στον πίνακα chapter επί της στήλης id. Η στήλη created\_at είναι χρονοσφραγίδα (με αυτόματη συμπλήρωση) και αφορά το πότε η εγγραφή εισήχθη στον πίνακα και η στήλη completed\_at είναι επίσης χρονοσφραγίδα και μας πληροφορεί το πότε ο μαθητής ολοκλήρωσε την ανάγνωση του κεφαλαίου.

### Ενδεικτικές εγγραφές

id	user_id	chapter_id	created_at	completed_at
53	1	1	2022-03-14 13:5	2022-03-14 13:55:55
54	1	2	2022-03-14 13:5	2022-03-14 13:55:59
55	1	1	2022-03-15 01:4	2022-03-15 01:44:40
56	1	1	2022-03-15 01:4	(Null)
57	1	1	2022-03-15 01:5	(Null)
58	1	1	2022-03-15 01:5	2022-03-15 01:54:16
59	1	1007	2022-03-15 01:5	(Null)
60	1	1	2022-03-15 01:5	(Null)
61	1	12	2022-03-15 01:5	(Null)
62	1	1	2022-03-15 01:5	2022-03-15 01:58:01
63	1	8	2022-03-15 01:5	2022-03-15 01:58:44
64	1	6	2022-03-15 01:5	2022-03-15 01:58:50
65	1	4	2022-03-15 01:5	2022-03-15 01:59:47
66	1	1	2022-03-17 00:2	2022-03-17 00:29:25
67	1	1	2022-03-17 00:3	2022-03-17 00:30:34
68	1	1	2022-03-17 00:3	2022-03-17 00:31:13
69	1	2	2022-03-17 00:3	2022-03-17 00:31:27
70	1	1	2022-03-17 00:3	2022-03-17 00:32:35

Εικόνα 4 Εγγραφές πίνακα read\_actions

### Πίνακας εξασκήσεων/διαγωνισμάτων (tests):

Εδώ αποθηκεύουμε τις διενεργηθέντες εξετάσεις. Έχουμε τη στήλη id και τη στήλη user\_id το οποίο είναι νοητό ξένο κλειδί στον πίνακα users. Η στήλη mode είναι τύπου tinyint (ή αλλιώς Boolean) και έχουμε την τιμή 0 όταν η εξέταση είναι εξάσκηση, ενώ η τιμή 1 αφορά ένα διαγώνισμα. Η στήλη section\_id είναι η ενότητα της εξέτασης και είναι ακέραιος 11 ψηφίων. Η στήλη total\_questions είναι ακέραιος και αφορά τις συνολικές ερωτήσεις που έχει μια εξέταση. Η στήλη questions\_answered είναι ακέραιος και είναι οι απαντηθείσες ερωτήσεις εκ μέρους του μαθητή. Η στήλη correct\_questions είναι ακέραιος και αποθηκεύει τις συνολικές σωστές απαντήσεις που έδωσε ο μαθητής. Η στήλη result είναι συμβολοσειρά (varchar 255 χαρακτήρων) και είναι ο δοθείσα απάντηση του μαθητή. Η στήλη created\_at είναι χρονοσφραγίδα και αποθηκεύει την ημερομηνία και ώρα που εκκίνησε η εξέταση. Τέλος η στήλη completed\_at είναι επίσης τύπου χρονοσφραγίδας και αποθηκεύει την ημερομηνία και ώρα που ολοκληρώθηκε η εξέταση. Σημειώνεται ότι μια εξέταση δεν ολοκληρώνεται πάντα με την έννοια ότι ο χρήστης ενδέχεται να μη φθάσει μέχρι το τέλος παραλείποντας τις εναπομείναντες ερωτήσεις. Σε αυτήν την περίπτωση, η εν λόγω στήλη δεν συμπληρώνεται.

### Εγγραφές

id	user_id	mode	section_id	total_questions	questions_answered	correct_questions	result	created_at	completed_at
132	1	0	1	30	30	9	fail	2022-04-15 11:32:11	2022-04-15

Παρακάτω βλέπουμε το διάγραμμα οντοτήτων συσχετίσεων της βάσης:

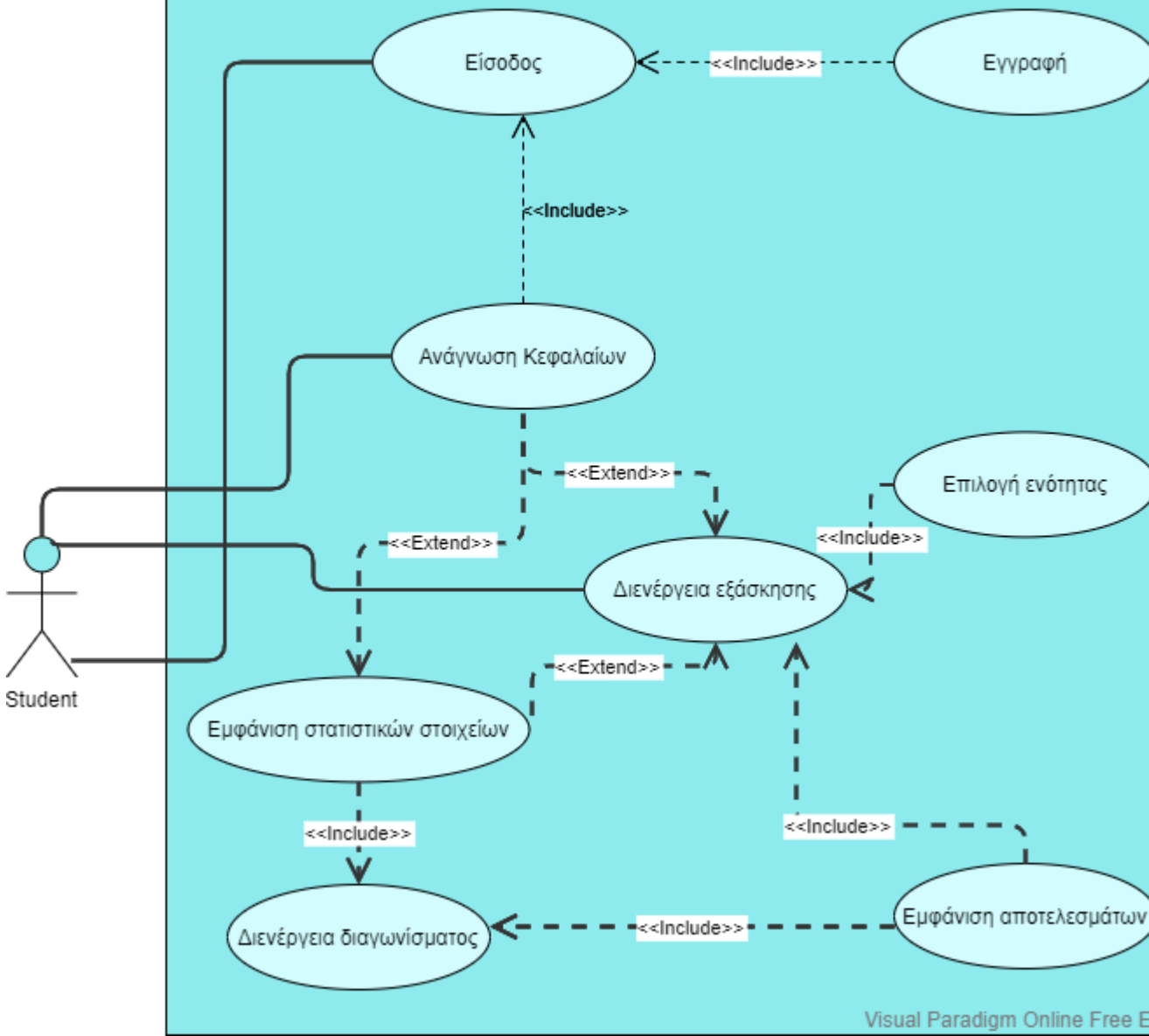


Εικόνα 6 Σχήμα της βάσης δεδομένων

## 2.5 Διαγράμματα UML

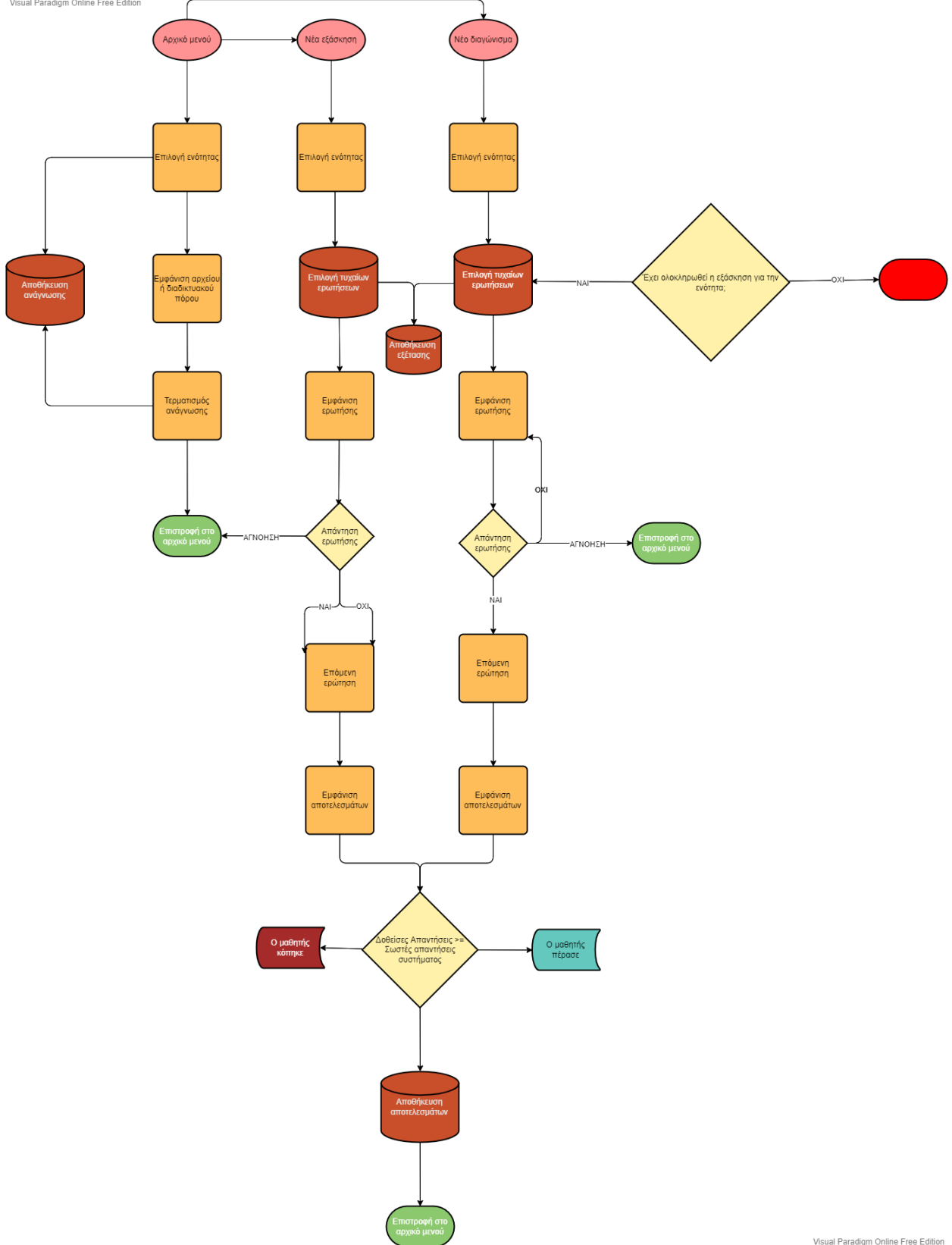
Παρακάτω θα δούμε μερικά διαγράμματα UML για να μας βοηθήσουν να κατανοήσουμε την εφαρμογή που θα αναπτυχθεί. Το πρώτο διάγραμμα είναι ένα use case diagram και περιγράφει γενικά τις κύριες λειτουργίες του συστήματος. Παρατηρούμε ότι το σημείο έναρξης όλων των λειτουργιών είναι η είσοδος του χρήστη όπως επίσης η διενέργεια εξάσκησης είναι απαραίτητη για τη διενέργεια διαγωνίσματος για την εκάστοτε ενότητα.





Εικόνα 7 Use Case

Το παρακάτω διάγραμμα ροής (flowchart) μας βοηθά να καταλάβουμε καλύτερα την εφαρμογή μέσα από μεμονωμένα βήματα:



Εικόνα 8 Flowchart

Εδώ τα σημαντικά στοιχεία είναι η συνεχής επικοινωνία με τη βάση δεδομένων σε κάθε ενέργεια του χρήστη. Επίσης παρατηρούμε ότι ο μαθητής δεν απαιτείται να ολοκληρώσει την εξέταση και μπορεί να επιστρέψει στο αρχικό μενού ανά πάσα στιγμή. Επιπλέον ο μαθητής υποχρεούται να απαντήσει στην ερώτηση όταν κάνει ένα διαγώνισμα για να συνεχίσει στην επόμενη ερώτηση σε αντίθεση με την εξάσκηση όπου αυτό δεν είναι απαραίτητο. Τέλος, ο μαθητής οφείλει να δώσει σωστές απαντήσεις ίσες ή περισσότερες από το 50% των συνολικών εμφανιζόμενων ερωτήσεων για να περάσει την εξέταση. Για παράδειγμα, εάν παρουσιαστούν 20 ερωτήσεις στο μαθητή σε μια ενότητα, ο τελευταίος οφείλει να απαντήσει σωστά 10 ή περισσότερες ερωτήσεις. Σημειώνεται ότι

ο αριθμός των ερωτήσεων σε μια εξέταση είναι δυναμικός και οι εμφανιζόμενες ερωτήσεις επιλέγονται τυχαία από τη βάση και είναι ένας αλγόριθμος που θα περιγράψουμε στη συνέχεια.

## Κεφάλαιο 3: Τεχνική τεκμηρίωση εφαρμογής σε Visual Studio

### 3.1 Δομή project

Η εφαρμογή δημιουργήθηκε στο Visual Studio 2022 και το project είναι σε Windows Form App με τη χρήση της γλώσσας προγραμματισμού C#. Για τους σκοπούς της εφαρμογής δημιουργήθηκαν συνολικά 8 αρχεία. Μια εφαρμογή σε Visual Studio αποτελείται από πολύ περισσότερα αρχεία απαραίτητα για την εκτέλεση του προγράμματος αλλά δε θα αναλυθούν καθώς δεν γράφτηκαν εξ'ιδίαν.

### 3.2 Σταθερές κατά μήκος της εφαρμογής (Constants.cs)

Η σύνδεση με τη βάση δεδομένων είναι παρούσα σχεδόν σε όλα τα αρχεία της εφαρμογής. Γι' αυτό το λόγο δημιουργήσαμε το αρχείο έτσι ώστε να έχουμε σε ένα σημείο τις παρακάτω σταθερές:

```
public static string dbServer = "localhost"; public static string dbName =  
"math_history_db"; public static string dbUsername = "root"; public static string  
dbPassword = "";  
public static string appTitle = "Ιστορία των Μαθηματικών";
```

Ο server της βάσης είναι ο τοπικός (localhost). Η ονομασία της βάσης είναι math\_history\_db. Ο χρήστης που συνδέεται στη βάση είναι ο root, ενώ δεν απαιτείται κωδικός για τη σύνδεση σε αυτή. Η σταθερά appTitle έχει την ονομασία της εφαρμογής και χρησιμοποιείται σε διάφορα σημεία όπως τα πλαίσια διαλόγου και τις φόρμες.

### 3.3 Επικοινωνία με τη βάση δεδομένων (DB.cs)

Ορίσαμε μια κλάση που αναλαμβάνει την επικοινωνία με τη βάση και αυτή είναι η MySQL (αντικείμενο MySqlConnection). Σημειώνεται ότι ο οδηγός MySQL δεν υπάρχει εγγενώς στο Visual Studio, αλλά υπάρχει ως πακέτο στη διαχείριση πακέτων (NuGet), δηλαδή πρόκειται για τρίτου-μέρους (third-party) βιβλιοθήκη έχοντας την ονομασία MySql.Data. Έχουμε ορίσει getters και setters για τα στοιχεία της σύνδεσης με τη βάση όπως φαίνεται παρακάτω:

```
public string Server { get; set; }  
public string DatabaseName { get; set; }  
public string UserName { get; set; }  
public string Password { get; set; }
```

Απαραίτητες μέθοδοι για την επικοινωνία με τη βάση είναι οι κάτωθι:

```
public static DB Instance()  
{  
    if (_instance == null)  
        _instance = new DB();  
    return _instance;  
}
```

```

public bool IsConnect()
{
    if (Connection == null)
    {
        string connstring = string.Format("Server={0}; database={1}; UID={2};
password={3}", Server, DatabaseName, UserName, Password);
        Connection = new MySqlConnection(connstring);
        Connection.Open();
    }
    else if (Connection.State == System.Data.ConnectionState.Closed)
    {
        Connection.Open();
    }
    return true;
}

```

Η πρώτη μέθοδος είναι ο δημιουργός της κλάσης που απλά δημιουργεί νέο αντικείμενο DB (κάθε φορά που καλείται), ενώ η δεύτερη μέθοδος είναι αυτή που εκκινεί την επικοινωνία με τη βάση πριν κάθε αιτούμενη αλληλεπίδραση με αυτήν.

Στη συνέχεια συναντούμε τη μέθοδο που αναλαμβάνει να ελέγξει εάν υπάρχει ο χρήστης στη βάση:

```

public static bool UserExists(DB dbCon, string username)
{
    try
    {
        var sql = @"SELECT * FROM `users` WHERE `username` = @user";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@user", username);
        var db_read = cmd.ExecuteReader();
        bool b = db_read.HasRows ? true : false;
        db_read.Close();
        return b;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return true;
    }
}

```

Αυτή η μέθοδος (δεχόμενη ένα αντικείμενο DB και σε μια συμβολοσειρά το όνομα χρήστη) θα κάνει ερώτηση επιλογής στον πίνακα users με βάση το username του και χρησιμοποιείται κατά την προσπάθεια εγγραφής ενός νέου χρήστη. Θα επιστρέψει true (Boolean) εάν υπάρχει ο χρήστης με αυτό το username (το ερώτημα επιστρέφει εγγραφές), διαφορετικά επιστρέφει false. Σε κάθε περίπτωση θυμόμαστε να τερματίσουμε τη σύνδεση με τη βάση μετά το πέρας του ερωτήματος προς αυτήν. Όλες οι μέθοδοι βρίσκονται σε try/catch blocks για να μπορέσουμε να πληροφορούμαστε για το μήνυμα λάθους κατά την επικοινωνία με βάση εάν αυτό συμβεί.

Στη συνέχεια θα εντοπίσουμε τη μέθοδο εγγραφής χρήστη στον πίνακα users:

```

public static bool Register(DB dbCon, string username, string password)
{
    string hashedPassword = HashPassword(password);
    //Console.WriteLine(hashedPassword);
    try
    {
        var sql = @"INSERT INTO `users` (`username`, `password`) VALUES
(@user, @pass)";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@user", username);

```

```

        cmd.Parameters.AddWithValue("@pass", hashedPassword);
        cmd.ExecuteNonQuery();
        return true;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return false;
    }
}

```

Η μέθοδος δέχεται ένα αντικείμενο DB και το username και τον κωδικό του χρήστη και θα κάνει ερώτηση προσθήκης στον πίνακα users εισάγοντας τα 2 τελευταία ορίσματα της μεθόδου. Επιστρέφει true εάν η εγγραφής ήταν επιτυχής.

Με την παρακάτω μέθοδο επιχειρείται είσοδος από το χρήστη:

```

public static User Login(DB dbCon, string username, string password)
{
    User user = new User();
    string hashedPassword = HashPassword(password);
    try
    {
        var sql = @"SELECT * FROM `users` WHERE `username` = @user AND
`password` = @pass LIMIT 1";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@user", username);
        cmd.Parameters.AddWithValue("@pass", hashedPassword);
        cmd.ExecuteNonQuery();
        var db_read = cmd.ExecuteReader();
        while (db_read.Read()) {
            user = new User();
            user.setId(db_read.GetInt32(0));
            user.setUsername(db_read.GetString(1));
        }
        db_read.Close();
        return user;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}

```

Θα λάβει ως ορίσματα ένα αντικείμενο DB και το username και password του χρήστη. Εδώ να επισημάνουμε ότι ο κωδικός του χρήστη στη βάση δεν αποθηκεύεται στην αυτούσια μορφή (as is) αλλά αποθηκεύεται κατακερματισμένος για λόγους ασφαλείας για να προλάβουμε πιθανή επίθεση hacking όπου ένας κακόβουλος χρήστης αποκτήσει πρόσβαση στα δεδομένα της βάσης και στην προσπάθειά του να προσποιηθεί έναν χρήστη (εκτός του ίδιου), αυτή θα αποτύχει καθώς δεν υπάρχει τρόπος να γνωρίζει τον πραγματικό κωδικό του άλλου χρήστη, αφού αυτός είναι σε μορφή μη κατανοητή και κατακερματισμένη με ισχυρό αλγόριθμο.

Η μέθοδος θα κάνει ερώτηση επιλογής στον πίνακα users αιτούμενοι μόνο την πρώτη εγγραφή και εφόσον επιστρέψει αποτέλεσμα, μέσω ενός while βρόγχου, αυτό θα διαβαστεί και θα δημιουργήσει νέο αντικείμενο (instance) User, που θα δούμε αργότερα. Για το χρήστη χρειάζεται να γνωρίζουμε το id του και το username, γι' αυτό και θέτουμε με αντίστοιχες μεθόδους επί του αντικειμένου User

τα ζητούμενα στοιχεία του χρήστη. Τη στήλη id (από τη βάση) τη μετατρέπουμε σε ακέραιο, ενώ το username το μετατρέπουμε σε συμβολοσειρά. Η μέθοδος επιστρέφει τη μεταβλητή user (του αντικειμένου User) όπως αυτή έχει διαμορφωθεί από το βρόγχο.

Η παρακάτω μέθοδος κατακερματίζει τον κωδικό του χρήστη:

```
private static string HashPassword(string str)
{
    using (var sha256 = SHA256.Create())
    {
        var hashedBytes =
sha256.ComputeHash(System.Text.Encoding.UTF8.GetBytes(str));
        var hash = BitConverter.ToString(hashedBytes).Replace("-",
""),
return hash;
    }
}
```

Ο αλγόριθμος sha256 είναι ένας από τους πιο ισχυρούς αλγόριθμους κατακερματισμού συμβολοσειρών παγκοσμίως και μπορούμε να πούμε με ασφάλεια ότι δεν «σπάει», τουλάχιστον σε πολύ μεγάλο χρονικό διάστημα με τη σημερινή υπολογιστική ισχύ. Ο αλγόριθμος δεν υπάρχει εγγενώς στο VS 2022 και τον αναζητήσαμε στο NuGet, ο οποίος έχει την ονομασία K4os.Hash.xxHash.

Προχωρούμε στη μέθοδο GetChapters, η οποία κάνει ερώτηση επιλογής στον πίνακα chapters και πραγματοποιείται μετά την επιτυχή ταυτοποίηση του χρήστη σε νέα WinForm

```
public static MySqlDataReader GetChapters(DB dbCon)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT * FROM `chapters`";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}
```

Επιστρέφει ένα αντικείμενο MySqlDataReader τα αποτελέσματα του οποίου θα διαβαστούν στο αρχείο IntroForm.cs.

Η επόμενη μέθοδος θα λάβει τις συνολικές ερωτήσεις για τη ζητούμενη ενότητα (μεταβλητή section) από τον πίνακα questions:

```
public static MySqlDataReader GetTotalQuestionsBySection(DB dbCon, string section)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT `id` AS `cnt` FROM `questions` WHERE `section` =
@section";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@section", section);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
    }
}
```

```

        return db_read;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}

```

Η επόμενη μέθοδος είναι υπεύθυνη για τη λήψη τυχαίων ερωτήσεων για τη διενέργεια μιας εξέτασης:

```

public static MySqlDataReader GetQuestionsBySection(DB dbCon, string section, int
count)
{
    MySqlDataReader db_read;
    try
    {
        //Console.WriteLine("section: "+ section);
        var sql = @"SELECT * FROM `questions` WHERE `section` = @section
ORDER BY RAND() LIMIT @count";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@section", section);
        cmd.Parameters.AddWithValue("@count", count);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}

```

Δίνουμε ως όρισμα μια ενότητα και ένα μέγιστο πλήθος ερωτήσεων. Τα αποτελέσματα ταξινομούνται τυχαία, επομένως διασφαλίζουμε την τυχαία επιλογής τους.

Με την παρακάτω μέθοδο θα λάβουμε την εξέταση με βάση το id της και χρησιμοποιείται για την εμφάνιση των αποτελεσμάτων μετά το πέρας της εξέτασης:

```

public static MySqlDataReader GetTestById(DB dbCon, long testId)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT * FROM `tests` WHERE id = @testId";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@testId", testId);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}

```

Η παρακάτω μέθοδος θα επιστρέψει ένα αντικείμενο `MySqlDataReader`, το οποίο περιέχει το συνολικό αριθμό ενοτήτων κάνοντας ερώτημα επιλογής στον πίνακα των κεφαλαίων. Ο τελεστής `COUNT` κάνει καταμέτρηση των εγγραφών:



```

public static MySqlDataReader GetTotalSections(DB dbCon)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT COUNT(*) AS `cnt` FROM `chapters` GROUP BY
`section`";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}

```

Η παρακάτω μέθοδος καταγράφει ένα συμβάν ανάγνωσης της ύλης ενός κεφαλαίου όταν ο χρήστης πατήσει το σχετικό κουμπί:

```

public static long LogReadAction(DB dbCon, int userId, string chapterId)
{
    long id = 0;
    if (chapterId != "")
    {
        try
        {
            var sql = @"INSERT INTO `read_actions` (`user_id`, `chapter_id`)
VALUES(@user_id, @chapter_id)";
            var cmd = new MySqlCommand(sql, dbCon.Connection);
            cmd.Parameters.AddWithValue("@user_id", userId);
            cmd.Parameters.AddWithValue("@chapter_id", chapterId);
            cmd.ExecuteNonQuery();
            id = cmd.LastInsertedId;
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }
    return id;
}

```

Θα κάνει ερώτημα προσθήκης στον πίνακα read\_actions καταγράφοντας το id του χρήστη και το id του κεφαλαίου. Επιστρέφει το id που μόλις εισήχθη στον πίνακα.

Με την παρακάτω μέθοδο θα καταγράψουμε το συμβάν εκκίνησης εξέτασης.

```

public static long StartTest(DB dbCon, int userId, string sectionId, string mode,
int totalQuestions)
{
    long id = 0;
    if (sectionId != "")
    {
        try
        {
            var sql = @"INSERT INTO `tests` (`user_id`, `section_id`, `mode`,
`total_questions`) VALUES(@user_id, @section_id, @mode, @total_questions)";
            var cmd = new MySqlCommand(sql, dbCon.Connection);
            cmd.Parameters.AddWithValue("@user_id", userId);
            cmd.Parameters.AddWithValue("@section_id", sectionId);
            cmd.Parameters.AddWithValue("@mode", mode.Equals("0") ? 0 : 1);
            cmd.Parameters.AddWithValue("@total_questions", totalQuestions);
            cmd.ExecuteNonQuery();
            id = cmd.LastInsertedId;
        }
    }
}

```

```

    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.Message);
    }
}
return id;
}

```

Για την καταγραφή χρειαζόμαστε το id του χρήστη, την ενότητα, το εάν η εξέταση είναι διαγώνισμα ή εξάσκηση και το συνολικό αριθμό των ερωτήσεων της εξέτασης. Τα δεδομένα αυτά εισάγονται στον πίνακα tests. Επιστρέφει το id της εξέτασης που μόλις εισήχθη.

Η παρακάτω μέθοδος καταγράφει το συμβάν ολοκλήρωσης εξέτασης στον πίνακα tests:

```

public static void EndTest(DB dbCon, long id, int questionsAnswered, int
correctQuestions, string result)
{
    try
    {
        var sql = @"UPDATE `tests` SET `questions_answered` =
@questions_answered, `correct_questions` = @correct_questions, `result` = @result,
`completed_at` = NOW() WHERE `id` = @id";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@questions_answered",
questionsAnswered);
        cmd.Parameters.AddWithValue("@correct_questions", correctQuestions);
        cmd.Parameters.AddWithValue("@result", result);
        cmd.Parameters.AddWithValue("@id", id);
        cmd.ExecuteNonQuery();
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.Message);
    }
}

```

Ενημερώνουμε τον πίνακα tests με βάση το id της εξέτασης που λάβαμε από την προηγούμενη μέθοδο με δεδομένα τον αριθμό των απαντημένων ερωτήσεων, τις σωστές ερωτήσεις του μαθητή και το αποτέλεσμα που έλαβε (pass εάν πέρασες, fail εάν κόπηκε).

Ακολουθεί η μέθοδος που ενημερώνει μία ανάγνωση κεφαλαίου όταν αυτή ολοκληρωθεί (ο μαθητής κλείσει το σχετικό παράθυρο). Έχοντας το id που επιστρέφει η μέθοδος LogReadAction, θα ενημερώσουμε τη στήλη completed\_at με την τρέχουσα χρονοσφραγίδα (NOW()):

```

public static void UpdateReadAction(DB dbCon, long readActionId)
{
    if (readActionId > 0)
    {
        try
        {
            var sql = @"UPDATE `read_actions` SET `completed_at` = NOW()
WHERE `id` = @id";
            var cmd = new MySqlCommand(sql, dbCon.Connection);
            cmd.Parameters.AddWithValue("@id", readActionId);
            cmd.ExecuteNonQuery();
        }
        catch (MySqlException ex)
        {
            Console.WriteLine(ex.Message);
        }
    }
}

```

Η παρακάτω μέθοδος επιστρέφει τις συνολικές αναγνώσεις κεφαλαίων με βάση το id του χρήστη:

```
public static MySqlDataReader GetTotalReadsByUserId(DB dbCon, int userId)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT COUNT(*) AS `cnt` FROM `read_actions` WHERE
`user_id` = @user_id";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@user_id", userId);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}
```

Η ακόλουθη μέθοδος επιστρέφει το μέσο χρόνο ανάγνωσης σε δευτερόλεπτα της ύλης για ένα χρήστη που δίνεται ως όρισμα:

```
public static MySqlDataReader GetAvgTimeForReadActionByUserIdInSeconds(DB dbCon, int
userId)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT AVG(TIMESTAMPDIFF(SECOND, `created_at`,
`completed_at`)) AS `avg` FROM `read_actions` WHERE `user_id` = @user_id";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@user_id", userId);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}
```

Αφαιρούμε την τιμή (χρονοσφραγίδα) της στήλης created\_at από τη στήλη completed\_at και λαμβάνουμε το μέσο όρο με τον τελεστή AVG για όλες τις εγγραφές.

Η παρακάτω μέθοδος επιστρέφει τις συνολικές εξετάσεις από τον πίνακα tests γνωρίζοντας το id του χρήστη και το είδος της εξέτασης (διαγώνισμα ή εξάσκηση):

```
public static MySqlDataReader GetTotalTestsByModeAndByUserId(DB dbCon, int userId,
int mode)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT COUNT(*) AS `cnt` FROM `tests` WHERE `user_id` =
@user_id AND mode = @mode";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@user_id", userId);
        cmd.Parameters.AddWithValue("@mode", mode);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
}
```

```

    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}

```

Η παρακάτω μέθοδος επιστρέφει τις εγγραφές του πίνακα tests με την προϋπόθεση ότι τα διαγωνίσματα που σχετίζονται έχουν ολοκληρωθεί με βάση το id του χρήστη:

```

public static MySqlDataReader GetLastValidExamByUserId(DB dbCon, int userId)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT * FROM `tests` WHERE `user_id` = @user_id AND
`mode` = 1 and `result` IS NOT NULL ORDER BY `id` DESC LIMIT 1";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@user_id", userId);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}

```

Με την παρακάτω μέθοδο γνωρίζουμε τις συνολικές εξασκήσεις ανά ενότητα με βάση το id του χρήστη:

```

public static MySqlDataReader GetCompletedSectionPracticesByUserId(DB dbCon, int
userId)
{
    MySqlDataReader db_read;
    try
    {
        var sql = @"SELECT DISTINCT `section_id` FROM `tests` WHERE `user_id`
= @user_id AND `mode` = 0 and `result` IS NOT NULL";
        var cmd = new MySqlCommand(sql, dbCon.Connection);
        cmd.Parameters.AddWithValue("@user_id", userId);
        cmd.ExecuteNonQuery();
        db_read = cmd.ExecuteReader();
        return db_read;
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.Message);
        return null;
    }
}

```

Η παρακάτω μέθοδος τερματίζει την επικοινωνία με τη βάση:

```

public void Close()
{
    Connection.Close();
}

```

### 3.4 Φόρμα πρόσβασης (SignInForm.cs)

Αυτή είναι η πρώτη φόρμα εκκίνησης της εφαρμογής. Παρατίθεται ο δημιουργός της κλάσης

```
public SignInForm()
{
    InitializeComponent();
    InstantiateDB();

    registerLinkLbl.Click += RegisterLink_Click;
    signBtn.Click += SignBtn_Click;

    //debug();
}
```

Η μέθοδος InitializeComponent είναι απαραίτητη σε όλες τις κλάσεις εφόσον αυτές προβάλλουν γραφικό περιβάλλον. Η μέθοδος InstantiateDB δημιουργεί ένα νέο αντικείμενο της κλάσης DB για τη διασύνδεση με τη βάση. Ακολουθούν οι ορισμοί των ακροατών συμβάντων στα κουμπιά Εισόδου και Εγγραφής στο σύστημα.

```
void SignBtn_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(usernameTxt.Text) ||
        string.IsNullOrEmpty(passwordTxt.Text))
    {
        MessageBox.Show("Εισάγετε το όνομα χρήστη και τον κωδικό σας",
            Constants.appTitle);
        return;
    }
    if (signBtn.Text == "Εγγραφή")
    {
        if (dbCon.IsConnect())
        {
            if (DB.UserExists(dbCon, usernameTxt.Text.ToString()))
            {
                MessageBox.Show("Το όνομα χρήστη που δώσατε υπάρχει ήδη στο
                    σύστημα!", Constants.appTitle);
                return;
            }
            else
            {
                if (DB.Register(dbCon, usernameTxt.Text.ToString(),
                    passwordTxt.Text.ToString()) )
                {
                    MessageBox.Show("Επιτυχής δημιουργία λογαριασμού.
                        Μπορείτε τώρα να πραγματοποιήσετε είσοδο", Constants.appTitle);
                    signBtn.Text = "Είσοδος";
                }
            }
        }
    }
    else
    {
        if (dbCon.IsConnect())
        {
            User user = DB.Login(dbCon, usernameTxt.Text.ToString(),
                passwordTxt.Text.ToString());
            //Console.WriteLine(user.getId());
            if ( user.getId() <= 0 )
            {
                MessageBox.Show("Μη έγκυρα στοιχεία εισόδου",
                    Constants.appTitle);
            }
            else
            {
                this.Hide();
                var introform = new IntroForm(user.getId(),
                    user.getUsername());
            }
        }
    }
}
```

```

        introform.Closed += (s, args) => this.Close();
        introform.Show();
    }
}
}

```

Στο συμβάν εγγραφής ή εισόδου στην εφαρμογή, θα γίνει έλεγχος εάν ο χρήστης έχει παράσχει έγκυρα (μη κενά) στοιχεία εισόδου (username και password). Σε αυτήν την περίπτωση θα εμφανιστεί ένα πλαίσιο διαλόγου (MessageBox) ενημερώνοντας σχετικά το χρήστη. Μετά θα γνωρίσουμε τι θέλει να κάνει ο χρήστης λαμβάνοντας το κείμενο του κουμπιού καθώς η ίδια μέθοδος διαχειρίζεται και το συμβάν εισόδου και εγγραφής. Εάν το συμβάν πρόκειται για εγγραφή, θα γίνει έλεγχος στον πίνακα users εάν το username που δίνει ο χρήστης υπάρχει ήδη και σε αυτήν την περίπτωση ο χρήστης θα ενημερωθεί σχετικά. Αλλιώς θα γίνει εγγραφή του χρήστη στον πίνακα users με τη μέθοδο DB.Register. Στο συμβάν εισόδου, εφόσον ταυτοποιηθεί ο χρήστης με τη μέθοδο DB.Login, η τελευταία επιστρέφει ένα αντικείμενο User. Εάν το αντικείμενο δεν είναι έγκυρο (το id του χρήστη είναι αρνητικό), ο χρήστης ενημερώνεται σχετικά. Διαφορετικά θα κρυφτεί η τρέχουσα φόρμα και θα κατευθύνουμε το χρήστη στη φόρμα IntroForm με ορίσματα το id και το username του χρήστη.

### 3.5 Κεντρική φόρμα (IntroForm.cs)

Σε αυτό το σημείο είναι σημαντικό να αναφέρουμε ότι το υλικό του κεφαλαίου προβάλλεται σε στοιχείο ελέγχου WebView2 και είναι βελτιωμένο αντικείμενο προβολής διαδικτυακών πόρων (αλλά και τοπικών αρχείων όπως HTML, Javascript PDF κλπ.) που δημιούργησε η Microsoft και αποτελεί τη νεότερη και αναβαθμισμένη έκδοση του παλαιότερου WebView. Πρόκειται ουσιαστικά για έναν browser (φυλλομετρητή Ιστού) και χρησιμοποιεί τη μηχανή Chromium του Microsoft Edge, δηλαδή του επίσημου προγράμματος της εταιρείας για την πλοήγηση στο Internet. Αυτό το στοιχείο χρησιμοποιούμε για την ανάγνωση του υλικού των κεφαλαίων εκ μέρους του μαθητή. Ο browser εκτελείται σε ξεχωριστό νήμα της εφαρμογής και όχι στο κύριο UI νήμα (για λόγους απόδοσης της εφαρμογής) γι' αυτό και είναι απαραίτητο να εκκινήσουμε με ασύγχρονο τρόπο ένα περιβάλλον εργασίας τους WebView2 κατά τη φόρτωση της φόρμας όπως φαίνεται στην παρακάτω μέθοδο:

```

async void TheoryForm_Load(object sender, EventArgs e)
{
    var userDataFolder =
Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.LocalApplicationData), Constants.appTitle);
    var webView2Environment = CoreWebView2Environment.CreateAsync(null,
userDataFolder).Result;
    await browser.EnsureCoreWebView2Async(webView2Environment);
}

```

Η φόρμα IntroForm είναι η κεντρική και μέσω αυτής ο χρήστης μπορεί να διαβάσει την ύλη ή να ξεκινήσει κάποια εξέταση. Παρακάτω φαίνεται ο δημιουργός της:

```

public IntroForm(int userId, string username)
{
    allSections = new List<string>();
    completedPractices = new List<string>();

    this.userId = userId;

    InitializeComponent();
    InstantiateDB();
    InitStatistics();
    GetCompletedPractices();
    FillDataGridView();

    if (username != null)
    {
        this.username = username;
        usernameTxt.Text = "Καλησπέρα " + username;
    }

    appDir = Path.GetDirectoryName(Application.ExecutablePath);
    //Console.WriteLine(appDir);
    debugText.Text = appDir;
    debugText.Visible = false;

    theoryTable.CellContentClick += TheoryTableCellClick;
    closeBrowserBtn.Click += CloseBrowserClick;
    practiceBtn.Click += PracticeBtnClick;
    examBtn.Click += ExamBtnClick;
    sectionCombo.SelectedIndexChanged += sectionSelectedIndexChanged;

    // DEBUG
    //sectionCombo.SelectedIndex = 0;
}

```

Θα αρχικοποιήσουμε ορισμένες μεταβλητές όπως μια λίστα (List) τύπου string για να γνωρίζουμε όλες τις ενότητες, μια λίστα που θα λάβει τις ολοκληρωμένες εξασκήσεις, το id του χρήστη, το username του (ως μέρος του καλωσορίσμάτός του στη φόρμα) και το φάκελο από τον οποίο εκτελείται το exe αρχείο της εφαρμογής έτσι ώστε να γνωρίζουμε πού βρίσκονται τα τοπικά αρχεία (PDF) της εφαρμογής. Επίσης θα βρούμε τη μέθοδο InitializeComponent η οποία όπως προαναφέρθηκε είναι απαραίτητη για κάθε εφαρμογή σε WinForms. Η μέθοδος InstantiateDB έχει περιγραφεί στο προηγούμενο αρχείο και αρχικοποιεί την επικοινωνία με τη βάση. Η μέθοδος InitStatistics θα εμφανίσει στατιστικά στοιχεία για το μαθητή και το σώμα της οποίας το βλέπουμε παρακάτω:

```

void InitStatistics()
{
    // GET TOTAL READS
    var db_read = DB.GetTotalReadsByUserId(dbCon, userId);
    long totalReads = 0;
    if (db_read.HasRows)
    {
        while (db_read.Read())
        {
            totalReads = (long) db_read["cnt"];
            break;
        }
    }
    db_read.Close();
    totalReadsLabel.Text += " " + totalReads;

    // GET AVG READ TIME
    db_read = DB.GetAvgTimeForReadActionByUserIdInSeconds(dbCon, userId);
    double avgReadTime = 0;
    if (db_read.HasRows)

```

```

    while (db_read.Read())
    {
        if (String.IsNullOrEmpty(db_read["avg"].ToString()) == false)
        {
            avgReadTime = Math.Truncate(Convert.ToDouble(db_read["avg"])
* 100) / 100;
        }
        break;
    }
}
db_read.Close();
if (avgReadTime > 0)
{
    totalReadsLabel.Text += "\nΜέσος χρόνος ανάγνωσης: " + avgReadTime +
" δεπτ.";
}

// GET TOTAL PRACTICES
db_read = DB.GetTotalTestsByModeAndByUserId(dbCon, userId, 0);
long totalPractices = 0;
if (db_read.HasRows)
{
    while (db_read.Read())
    {
        totalPractices = (long) db_read["cnt"];
        break;
    }
}
db_read.Close();
totalPracticesLabel.Text += " " + totalPractices;

// GET TOTAL EXAMS
db_read = DB.GetTotalTestsByModeAndByUserId(dbCon, userId, 1);
long totalExams = 0;
if (db_read.HasRows)
{
    while (db_read.Read())
    {
        totalExams = (long) db_read["cnt"];
        break;
    }
}
db_read.Close();
totalExamsLabel.Text += " " + totalExams;

// GET LAST VALID EXAM
db_read = DB.GetLastValidExamByUserId(dbCon, userId);
string lastResult = "";
string grade = "";
if (db_read.HasRows)
{
    while (db_read.Read())
    {
        lastResult = db_read["correct_questions"].ToString() + "/" +
db_read["total_questions"].ToString();
        if (db_read["correct_questions"].ToString().Equals("pass"))
        {
            grade = "(Περάσατε)";
        }
        else
        {
            grade = "(Κοπήκατε)";
        }
    }
}
lastResult += " " + grade;
db_read.Close();
lastExamLabel.Text += " " + lastResult;
}

```



Αρχικά θα λάβουμε τις συνολικές αναγνώσεις της ύλης των κεφαλαίων και θα εμφανίσουμε το νούμερο που θα επιστρέφει σε μια ετικέτα (Label) δεξιά στη φόρμα. Παρόμοια θα κάνουμε (όπως και τις επόμενες ενέργειες) και για το μέσο χρόνο ανάγνωσης της ύλης. Κατόπιν θα λάβουμε τις αριθμούς των συνολικών εξασκήσεων και διαγωνισμάτων και τέλος τον τελευταίο βαθμό (και το εάν πέρασε ή κόπηκε ο μαθητής) διαγωνίσματος. Τα συσχετιζόμενα ερωτήματα τα έχουμε περιγράψει σε προηγούμενη ενότητα.

Στη συνέχεια κάνουμε κλήση στη μέθοδο `GetCompletedPractices` για να λάβουμε τις συνολικές εξασκήσεις του μαθητή με σκοπό τη συμπλήρωση της λίστας τύπου `string completedPractices`. Τη λίστα τη χρησιμοποιούμε σε επόμενη μέθοδο για να διαπιστώσουμε εάν ο χρήστης έχει πραγματοποιήσει την εξάσκηση μιας ενότητας για να έχει τη δυνατότητα να κάνει το διαγώνισμα της εκάστοτε ενότητας. Το σώμα της φαίνεται παρακάτω:

```
void GetCompletedPractices()
{
    var db_read = DB.GetCompletedSectionPracticesByUserId(dbCon, userId);
    if (db_read.HasRows)
    {
        while (db_read.Read())
        {
            if (completedPractices.Contains(db_read["section_id"].ToString())
== false)
            {
                completedPractices.Add(db_read["section_id"].ToString());
            }
        }
    }
    db_read.Close();
    //completedPractices.ForEach(p => Console.WriteLine(p));
}
```

Η μέθοδος `FillDataGridView` είναι υπεύθυνη για τη συμπλήρωση του κεντρικού πίνακα με πληροφορίες των κεφαλαίων έτσι ώστε ο μαθητής να έχει τη δυνατότητα να διαβάσει την ύλη τους. Αρχικά θα κάνουμε κλήση στη μέθοδο `GetChapters` (μέσω του DB αντικειμένου) και για κάθε εγγραφή που θα επιστραφεί θα εισάγουμε μία γραμμή στο `DataGridView`. Θα ελέγξουμε τη στήλη `type` κάθε εγγραφής και εάν έχει την τιμή `pdf`, τότε εμφανίζουμε στην 4<sup>η</sup> στήλη την τιμή 'Εγγραφο'. Αντίστοιχα θα γίνει και για τις τιμές `online_page`, `online_video` της ίδιας στήλης όπου θα εμφανίσουμε Διαδικτυακός πόρος και Διαδικτυακό βίντεο αντίστοιχα. Παράλληλα σε κάθε επανάληψη εγγραφής θα γίνεται κλήση στη μέθοδο `FillSections` για να συμπληρώσουμε τη λίστα (ComboBox) με κάθε ενότητα μέσω της οποίας (της λίστας) επιλέγει την ενότητα που επιθυμεί να κάνει εξέταση. Επίσης συμπληρώνουμε και τη λίστα `allSections` για να γνωρίζουμε τις ενότητες που υπάρχουν στη βάση κατά μήκος της εφαρμογής. Παρακάτω είναι το σώμα της μεθόδου που μόλις περιγράψαμε:

```
void FillDataGridView()
{
    var db_read = DB.GetChapters(dbCon);
    if (db_read.HasRows)
    {
        int row = 0;
        while (db_read.Read())
        {
```

```

var typeNice = "";
if (db_read["type"].Equals("pdf")){
    typeNice = "Έγγραφο";
}
else if (db_read["type"].Equals("online_page"))
{
    typeNice = "Διαδικτυακός πόρος";
}
else if (db_read["type"].Equals("online_video"))
{
    typeNice = "Διαδικτυακό βίντεο";
}
theoryTable.Rows.Add(
    db_read["id"],
    db_read["section"],
    db_read["title"],
    typeNice
);

FillSections(db_read["section"].ToString());

theoryTable.Rows[row].Cells[0].Tag = db_read["id"].ToString();

theoryTable.Rows[row].Cells[readCol].Value = "Προβολή";
theoryTable.Rows[row].Cells[readCol].Tag =
db_read["file"].ToString();

        row++;
    }
}
db_read.Close();
}

```

Θα ορίσουμε ακροατές συμβάντων στα στοιχεία ελέγχου της φόρμας. Ο κυριότερος ακροατής είναι στο συμβάν click σε ένα κελί (μέθοδος TheoryTableCellClick). Θα γίνει έλεγχος ότι ο χρήστης έχει επιλέξει ακριβώς μία γραμμή στον πίνακα των κεφαλαίων και θα λάβουμε το id του κεφαλαίου που βρίσκεται στην 1<sup>η</sup> στήλη της επιλεγμένης γραμμής. Ύστερα θα βεβαιωθούμε ότι ο χρήστης έχει πατήσει την 4<sup>η</sup> στήλη της γραμμής, δηλαδή έχει πρόθεση να διαβάσει την ύλη του κεφαλαίου. Η 4<sup>η</sup> στήλη είναι τύπου LinkLabel, έχει δηλαδή τη μορφοποίηση ενός υπερσυνδέσμου. Σημειώνουμε ότι κατά την εισαγωγή κάθε γραμμής σε προηγούμενη μέθοδο στον πίνακα στην 4<sup>η</sup> στήλη έχουμε ορίσει στην ετικέτα (tag) του LinkLabel τη διαδρομή του αρχείου με την οποία σχετίζεται η ύλη του. Επομένως, θα λάβουμε το tag του επιλεγμένου κελιού και θα αρχικοποιήσουμε νέο αντικείμενο URI. Το εν λόγω αντικείμενο είναι απαραίτητο για την προβολή ιστοσελίδων και PDF μέσα από το αντικείμενο WebView2, που θα περιγράψουμε στη συνέχεια. Εάν το αρχείο έχει την κατάληξη PDF τότε θα αναζητήσουμε το φάκελο assets στον οποία βρίσκονται τα αρχεία PDF. Η διαδρομή του φακέλου είναι διαφορετική όταν εκτελείται η εφαρμογή μέσα από το Visual Studio (διαμέσου του debugging) και όταν εκτελείται μετά την επίσημη εγκατάσταση της εφαρμογής. Εάν το υλικό του κεφαλαίου είναι ιστοσελίδα τότε το URI θα ισούται με το URL αυτής. Εάν το υλικό είναι YouTube βίντεο, τότε θα κάνουμε κλήση στη μέθοδο NavigateToString (επί του αντικειμένου browser, δηλαδή του WebView2) δίνοντας ως όρισμα την επιστροφή της μεθόδου EmbedVideo με όρισμα το URL του βίντεο και θα περιγραφούν στη συνέχεια. Σε κάθε άλλη περίπτωση, εμφανίζεται μήνυμα λάθους αναφέροντας ότι δε βρέθηκε υλικό στο επιλεγμένο κεφάλαιο σε πλαίσιο διαλόγου και θα τερματιστεί η εκτέλεση. Τελικά θα εμφανιστεί το WebView έχοντας φορτώσει το αρχείο ή την

ιστοσελίδα σε όλο το διαθέσιμο χώρο της φόρμας και θα αποκρύψει τα υπόλοιπα στοιχεία της φόρμας με την εντολή `ToggleElementsVisibility` (αλλά εμφανίζοντας το κουμπί Κλείσιμο) για να επιτρέψει στο μαθητή να μελετήσει άνετα το υλικό. Ταυτόχρονα θα γίνει καταγραφή του συμβάντος εκκίνησης της ανάγνωσης της ύλης του επιλεγμένου κεφαλαίου (μέθοδος `LogReadAction`) για να έχουμε στατιστικά δεδομένα.

Στην επόμενη μέθοδο βλέπουμε πώς μπορούμε να ενσωματώσουμε ένα YouTube στον browser της εφαρμογής. Είναι απαραίτητο το βίντεο να ανήκει σε ένα έγκυρο έγγραφο HTML και να προβάλλει το βίντεο μέσω ενός `iframe`. Σημειώνουμε ότι δεν μπορούμε να προβάλλουμε βίντεο σε `iframe` έχοντας το URL του βίντεο, αλλά μπορούμε να το κάνουμε μέσω του κώδικα ενσωμάτωσης `iframe` που παρέχει το YouTube γι' αυτό το σκοπό όντας βρισκόμενοι στη διεύθυνση του βίντεο. Το σώμα της μεθόδου `EmbedVideo` φαίνεται παρακάτω:

```
string EmbedVideo(string url)
{
    var embed = "<html><head>" +
        "<meta http-equiv=\"X-UA-Compatible\" content=\"IE=Edge\"/>" +
        "</head><body>" +
        "<iframe width='1280' height='400' src='{0}' frameborder='0' " +
        "allow='accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture' allowfullscreen></iframe>" +
        "</body></html>";
    return string.Format(embed, url);
}
```

Παρακάτω βλέπουμε το σώμα της μεθόδου που περιγράψαμε προηγουμένως δηλαδή της `TheoryTableCellClick`:

```
void TheoryTableCellClick(object sender, EventArgs e)
{
    //await browser.EnsureCoreWebView2Async().ConfigureAwait(true);
    Int32 selectedRow =
theoryTable.Rows.GetRowCount(DataGridViewElementStates.Selected);
    // ΑΝΑΓΝΩΣΗ
    if ( selectedRow == 1 )
    {
        int row = theoryTable.CurrentCell.RowIndex;
        int col = theoryTable.CurrentCell.ColumnIndex;
        currentChapterId = theoryTable.Rows[row].Cells[0].Value.ToString();
        //Console.WriteLine("row: "+row);
        Console.WriteLine("col: " + col);
        theoryTable.ClearSelection();
        // ΑΝΑΓΝΩΣΗ
        if (col == readCol)
        {
            var file = theoryTable.Rows[row].Cells[col].Tag.ToString();
            Uri uri = new Uri("about:blank");
            if (file.Contains("pdf"))
            {
                var myFile = Path.Combine(appDir, "assets\\" + file);
                //Debug.WriteLine(myFile);
                // MY LOCAL DEBUGGING PATH
                if (appDir.Contains("projects"))
                {
                    myFile = Path.Combine(appDir, "..\\..\\assets\\" + file);
                }
                if ( File.Exists(myFile) )
                {
                    uri = new Uri(myFile);
                }
            }
            else if (file.Contains("youtube"))

```

```

    {
        browser.NavigateToString(EmbedVideo(file));
    }
    else if (file.Contains("http"))
    {
        uri = new Uri(file);
    }
    else
    {
        MessageBox.Show("Δε βρέθηκε υλικό στο συγκεκριμένο κεφάλαιο",
Constants.appTitle);
        return;
    }

    readActionId = DB.LogReadAction(dbCon, userId, currentChapterId);

    browser.Dock = DockStyle.Fill;
    closeBrowserBtn.BringToFront();
    ToggleElementsVisibility(false);

    if ( uri.Equals("about:blank") == false )
    {
        browser.Source = uri;
    }
}
}
}
}

```

Η παρακάτω μέθοδος εκτελείται όταν ο χρήστης πατήσει το κουμπί Κλείσιμο όταν ολοκληρώσει τη μελέτη υλικού. Εισάγουμε κενό HTML περιεχόμενο στον browser, επαναφέρουμε το φόρμα στην αρχική κατάσταση και καταγράφουμε το συμβάν ολοκλήρωσης της ανάγνωσης. Το σώμα της φαίνεται παρακάτω:

```

void CloseBrowserClick(object sender, EventArgs e)
{
    browser.NavigateToString("<html><head><body></body></html>");
    browser.Source = new Uri("about:blank");
    browser.Dock = DockStyle.None;
    ToggleElementsVisibility(true);
    DB.UpdateReadAction(dbCon, readActionId);
    currentChapterId = "";
    readActionId = 0;
}

```

Το συμβάν επιλογής στο ComboBox, δηλαδή στο στοιχείο ελέγχου επιλογής ενότητας κάτω από τον πίνακα των κεφαλαίων διαχειρίζεται από τη μέθοδο sectionSelectedIndexChanged. Από προεπιλογή τα διαγωνίσματα για ενότητες είναι κλειδωμένα, δηλαδή το κουμπί Διαγώνισμα είναι ανενεργό, δεν μπορεί να το πατήσει ο χρήστης. Θα γίνει έλεγχος εάν η επιλεγμένη ενότητα υπάρχει στη λίστα completedPractices, που περιγράψαμε νωρίτερα. Εάν υπάρχει, δηλαδή ο μαθητής έχει πραγματοποιήσει την εξάσκηση για επιθυμητή ενότητα, τότε το κουμπί Διαγώνισμα θα ξεκλειδώσει, δηλαδή θα γίνει ενεργό επιτρέποντας στο μαθητή να κάνει το διαγώνισμα εφόσον επιθυμεί. Σε αντίθετη περίπτωση, το κουμπί θα γίνει μη ενεργό. Παρακάτω έχουμε το σώμα της μεθόδου:

```

void sectionSelectedIndexChanged(object sender, EventArgs e)
{
    if (String.IsNullOrEmpty(sectionCombo.Text) == true)
    {
        return;
    }
}

```

```

    }
    var selection = sectionCombo.SelectedItem.ToString();
    //Console.WriteLine("selection: "+ selection);
    if ( completedPractices.Contains(selection) )
    {
        examBtn.Enabled = true;
    }
    else
    {
        examBtn.Enabled = false;
    }
}
}

```

Η τελευταία μέθοδος που θα περιγράψουμε είναι όταν ο μαθητής πατήσει το κουμπί Εξάσκηση ή Διαγώνισμα. Το κουμπί Εξάσκηση θα θέσει τη μεταβλητή mode ως 0, ενώ το κουμπί Διαγώνισμα θα τη θέσει ως 1 και θα γίνει κλήση στη μέθοδο GoToTestForm με όρισμα τη μεταβλητή αυτή. Η τρέχουσα φόρμα θα κρυφτεί, ενώ θα εμφανιστεί η TestForm φόρμα με ορίσματα το id του χρήστη, το username του, την ενότητα που επιλέγει ο χρήστης να εξεταστεί και το mode (Εξάσκηση ή Διαγώνισμα). Παρακάτω είναι το σώμα της:

```

void GoToTestForm(int mode)
{
    if (String.IsNullOrEmpty(sectionCombo.Text) == true)
    {
        return;
    }
    var selection = sectionCombo.SelectedItem.ToString();
    this.Hide();
    var form = new TestForm(userId, username, selection, mode);
    form.Closed += (s, args) => this.Close();
    form.Show();
}

```

### 3.6 Φόρμα εξέτασης (TestForm.cs)

Στο σώμα του δημιουργού θα περιγραφούν μόνο τα καινούρια στοιχεία. Εντός της κλάσης έχουμε μία λίστα με ονομασία questionsList με αντικείμενα τύπου Question, μια κλάση που δημιουργήσαμε για να διαχειριστούμε την εξέταση. Κατά την άφιξη του χρήστη σε αυτή τη φόρμα αρχικοποιούμε ορισμένες μεταβλητές έτσι ώστε να αυτές να εκκαθαριστούν από προηγούμενες εξετάσεις του χρήστη κατά μία εκτέλεση προγράμματος. Θα γίνει κλήση στη μέθοδο GetQuestions με σκοπό να λάβουμε τις ερωτήσεις της ενότητας. Θα καλέσουμε έπειτα τη μέθοδο GetQuestionsBySection επί του DB instance με ορίσματα την ενότητα και την επιστροφή της μεθόδου GetDownsizedQuestionCount. Η μέθοδος είναι το πρώτο μέρος του αλγορίθμου σχετικά με την επιλογή της τυχαιότητας των ερωτήσεων της εξέτασης. Θα λάβει το συνολικό αριθμό των ερωτήσεων της ενότητας προς εξάσκηση (μέθοδος GetTotalQuestionsBySection στο αντικείμενο DB). Θα ορίσουμε 2 μεταβλητές (minQuestions και maxQuestions), με την πρώτη να έχει το ελάχιστο αριθμό ερωτήσεων μιας ενότητας (10) και τη δεύτερη να έχει το μέγιστο αριθμό ερωτήσεων (30). Εάν ο συνολικός αριθμός ερωτήσεων είναι μεγαλύτερος των 2, τότε θα διαιρέσουμε το συνολικό αριθμό με το 2, δηλαδή θα λάβουμε τον αριθμό των μισών ερωτήσεων. Εάν ο αριθμός των μισών ερωτήσεων είναι μεγαλύτερος του μέγιστου αριθμού, τότε τροποποιούμε τον αριθμό των μισών

ερωτήσεων έτσι ώστε να ισούται με 30. Εάν ο αριθμός των μισών ερωτήσεων είναι μικρότερο του ελάχιστου αριθμού (10) τότε η μεταβλητή `halfQuestions` ισούται με 10. Τελικά επιστρέφεται το διαμορφωμένο νούμερο της μεταβλητής `halfQuestions`. Εάν ο συνολικός αριθμός των ερωτήσεων μιας ενότητας είναι μικρότερος του 3, τότε επιστρέφεται αυτός ο αριθμός και δε γίνεται καμία επεξεργασία επί του αριθμού των εμφανιζόμενων ερωτήσεων. Σώμα μεθόδου:

```
int GetDownsizedQuestionCount()
{
    var db_read = DB.GetTotalQuestionsBySection(dbCon, sectionId);
    int totalQuestions = 0;
    if (db_read.HasRows)
    {
        while (db_read.Read())
        {
            totalQuestions++;
        }
    }
    db_read.Close();
    //Console.WriteLine("total: " + totalQuestions);
    int minQuestions = 10;
    int maxQuestions = 30;
    if (totalQuestions > 2)
    {
        int halfQuestions = totalQuestions / 2;
        //Console.WriteLine("total: " + totalQuestions);
        //Console.WriteLine("half: " + halfQuestions);
        if (halfQuestions > maxQuestions)
        {
            halfQuestions = maxQuestions;
        }
        if (halfQuestions < minQuestions)
        {
            halfQuestions = minQuestions;
        }
        return halfQuestions;
    }
    return totalQuestions;
}
```

Επιστρέφουμε στη μέθοδο `GetQuestions` και σε ένα βρόγχο διαβάζουμε τις εγγραφές που επιστρέφονται από το ερώτημα επιλογής στον πίνακα `questions`. Σε κάθε επανάληψη, δημιουργούμε ένα νέο αντικείμενο `Question` και θέτουμε τις ιδιότητες της κλάσης (`id`, `title`, `correct`, `type` και `chapterId`) με τις τιμές των αντίστοιχων στηλών της ερώτησης. Το αντικείμενο `Question` το εισάγουμε στη λίστα `questionsList` που είδαμε νωρίτερα και τέλος γίνεται κλήση στη μέθοδο `GoToNextQuestion`. Στην περίπτωση που δεν υπάρχουν ερωτήσεις για την επιλεγμένη ενότητα, τότε εμφανίζεται ένα πλαίσιο διαλόγου ενημερώνοντας το χρήστη σχετικά. Ο χρήστης μπορεί να πατήσει το κουμπί Πίσω για να μεταφερθεί στην προηγούμενη φόρμα. Το σώμα της μεθόδου που μόλις περιγράψαμε:

```
void GetQuestions()
{
    var db_read = DB.GetQuestionsBySection(dbCon, sectionId,
    GetDownsizedQuestionCount());
    if (db_read.HasRows)
    {
        int count = 0;
        while (db_read.Read())
        {
            var question = new Question();
            question.id = (int)db_read["id"];
            question.title = (string)db_read["title"];
        }
    }
}
```

```

        question.correct = (string)db_read["correct"];
        question.type = (string)db_read["type"];
        question.chapterId = (int)db_read["chapter_id"];
        questionsList.Add(question);
        count++;
    }
    db_read.Close();
    GoToNextQuestion(null, null, false);
}
else
{
    foreach (Control c in Controls)
    {
        c.Hide();
    }
    db_read.Close();
    questionHeader.Text = "Δε βρέθηκαν ερωτήσεις στη συγκεκριμένη
ενότητα";
    backButton.Visible = true;
    questionHeader.Visible = true;
}
}
}

```

Η μέθοδος που θα δούμε τώρα αναλαμβάνει να εμφανίσει στο χρήστη να δείξει την επόμενη ερώτηση στην εξέταση. Ελέγχουμε ότι ο αριθμός των ερωτήσεων είναι μεγαλύτερος από την τρέχουσα ερώτηση (μεταβλητή `currentQuestionId`) και μετά εάν η μεταβλητή `checkQuestion` είναι `false`, δηλαδή δεν έχει εμφανιστεί ακόμα η πρώτη ερώτηση, οπότε θα καταγράψουμε την εκκίνηση της εξέτασης στη βάση στον πίνακα `tests` δίνοντας ως ορίσματα το `id` του χρήστη, την ενότητα, το εάν είναι εξάσκηση ή διαγώνισμα (`mode` μεταβλητή) και το συνολικό αριθμό ερωτήσεων που έχει η εξέταση. Όταν εμφανιστεί η πρώτη ερώτηση τότε η μεταβλητή `checkQuestion` είναι `true`, οπότε θα κάνουμε κλήση στη μέθοδο `CanProceedToNextQuestion`, η οποία έχει ως σκοπό να ελέγξει εάν ο χρήστης έχει απαντήσει την ερώτηση. Θα γίνει διαφορετικός έλεγχος για το εάν η ερώτηση έχει απαντηθεί, ο οποίος εξαρτάται από τον τύπο της ερώτησης. Παρακάτω βλέπουμε τους ελέγχους:

1. Ερώτηση συμπλήρωσης κενού: Θα ελέγξουμε ότι το πεδίο κειμένου (`TextBox`) της φόρμας είναι κενό, περιέχει τουλάχιστον έναν χαρακτήρα.
2. Ερώτηση σωστού/λάθους: Όταν ο χρήστης κάνει `click` στο κουμπί Σωστό ή στο κουμπί Λάθος, ορίζουμε τη μεταβλητή `userSelection` ως 1 εάν ο μαθητής πάτησε το κουμπί Σωστό, αλλιώς 0 εάν ο μαθητής πάτησε το κουμπί του Λάθους. Επομένως ελέγχουμε ότι η μεταβλητή είναι μεγαλύτερη ή ίση του 0.
3. Ερώτηση πολλαπλής επιλογής: Έχουμε ένα `DropDown` στοιχείο στη φόρμα, το οποίο επιτρέπει να επιλέξει μία απάντηση από τις 4 διαθέσιμες. Επομένως, εάν ένα στοιχείο έχει επιλεγεί, τότε ο μαθητής έχει δώσει την απάντησή του.
4. Ερώτηση αντιστοίχισης: Εδώ έχουμε 8 στοιχεία ελέγχου `DropDown` στη φόρμα της ερώτησης. Όταν ο χρήστης έχει κάνει επιλογές και στα 8, τότε θεωρούμε ότι έχει δώσει την απάντησή του.

Η μέθοδος θα επιστρέψει `true` είτε όταν ο χρήστης έχει απαντήσει στην ερώτηση σε διαγώνισμα είτε όταν πρόκειται για εξάσκηση, οπότε δεν είναι απαραίτητη η απάντηση του χρήστη για να εμφανιστεί η επόμενη ερώτηση. Θα επιστρέψει `false` όταν ο χρήστης δεν έχει δώσει απάντηση



στην ερώτηση. Ταυτόχρονα η ίδια μέθοδος κάνει καταμέτρηση των απαντηθείσων ερωτήσεων, ένα νούμερο το οποίο καταγράφουμε για κάθε εξέταση. Το σώμα της μεθόδου που περιγράψαμε:

```
public bool CanProceedToNextQuestion()
{
    bool isQuestionAnswered = false;
    // FILL THE WORD
    if (questionsList[currentQuestionId].getType().Equals("fill_empty"))
    {
        if (String.IsNullOrEmpty(userAnswerTxt.Text) == false)
        {
            isQuestionAnswered = true;
        }
    }
    // CORRECT OR WRONG
    else if
    (questionsList[currentQuestionId].getType().Equals("correct_wrong"))
    {
        if (userSelection > -1)
        {
            isQuestionAnswered = true;
        }
    }
    // MAP
    else if (questionsList[currentQuestionId].getType().Equals("map"))
    {
        if (
            string.IsNullOrEmpty(userMapLeft1.Text) == false &&
            string.IsNullOrEmpty(userMapRight1.Text) == false &&
            string.IsNullOrEmpty(userMapLeft2.Text) == false &&
            string.IsNullOrEmpty(userMapRight2.Text) == false &&
            string.IsNullOrEmpty(userMapLeft3.Text) == false &&
            string.IsNullOrEmpty(userMapRight3.Text) == false &&
            string.IsNullOrEmpty(userMapLeft4.Text) == false &&
            string.IsNullOrEmpty(userMapRight4.Text) == false
        )
        {
            isQuestionAnswered = true;
        }
    }
    // MULTIPLE CHOICE
    else if (questionsList[currentQuestionId].getType().Equals("choice"))
    {
        if (string.IsNullOrEmpty(userChoice.Text) == false)
        {
            isQuestionAnswered = true;
        }
    }

    if (isQuestionAnswered == true)
    {
        questionsAnswered++;
    }
    if (mode == 0)
    {
        return true;
    }
    else
    {
        return isQuestionAnswered;
    }
}
}
```

Επιστρέφουμε στη μέθοδο GoToNextQuestion και εάν η μέθοδος CanProceedToNextQuestion επιστρέψει true τότε θα κάνουμε επαλήθευση της απάντησης του μαθητή, δηλαδή θα ελέγξουμε ότι είναι σωστή με τη μέθοδο ValidateUserAnswer. Διαφορετικά, εάν επιστραφεί false,



θα εμφανιστεί το πλαίσιο διαλόγου με μήνυμα: «Παρακαλώ δώστε την απάντησή σας» χωρίς να επιτρέπει το μαθητή να προχωρήσει εάν η ερώτηση δεν απαντηθεί.

Είμαστε στη μέθοδο `ValidateUserAnswer` και θα λάβουμε την απάντηση του συστήματος της ενεργής ερώτησης μέσω της λίστας `questionsList` (αντικείμενο `Question`). Η μεταβλητή `currentQuestionId` έχει τον αριθμό της τρέχουσας ερώτησης όπως έχει ειπωθεί. Σημειώνεται ότι η αρίθμηση στον προγραμματισμό ξεκινά από το 0, και άρα εάν η μεταβλητή ισούται με το 5, τότε βρισκόμαστε στην 6<sup>η</sup> ερώτηση. Θα μετατρέψουμε τη σωστή απάντηση του συστήματος σε κεφαλαία για το λόγο ότι θέλουμε να αποφύγουμε τη σύγκριση της ίδιας λέξης σε πεζά και κεφαλαία καθώς δεν πρόκειται για την ίδια λέξη στη μηχανή. Εφόσον η ερώτηση είναι συμπλήρωσης κενού, θα μετατρέψουμε την απάντηση του χρήστη σε κεφαλαία εξίσου. Θα γίνει περεταίρω επεξεργασία σε αυτόν τον τύπο της ερώτησης με την αφαίρεση των διακριτικών (χαρακτήρων) της ελληνικής γλώσσας, όπως τόνους, διαλυτικά κλπ. και θεωρούνται στοιχεία διαλέκτου (accent) καθώς η ίδια λέξη με τόνο και χωρίς δε θεωρείται η ίδια στη μηχανή. Επομένως, θέλουμε να αποφύγουμε αυτή τη σύγκριση λέξης για την επαλήθευση της απάντησης μεταξύ συστήματος και μαθητή καθώς πρόκειται για την ίδια λέξη. Βοηθός σε αυτή την επεξεργασία είναι η μέθοδος `RemoveDiacritics`:

```
static string RemoveDiacritics(string text)
{
    if (text == null || text == "")
    {
        return null;
    }
    var normalizedString = text.Normalize(NormalizationForm.FormD);
    var stringBuilder = new StringBuilder(capacity: normalizedString.Length);
    for (int i = 0; i < normalizedString.Length; i++)
    {
        char c = normalizedString[i];
        var unicodeCategory = CharUnicodeInfo.GetUnicodeCategory(c);
        if (unicodeCategory != UnicodeCategory.NonSpacingMark)
        {
            stringBuilder.Append(c);
        }
    }
    return stringBuilder
        .ToString()
        .Normalize(NormalizationForm.FormC);
}
```

Αυτή η μέθοδος θα δημιουργήσει ένα κενό αντικείμενο `StringBuilder` και μέσω ενός βρόγχου θα γίνει επανάληψη κάθε χαρακτήρα της λέξης που δίνεται ως όρισμα στη μέθοδο, θα λάβουμε την κατηγορία `Unicode` που ανήκει ο εκάστοτε χαρακτήρας και εάν αυτός είναι γράμμα και όχι άλλος χαρακτήρας στην ανθρώπινη γλώσσα θα προσθέσει στο αντικείμενο `StringBuilder` το χαρακτήρα και τελικά θα επιστραφεί το διαμορφωμένο και κανονικοποιημένο πλέον `string` που δημιουργήθηκε.

Επιστρέφουμε στη μέθοδο `ValidateUserAnswer`, και εάν η ερώτηση είναι τύπου πολλαπλής επιλογής ή σωστού/λάθους θα θέσουμε τη μεταβλητή `userAnswer` ίση με την επιλεγμένη τιμή του `DropDown` στοιχείου στην πρώτη περίπτωση, ενώ θα ισούται με τη μεταβλητή `userSelection` (περιγράψαμε πιο

πάνω) στη 2<sup>η</sup> περίπτωση. Στην ερώτηση αντιστοίχισης θα λάβουμε την τιμή του Dropdown του αριστερού στοιχείου ελέγχου και την τιμή (έχοντας μετατραπεί σε κεφαλαία) το δεξιού στοιχείου ελέγχου χωρισμένα με παύλα και η ίδια διαδικασία θα γίνει για τα υπόλοιπα 3 στοιχεία ελέγχου και κάθε γραμμή αντιστοίχισης στοιχείων θα χωρίζεται με κόμμα καθώς σε αυτή τη μορφή είναι αποθηκευμένη η απάντηση στη βάση (πχ. 1-b,2-c,3-a,4-d). Τέλος με τη συνθήκη ισότητας απάντησης χρήστη και απάντηση συστήματος θα μας πει εάν ο μαθητής έδωσε τη σωστή απάντηση και θα αυξήσουμε τη μεταβλητή κατά μία μονάδα correctQuestions, η οποία καταμετρά τις σωστές απαντήσεις του μαθητή. Το σώμα της:

```

public void ValidateUserAnswer()
{
    var correctAnswer =
questionsList[currentQuestionId].getCorrectAnswer().ToUpper();
    var userAnswer = "";
    if (questionsList[currentQuestionId].getType().Equals("fill_empty"))
    {
        userAnswer = userAnswerTxt.Text.ToUpper();
    }
    else if
(questionsList[currentQuestionId].getType().Equals("correct_wrong"))
    {
        userAnswer = userSelection.ToString();
    }
    else if (questionsList[currentQuestionId].getType().Equals("map"))
    {
        userAnswer = userMapLeft1.Text + "-" + userMapRight1.Text.ToUpper() +
", "
        + userMapLeft2.Text + "-" + userMapRight2.Text.ToUpper() + ", "
        + userMapLeft3.Text + "-" + userMapRight3.Text.ToUpper() + ", "
        + userMapLeft4.Text + "-" + userMapRight4.Text.ToUpper();
    }
    else if (questionsList[currentQuestionId].getType().Equals("choice"))
    {
        userAnswer = userChoice.ToString();
    }
    if (questionsList[currentQuestionId].getType().Equals("fill_empty") ==
true)
    {
        correctAnswer = RemoveDiacritics(correctAnswer);
        userAnswer = RemoveDiacritics(userAnswer);
    }
    if (correctAnswer.Equals(userAnswer))
    {
        correctQuestions++;
    }
    //Debug.WriteLine("current question: " + currentQuestionId + " user: " +
userAnswer + " | correct: " + correctAnswer + " | result: " +
correctAnswer.Equals(userAnswer));
}

```

Επιστρέφουμε στη μέθοδο GoToNextQuestion και αυξάνουμε τον αριθμό (currentQuestionId) της ενεργής κατά 1 μονάδα (ξεκινά με τιμή -1). Εν συνεχεία θα αποκρύψουμε όλα τα στοιχεία ελέγχου που αφορούν είσοδο απάντησης μαθητή με την παρακάτω μέθοδο HideElements:

```

void HideElements()
{
    correctAnswerLbl.Visible = false;
    userAnswerTxt.Visible = false;
    userCorrectSelectionBtn.Visible = false;
    userWrongSelectionBtn.Visible = false;
    userMapLeft1.Visible = false; userMapRight1.Visible = false;
    userMapLeft2.Visible = false; userMapRight2.Visible = false;
    userMapLeft3.Visible = false; userMapRight3.Visible = false;
}

```

```

        userMapLeft4.Visible = false; userMapRight4.Visible = false;
        userChoice.Visible = false;
    }

```

Θα επαναφέρουμε επίσης τα στοιχεία ελέγχου για τις απαντήσεις του μαθητή στις αρχικές τους καταστάσεις καθώς δε θέλουμε να μεταφέρουμε τις απαντήσεις του μαθητή σε κάθε επόμενη ερώτηση. Το σώμα της μεθόδου ResetElements φαίνεται παρακάτω:

```

void ResetElements()
{
    userAnswerTxt.Text = "";
    userChoice.SelectedIndex = -1;
    userSelection = -1;
    userMapLeft1.SelectedIndex = -1; userMapRight1.SelectedIndex = -1;
    userMapLeft2.SelectedIndex = -1; userMapRight2.SelectedIndex = -1;
    userMapLeft3.SelectedIndex = -1; userMapRight3.SelectedIndex = -1;
    userMapLeft4.SelectedIndex = -1; userMapRight4.SelectedIndex = -1;
}

```

Επιστρέφουμε πάλι στην κύρια μέθοδο θα εμφανίσουμε μόνο τα στοιχεία ελέγχου της απάντησης τα οποία είναι απαραίτητα ανάλογα με τον τύπο ερώτησης. Να θυμηθούμε ότι τα είχαμε αποκρύψει όλα αρχικά. Επίσης για την ερώτηση αντιστοίχισης θα προ-συμπληρώσουμε την αριστερή με τις επιλογές: a, b, c, d προς διευκόλυνση του χρήστη επιτρέποντας να συμπληρώσουμε τα στοιχεία της δεξιάς στήλης.

Όταν ο μαθητής φτάσει στην τελευταία ερώτηση (ο αριθμός της ενεργής ερώτησης ισούται με το πλήθος των ερωτήσεων μειωμένος κατά 1) τότε το κουμπί Επόμενη ερώτηση θα έχει το κείμενο Ολοκλήρωση και κατά την ολοκλήρωση της εξέτασης θα λάβουμε το μισό του συνολικού πλήθους των ερωτήσεων και εάν οι σωστές απαντήσεις του μαθητή είναι ίσες ή περισσότερες από την τιμή του μισού πλήθους τότε θεωρούμε ότι πέρασε την εξέταση (pass). Fail ορίζουμε όταν ο μαθητής δεν περάσει την εξέταση. Θα καταγράψουμε την ολοκλήρωση της εξέτασης με τη μέθοδο EndTest (αντικείμενο DB) και θα μεταφέρουμε το χρήστη στη φόρμα των αποτελεσμάτων (ResultForm) δίνοντας ως ορίσματα το id του χρήστη, το username και το id της εξέτασης. Το σώμα της μεθόδου που περιγράψαμε εκτεταμένα είναι παρακάτω:

```

void GoToNextQuestion(object sender, EventArgs e, bool checkQuestion)
{
    //Console.WriteLine("questions cnt: "+questionsList.Count);
    //Console.WriteLine("current question: " + currentQuestionId);
    bool proceed = false;
    if (questionsList.Count > currentQuestionId)
    {
        if (checkQuestion == false)
        {
            testId = DB.StartTest(dbCon, userId, sectionId, mode.ToString(),
questionsList.Count);
        }
        else
        {
            proceed = CanProceedToNextQuestion();
            //Console.WriteLine("can proceed: " + (proceed == true ? "YES" :
"NO"));
            if (proceed == true)
            {

```

```

        ValidateUserAnswer();
    }
    else
    {
        MessageBox.Show("Παρακαλώ δώστε την απάντησή σας",
Constants.appTitle);
        return;
    }
}

currentQuestionId++;

HideElements();
ResetElements();

if (questionsList.ElementAtOrDefault(currentQuestionId) != null)
{
    questionHeader.Text = "Ερώτηση " + (currentQuestionId + 1) + "
από " + questionsList.Count;

    if
(questionsList[currentQuestionId].GetType().Equals("fill_empty"))
    {
        userAnswerTxt.Visible = true;
    }
    else if
(questionsList[currentQuestionId].GetType().Equals("correct_wrong"))
    {
        userCorrectSelectionBtn.Visible = true;
        userWrongSelectionBtn.Visible = true;
    }
    else if
(questionsList[currentQuestionId].GetType().Equals("map"))
    {
        userMapLeft1.Visible = true; userMapRight1.Visible = true;
        userMapLeft2.Visible = true; userMapRight2.Visible = true;
        userMapLeft3.Visible = true; userMapRight3.Visible = true;
        userMapLeft4.Visible = true; userMapRight4.Visible = true;

        // FILL LEFT ITEMS
        userMapLeft1.SelectedIndex = 0;
        userMapLeft2.SelectedIndex = 1;
        userMapLeft3.SelectedIndex = 2;
        userMapLeft4.SelectedIndex = 3;
    }
    else if
(questionsList[currentQuestionId].GetType().Equals("choice"))
    {
        userChoice.Visible = true;
    }

    questionTitle.Text = questionsList[currentQuestionId].getTitle();
}
}

// FINAL QUESTION
if ((questionsList.Count - 1) == currentQuestionId)
{
    nextQuestionBtn.Text = "Ολοκλήρωση";
}

// COMPLETED
if (questionsList.Count == currentQuestionId)
{
    //MessageBox.Show("test", Constants.appTitle);
    int half = (questionsList.Count / 2);
    string result = correctQuestions >= half ? "pass" : "fail";
    DB.EndTest(dbCon, testId, questionsAnswered, correctQuestions,
result);

    this.Hide();
    var form = new ResultForm(userId, username, testId);
    form.Closed += (s, args) => this.Close();
    form.Show();
}
}
}

```

Η τελευταία μέθοδος που θα δούμε είναι `showCorrectAnswerBtnClick`, η οποία εκτελείται όταν ο μαθητής βρίσκεται σε εξάσκηση και πατάει το κουμπί Προβολή σωστού. Θα λάβουμε τη σωστή απάντηση της ενεργής ερώτησης. Θα γίνει επεξεργασία εφόσον η απάντηση είναι σωστού/λάθους μετατρέποντας το 0 σε Λάθος και το 1 σε Σωστό. Όταν η ερώτηση είναι πολλαπλής επιλογής τότε θα αντιστοιχίσουμε τα 1,2,3,4 όπως βρίσκονται αποθηκευμένα στη βάση σε a,b,c,d αντίστοιχα. Επίσης θα εμφανίσουμε το κεφάλαιο στο οποίο ανήκει η ερώτηση για να βοηθήσουμε το μαθητή να ανατρέξει στην ύλη για να μάθει τη σωστή απάντηση της ερώτησης. Το σώμα της είναι παρακάτω:

```
void showCorrectAnswerBtnClick(object sender, EventArgs e)
{
    var correct = questionsList[currentQuestionId].getCorrectAnswer();
    //Debug.WriteLine(questionsList[currentQuestionId].getType() + " |
"+correct);
    if (questionsList[currentQuestionId].getType().Equals("correct_wrong"))
    {
        correct = correct.Equals("0") ? "Λάθος" : "Σωστό";
    }
    else if (questionsList[currentQuestionId].getType().Equals("choice"))
    {
        if (correct.Equals("1"))
        {
            correct = "a";
        }
        else if (correct.Equals("2"))
        {
            correct = "b";
        }
        else if (correct.Equals("3"))
        {
            correct = "b";
        }
        else if (correct.Equals("4"))
        {
            correct = "b";
        }
    }
    correct += "\nΚεφάλαιο " +
questionsList[currentQuestionId].getChapterId();
    correctAnswerLbl.Text = correct;
    correctAnswerLbl.Visible = true;
}
```

Επίσης έχουμε γράψει τη μέθοδο `randomAnswerBtnClick`, η οποία δε χρειάζεται ανάλυση καθώς συμπληρώνει αυτόματα με τυχαίο τρόπο τη σωστή απάντηση σε κάθε ερώτηση. Δημιουργήθηκε για λόγους αποσφαλμάτωσης (debugging) της εφαρμογής.

### 3.7 Φόρμα αποτελεσμάτων (ResultForm.cs)

Η φόρμα είναι υπεύθυνη για να πληροφορήσει το χρήστη για το πώς τα πήγε στην εξέταση. Θα εμφανίσουμε τις εξής πληροφορίες σε μορφή ετικετών (Label):

1. Τύπος εξέτασης (διαγώνισμα ή εξάσκηση)
2. Ενότητα

3. Συνολικές ερωτήσεις
4. Απαντημένες ερωτήσεις μαθητή
5. Σωστές ερωτήσεις μαθητή
6. Βαθμός

Θα γίνει εκτέλεση της μεθόδου `getTest`, η οποία θα καλέσει τη μέθοδο `GetTestById` (αντικείμενο `DB`) για να λάβει την εξέταση που διενεργήθηκε μόλις. Θα προβληθεί απλά το περιεχόμενο που έχει αποθηκευτεί στον πίνακα `tests` στις αντίστοιχες ετικέτες που είδαμε παραπάνω. Στην περίπτωση που δε βρεθεί εξέταση με το `id` της, θα εμφανίσουμε το μήνυμα στη φόρμα: «**Δε βρέθηκαν tests με id: #{ID}**» όπου `{ID}`, το `id` της εξέτασης. Το σώμα της μεθόδου προβάλλεται παρακάτω:

```

void getTest(long testId)
{
    var db_read = DB.GetTestById(dbCon, testId);
    if (db_read.HasRows)
    {
        while (db_read.Read())
        {
            //Console.WriteLine("mode: " + db_read["mode"]);
            typeLabel.Text += (bool)db_read["mode"] == false ? " Εξάσκηση" :
" Διαγώνισμα";

            sectionLabel.Text += " " + (int)db_read["section_id"];
            //progressLabel.Text += " " + progress + "/" + totalSections;
            totalQuestionsLabel.Text += " " +
(int)db_read["total_questions"];
            answeredQuestionsLabel.Text += " " +
(int)db_read["questions_answered"];
            correctQuestionsLabel.Text += " " +
(int)db_read["correct_questions"];
            string result = (string)db_read["result"];
            string grade = "Κοπήκατε";
            if (result.Equals("pass") == true)
            {
                grade = "Περάσατε";
            }
            resultLabel.Text += " " + db_read["correct_questions"] + "/" +
db_read["total_questions"] + " (" + grade + ")";
        }
        db_read.Close();
    }
    else
    {
        foreach (Control c in Controls)
        {
            c.Hide();
        }
        db_read.Close();
        resultsHeader.Text = "Δε βρέθηκαν tests με id: #" + testId;
        backBtn.Visible = true;
        resultsHeader.Visible = true;
    }
}

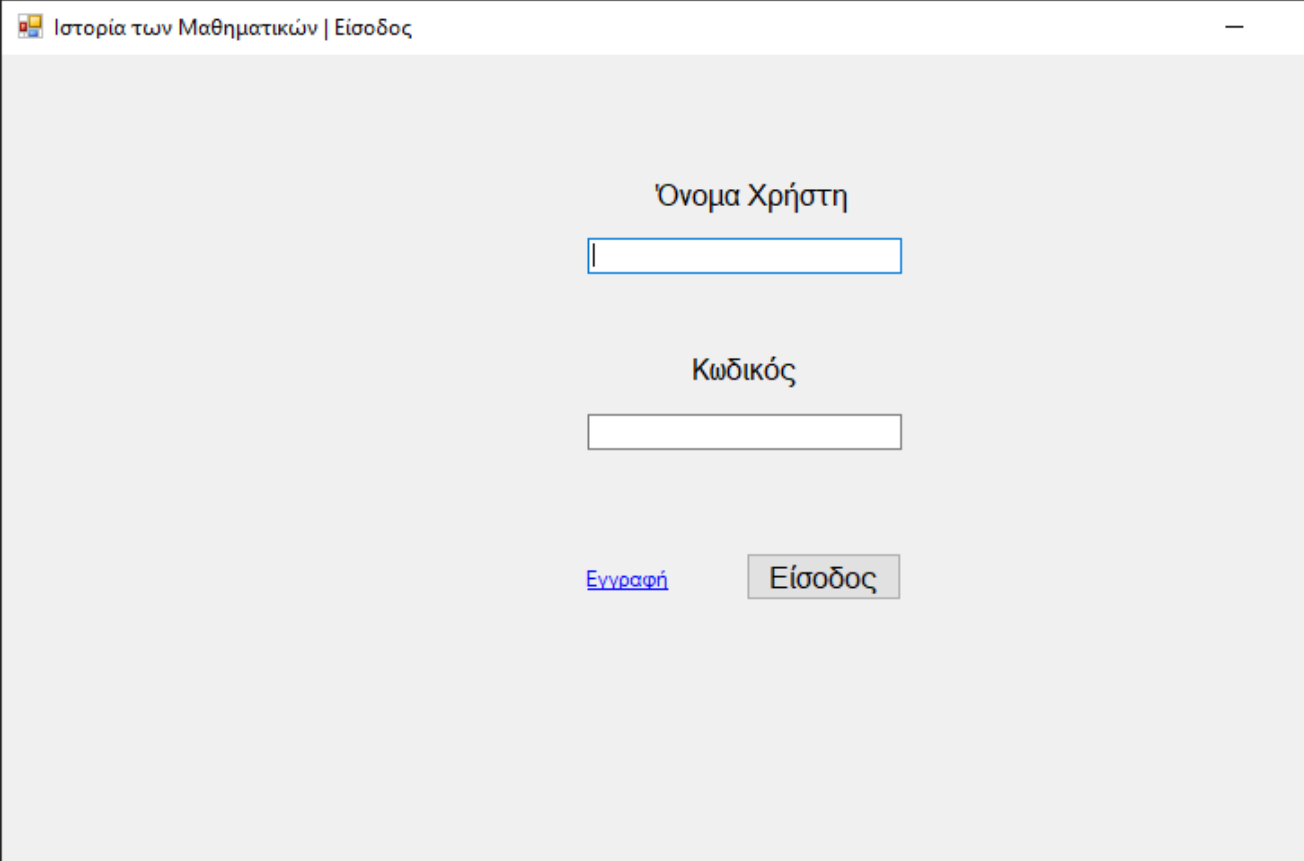
```

### 3.8 Δημιουργία εγκατάστασης και φιλοξενία βάσης δεδομένων

Σε αυτό το σημείο ολοκληρώθηκε η τεχνική τεκμηρίωση του κώδικα της εφαρμογής. Μπορούμε να δημιουργήσουμε εγκατάσταση της εφαρμογής για την εκτέλεσή της στα μηχανήματα των χρηστών. Γι' αυτό το σκοπό μπορούμε να χρησιμοποιήσουμε τη δωρεάν έκδοση της εφαρμογής

Advanced Installer (<https://www.advancedinstaller.com/>), η οποία προσφέρει έναν εύκολο τρόπο αλλά με αμέτρητες επιλογές και δυνατότητες για τη δημιουργία εγκαταστάτη μιας εφαρμογής που έχει δημιουργηθεί σε Visual Studio. Για την εκτέλεση της εφαρμογής είναι απαραίτητο να προηγηθεί η μεταφορά της υφιστάμενης βάσης δεδομένων σε server (διακομιστή Ιστού) που να επιτρέπει την απομακρυσμένη διασύνδεση MySQL και να ενημερώσουμε το αρχείο DB.cs με τα στοιχεία σύνδεσης σε αυτή. Εμείς επιλέξαμε τη μεταφορά της βάσης στο Heroku (<https://id.heroku.com/>), μια πλατφόρμα που επιτρέπει τη δωρεάν φιλοξενία εκτέλεση εφαρμογών με περιορισμένους πόρους. Οι εμπορικές εκδόσεις του Heroku όμως παρέχουν πολλές και ενδιαφέρουσες δυνατότητες. Εμείς χρησιμοποιήσαμε τον πόρο Jaws Maria DB, ο οποίος προσφέρει ένα «σπίτι» για τη βάση δεδομένων και επιτρέπει την απομακρυσμένη σύνδεση. Μετά τη δημιουργία του πόρου, μεταφέραμε τη βάση δεδομένων εκεί και ενημερώσαμε τα στοιχεία σύνδεσης στο αρχείο Constants.cs

## Κεφάλαιο 4: Στιγμιότυπα εκτέλεσης εφαρμογής



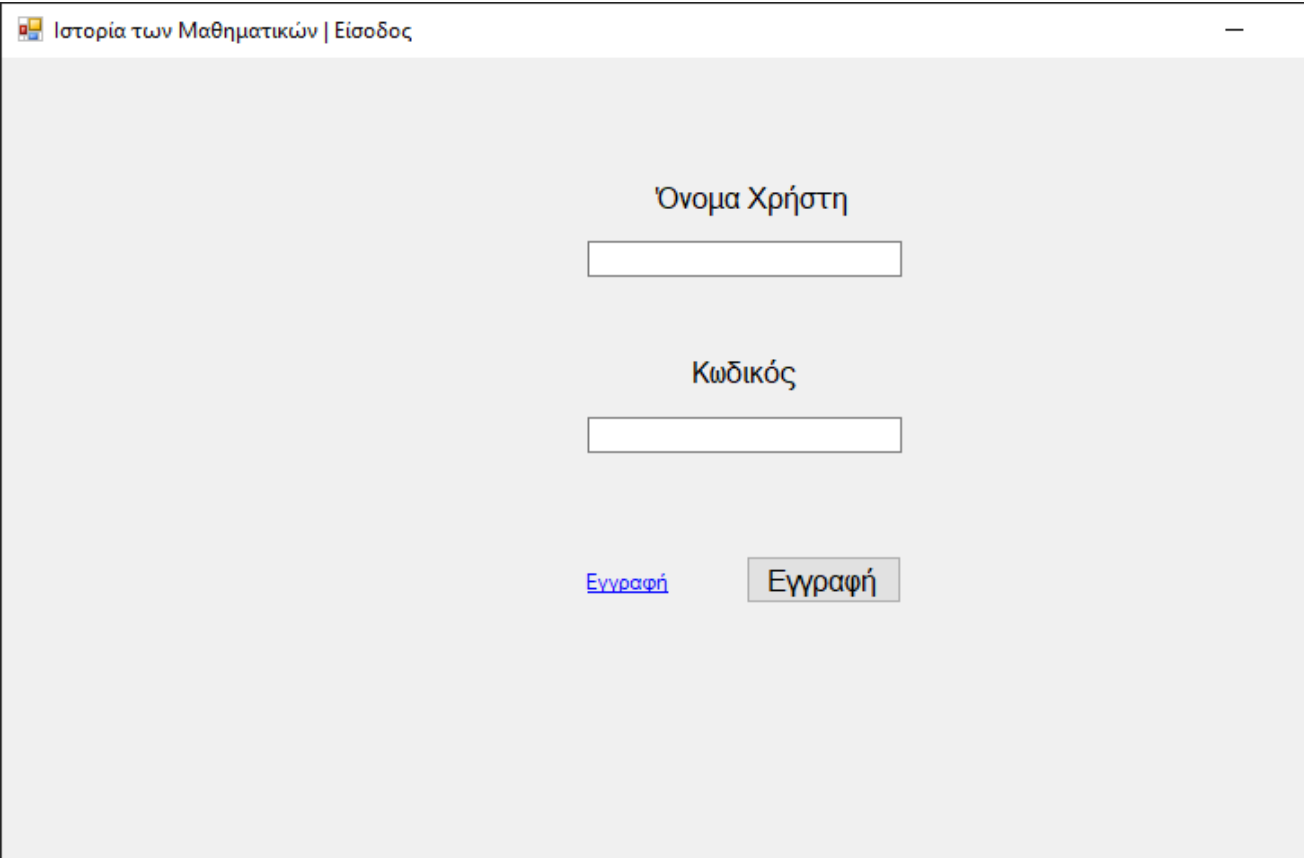
Ιστορία των Μαθηματικών | Είσοδος

Όνομα Χρήστη

Κωδικός

[Εγγραφή](#)

Εικόνα 9 Αρχική φόρμα: Ο χρήστης να καλείται να εισάγει τα στοιχεία εισόδου του



Ιστορία των Μαθηματικών | Είσοδος

Όνομα Χρήστη

Κωδικός

[Εγγραφή](#)

Εικόνα 10 Ο χρήστης πάτησε το κουμπί Εγγραφή



**Κεφάλαια**

Κεφάλαιο	Ενότητα	Τίτλος	Τύπος	Υλικό
1	1	Εισαγωγή στην ιστορία των μαθηματικών	Έγγραφο	<a href="#">Προβολή</a>
2	1	Τα Μαθηματικά των αρχαίων Αιγυπτίων, συστήματα αρίθμησης	Έγγραφο	<a href="#">Προβολή</a>
3	1	Τα Μαθηματικών των Βαβυλωνίων	Έγγραφο	<a href="#">Προβολή</a>
4	1	Τι είναι απόδειξη?	Έγγραφο	<a href="#">Προβολή</a>
5	2	Θαλής, Πυθαγόρας, Πλατωνικά Στερεά, άρρητα μεγέθη, παράδοξα του Ζήνωνα, τα περίφημα γε...	Έγγραφο	<a href="#">Προβολή</a>
6	2	Εύδοξος	Έγγραφο	<a href="#">Προβολή</a>
7	2	Τα Στοιχεία του Ευκλείδη, το πέμπτο αίτημα και οι μη Ευκλείδειες γεωμετρίες	Έγγραφο	<a href="#">Προβολή</a>
8	2	Αρχιμήδης: Ψαμμίτης, το βοσεικό πρόβλημα και το εμβαδόν ανάμεσα στη παραβολή και τη τέμνο...	Έγγραφο	<a href="#">Προβολή</a>
9	2	Αρχιμήδης: Η μέθοδος: κύβος και κώνυρος, Διάφαντος	Έγγραφο	<a href="#">Προβολή</a>
10	3	Τι είναι Άλγεβρα? Δευτεροβάθμια εξίσωση από το 2000 π.Χ. -250 μ.Χ.	Έγγραφο	<a href="#">Προβολή</a>
11	3	Αραβικά μαθηματικά. Liber abaci. Ιταλοί και τριτοβάθμια εξίσωση	Έγγραφο	<a href="#">Προβολή</a>
12	3	Cardano και λύση της τριτοβάθμιας	Έγγραφο	<a href="#">Προβολή</a>
13	3	Τεταρτοβάθμια, Bombelli, Viete: οι μαγικοί αριθμοί είναι χρήσιμοι	Έγγραφο	<a href="#">Προβολή</a>
14	3	Ο καιρός των λογαρίθμων	Έγγραφο	<a href="#">Προβολή</a>

**Στατιστικά**

Αναγνώσεις κεφαλαίων: 55  
Μέσος χρόνος ανάγνωσης: 3

Εξασκήσεις: 1

Διαγωνίσματα: 0

Τελευταίος βαθμός:

**Ενότητες**

Εικόνα 11 Έγινε είσοδος του χρήστη user1 και μεταφέρεται στην κύρια φόρμα της εφαρμογής

**Κεφάλαια**

Κεφάλαιο	Ενότητα	Τίτλος	Τύπος	Υλικό
1	1	Εισαγωγή στην ιστορία των μαθηματικών	Έγγραφο	<a href="#">Προβολή</a>
2	1	Τα Μαθηματικά των αρχαίων Αιγυπτίων, συστήματα αρίθμησης	Έγγραφο	<a href="#">Προβολή</a>
3	1	Τα Μαθηματικών των Βαβυλωνίων	Έγγραφο	<a href="#">Προβολή</a>
4	1	Τι είναι απόδειξη?	Έγγραφο	<a href="#">Προβολή</a>
5	2	Θαλής, Πυθαγόρας, Πλατωνικά Στερεά, άρρητα μεγέθη, παράδοξα του Ζήνωνα, τα περίφημα γε...	Έγγραφο	<a href="#">Προβολή</a>
6	2	Εύδοξος	Έγγραφο	<a href="#">Προβολή</a>
7	2	Τα Στοιχεία του Ευκλείδη, το πέμπτο αίτημα και οι μη Ευκλείδειες γεωμετρίες	Έγγραφο	<a href="#">Προβολή</a>
8	2	Αρχιμήδης: Ψαμμίτης, το βοσεικό πρόβλημα και το εμβαδόν ανάμεσα στη παραβολή και τη τέμνο...	Έγγραφο	<a href="#">Προβολή</a>
9	2	Αρχιμήδης: Η μέθοδος: κύβος και κώνυρος, Διάφαντος	Έγγραφο	<a href="#">Προβολή</a>
10	3	Τι είναι Άλγεβρα? Δευτεροβάθμια εξίσωση από το 2000 π.Χ. -250 μ.Χ.	Έγγραφο	<a href="#">Προβολή</a>
11	3	Αραβικά μαθηματικά. Liber abaci. Ιταλοί και τριτοβάθμια εξίσωση	Έγγραφο	<a href="#">Προβολή</a>
12	3	Cardano και λύση της τριτοβάθμιας	Έγγραφο	<a href="#">Προβολή</a>
13	3	Τεταρτοβάθμια, Bombelli, Viete: οι μαγικοί αριθμοί είναι χρήσιμοι	Έγγραφο	<a href="#">Προβολή</a>
14	3	Ο καιρός των λογαρίθμων	Έγγραφο	<a href="#">Προβολή</a>

**Στατιστικά**

Αναγνώσεις κεφαλαίων: 55  
Μέσος χρόνος ανάγνωσης: 3

Εξασκήσεις: 1

Διαγωνίσματα: 0

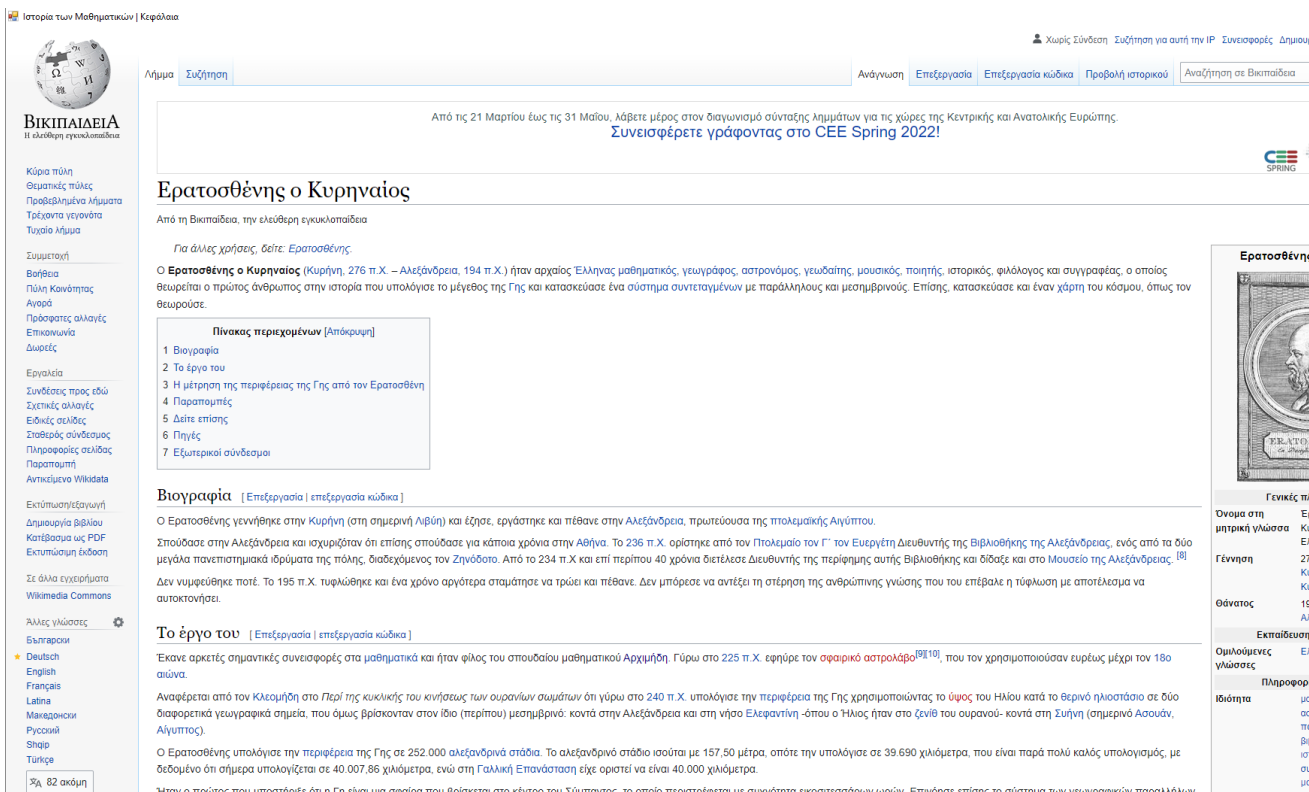
Τελευταίος βαθμός:

**Ενότητες**

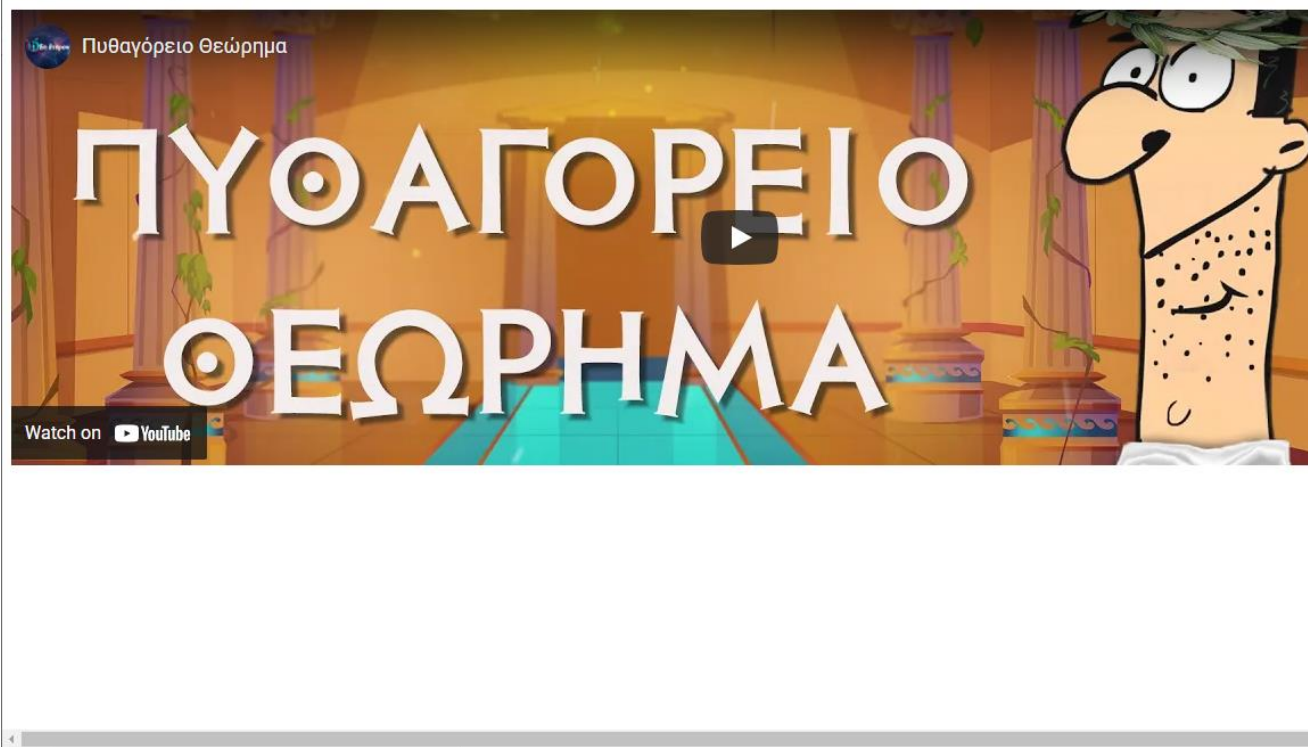
Εικόνα 12 Ο χρήστης μπορεί να επιλέξει κάτω αριστερά στην οποία θα κάνει εξάσκηση ή διαγώνισμα



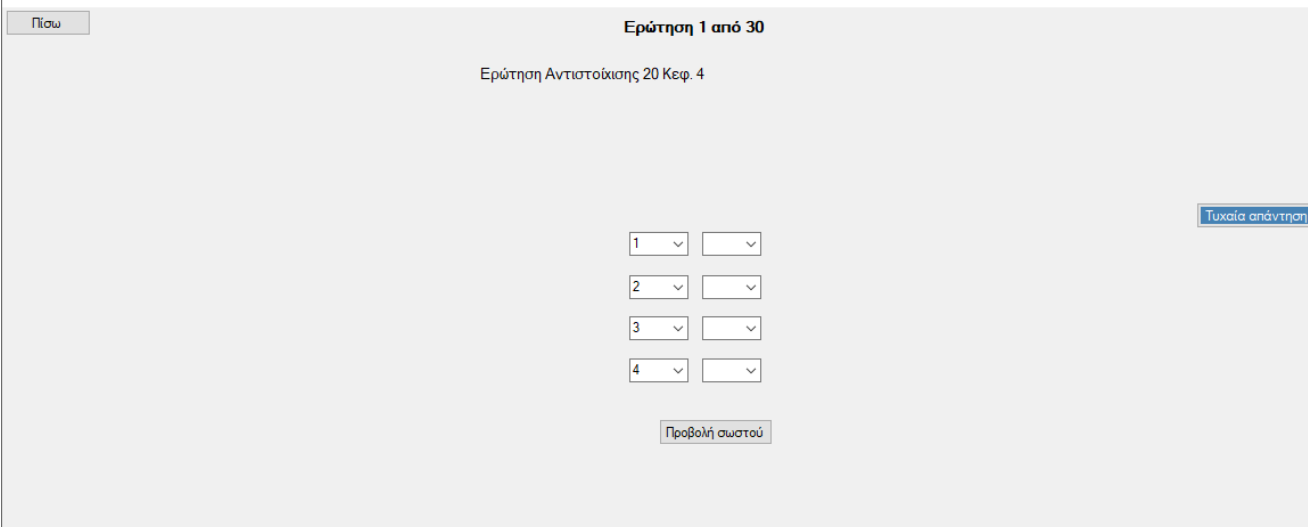
Εικόνα 13 Ο χρήστης επέλεξε να προβάλλει τοπικό αρχείο PDF για ανάγνωση



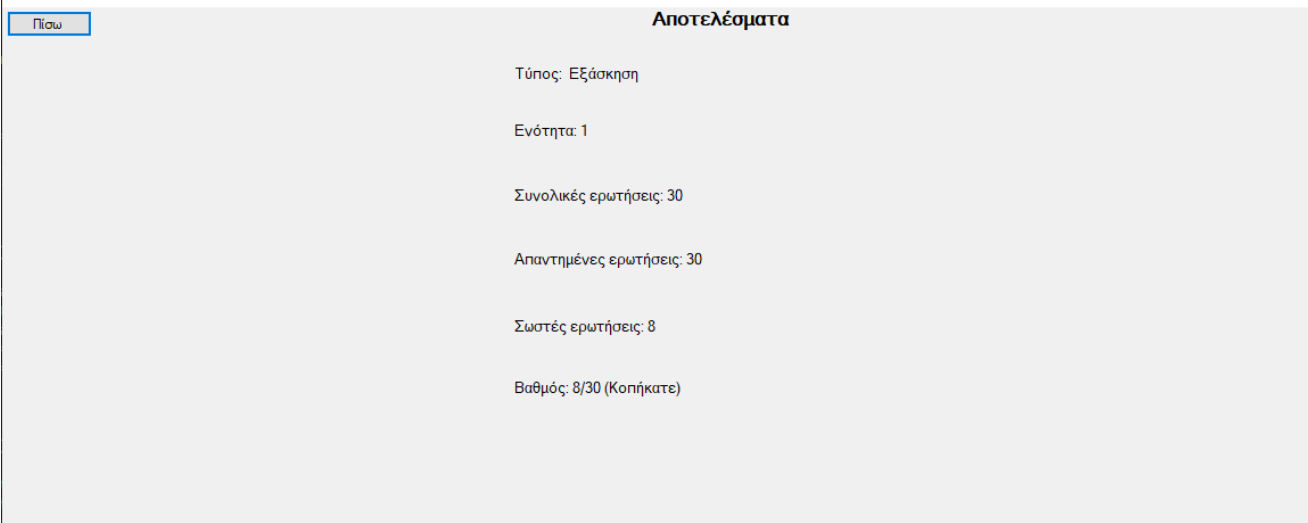
Εικόνα 14 Ο χρήστης επέλεξε να προβάλλει διαδικτυακό πόρο

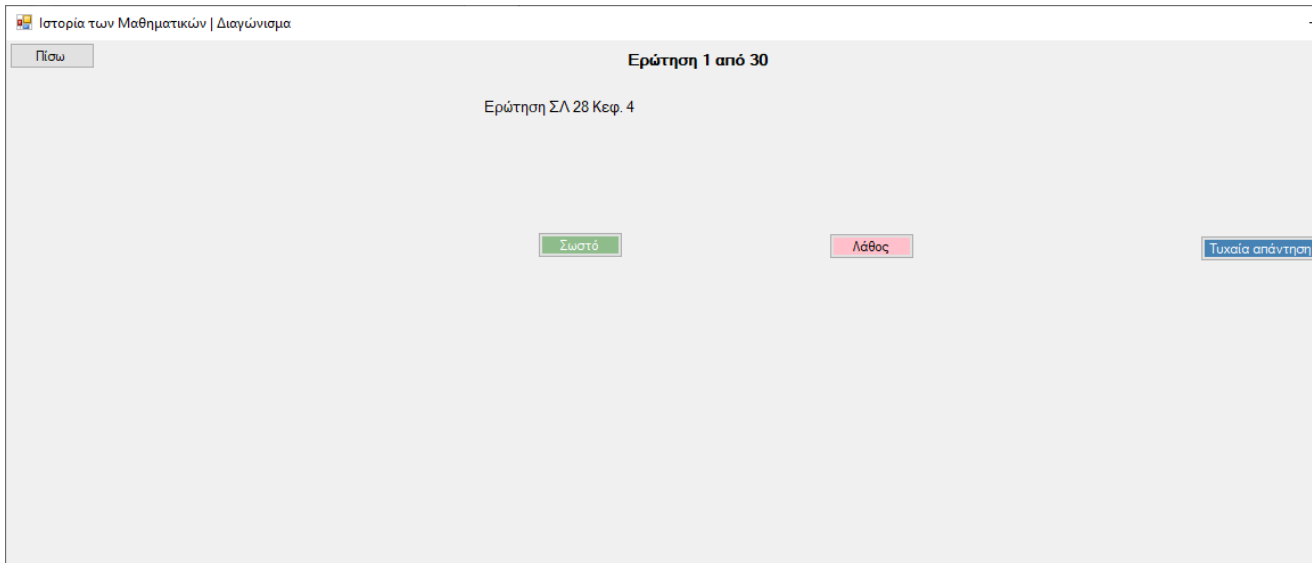


Εικόνα 15 Ο χρήστης επέλεξε να προβάλει διαδικτυακό βίντεο



Εικόνα 16 Ο χρήστης ξεκίνησε μία εξάσκηση





Εικόνα 18 Ο χρήστης ξεκίνησε ένα διαγώνισμα

## Κεφάλαιο 5: Εγχειρίδιο χρήστη της εφαρμογής «Ιστορία των Μαθηματικών»

Η εφαρμογή «Ιστορία των Μαθηματικών» είναι μια πλατφόρμα για εξ αποστάσεως εκπαίδευση.

Η εφαρμογή παρέχει στον χρήστη το υλικό το οποίο είναι απαραίτητο για την μελέτη του μαθήματος της ιστορίας των μαθηματικών καθώς και τη δυνατότητα να εξασκηθεί σε ερωτήσεις για την κάθε ενότητα, και

αργότερα σε διαγωνίσματα. Ακόμη παρέχει πληροφορίες σχετικά με την επίδοση του και το χρόνο που αφιέρωσε διαβάζοντας το κάθε κεφάλαιο (στατιστικά).

# Εγγραφή/Είσοδος

Ιστορία των Μαθηματικών | Είσοδος

Όνομα Χρήστη

Κωδικός

[Εγγραφή](#)

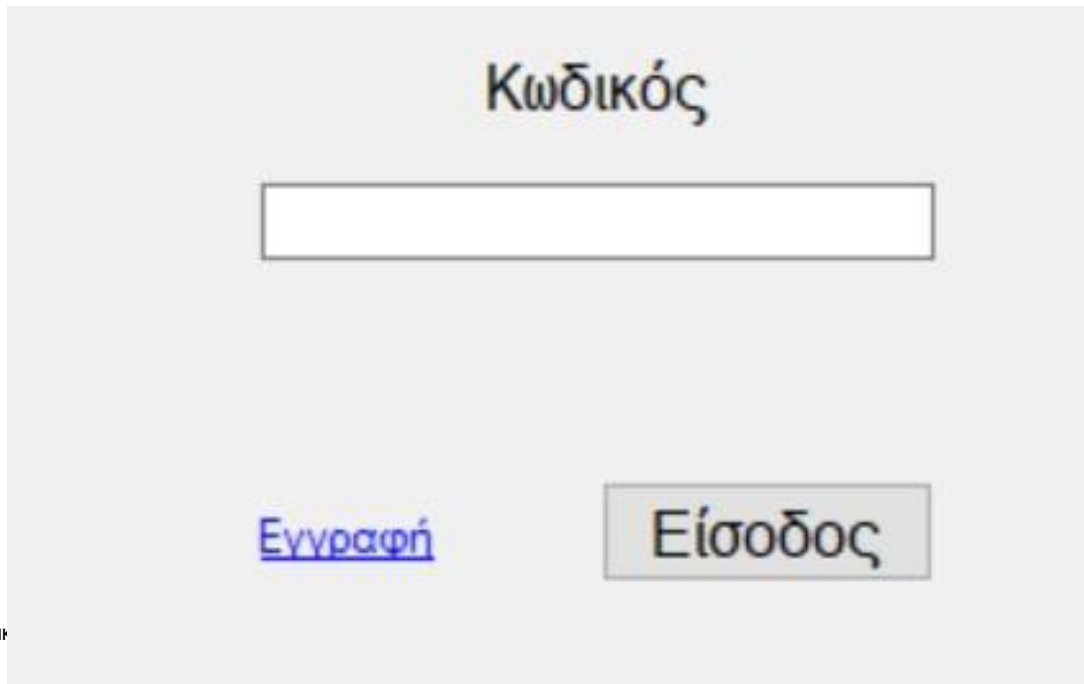
Είσοδος

Πληκτρολογούμε στο πρώτο κουτί το Όνομα Χρήστη.

Αντίστοιχα στο δεύτερο κουτί πληκτρολογούμε τον

κωδικό μας.

# Εγγραφή/Είσοδος



Κωδικός

[Εγγραφή](#)

Εάν δεν είμαστε εγγεγραμμένοι χρήστες, αφού πληκτρολογήσουμε τα στοιχεία μας (Όνομα Χρήστη, Κωδικό) πατάμε το πλήκτρο «Εγγραφή».

Μετά την εγγραφή μας (ή στη



περίπτωση που είμαστε ήδη  
εγγεγραμμένοι) πληκτρολογούμε τα  
στοιχεία μας και πατάμε το πλήκτρο  
«Είσοδος».

Αφού συνδεθούμε, εισερχόμαστε στην κύρια σελίδα της εφαρμογής στην οποία βρίσκεται πίνακας, ο οποίος περιέχει τα αρχεία προς μελέτη, κατανεμημένα κατά κεφάλαιο, ενότητα, τίτλο και τύπο αρχείου ενώ στα δεξιά του πίνακα καταγράφονται διάφορα στατιστικά του φοιτητή και τέλος κάτω από τον πίνακα μας δίνεται η δυνατότητα να εξασκηθούμε σε όποια ενότητα επιλέξουμε.

## Κεφάλαια

Καλησπέρα Χρήστης

	Κεφάλαιο	Ενότητα	Τίτλος	Τύπος	Υλικό
▶	1	1	Εισαγωγή στην ιστορία των μαθηματικών	Έγγραφο	<a href="#">Προβολή</a>
	2	1	Τα Μαθηματικά των αρχαίων Αιγυπτίων, συστήματα αρίθμησης	Έγγραφο	<a href="#">Προβολή</a>
	3	1	Τα Μαθηματικών των Βαβυλωνίων	Έγγραφο	<a href="#">Προβολή</a>
	4	1	Τι είναι απόδειξη?	Έγγραφο	<a href="#">Προβολή</a>
	5	2	Θαλής, Πυθαγόρας, Πλατωνικά Στερεά, άρρητα μεγέθη, παράδοξα του Ζήνωνα, τα περίφημα γε...	Έγγραφο	<a href="#">Προβολή</a>
	6	2	Εύδοξος	Έγγραφο	<a href="#">Προβολή</a>
	7	2	Τα Στοιχεία του Ευκλείδη, το πέμπτο αίτημα και οι μη Ευκλείδειες γεωμετρίες	Έγγραφο	<a href="#">Προβολή</a>
	8	2	Αρχιμήδης: Ψαμμίτης, το βοεικό πρόβλημα και το εμβαδόν ανάμεσα στη παραβολή και τη τέμνο...	Έγγραφο	<a href="#">Προβολή</a>
	9	2	Αρχιμήδης: Η μέθοδος: κύβος και κώνιδρος, Διόφαντος	Έγγραφο	<a href="#">Προβολή</a>
	10	3	Τι είναι Άλγεβρα? Δευτεροβάθμια εξίσωση από το 2000 π.Χ. -250 μ.Χ.	Έγγραφο	<a href="#">Προβολή</a>
	11	3	Αραβικά μαθηματικά. Liber abaci. Ιταλοί και τριτοβάθμια εξίσωση	Έγγραφο	<a href="#">Προβολή</a>
	12	3	Cardano και λύση της τριτοβάθμιας	Έγγραφο	<a href="#">Προβολή</a>
	13	3	Τεταρτοβάθμια, Bombelli, Viète: οι μαγικοί αριθμοί είναι χρήσιμοι	Έγγραφο	<a href="#">Προβολή</a>
	14	3	Ο καιρός των λογαρίθμων	Έγγραφο	<a href="#">Προβολή</a>

## Στατιστικά

Αναγνώσεις κεφαλαίων: 0

Εξασκήσεις: 0

Διαγωνίσματα: 0

Τελευταίος βαθμός:

## Ενότητες

Εξάσκηση

Διαγώνισμα

Κεφάλαιο	Ενότητα	Τίτλος	Τύπος	Υλικό
1	1	Εισαγωγή στην ιστορία των μαθηματικών	Έγγραφο	<a href="#">Προβολή</a>
2	1	Τα Μαθηματικά των αρχαίων Αιγυπτίων, συστήματα αρίθμησης	Έγγραφο	<a href="#">Προβολή</a>
3	1	Τα Μαθηματικών των Βαβυλωνίων	Έγγραφο	<a href="#">Προβολή</a>
4	1	Τι είναι απόδειξη?	Έγγραφο	<a href="#">Προβολή</a>
5	2	Θαλής, Πυθαγόρας, Πλατωνικά Στερεά, άρρητα μεγέθη, παράδοξα του Ζήνωνα, τα περίφημα γε...	Έγγραφο	<a href="#">Προβολή</a>
6	2	Εύδοξος	Έγγραφο	<a href="#">Προβολή</a>
7	2	Τα Στοιχεία του Ευκλείδη, το πέμπτο αίτημα και οι μη Ευκλείδειες γεωμετρίες	Έγγραφο	<a href="#">Προβολή</a>
8	2	Αρχιμήδης: Ψαμμίτης, το βοεϊκό πρόβλημα και το εμβαδόν ανάμεσα στη παραβολή και τη τέμνο...	Έγγραφο	<a href="#">Προβολή</a>
9	2	Αρχιμήδης: Η μέθοδος: κύβος και κύλινδρος, Διόφαντος	Έγγραφο	<a href="#">Προβολή</a>
10	3	Τι είναι Άλγεβρα? Δευτεροβάθμια εξίσωση από το 2000 π.Χ. -250 μ.Χ.	Έγγραφο	<a href="#">Προβολή</a>
11	3	Αραβικά μαθηματικά. Liber abaci. Ιταλοί και τριτοβάθμια εξίσωση	Έγγραφο	<a href="#">Προβολή</a>
12	3	Cardano και λύση της τριτοβάθμιας	Έγγραφο	<a href="#">Προβολή</a>
13	3	Τεταρτοβάθμια, Bombelli, Viète: οι μιγαδικοί αριθμοί είναι χρήσιμοι	Έγγραφο	<a href="#">Προβολή</a>
14	3	Ο καιρός των λογαρίθμων	Έγγραφο	<a href="#">Προβολή</a>

Πατώντας το πλήκτρο «Προβολή» για οποιοδήποτε αρχείο τύπου «έγγραφο» μεταφερόμαστε στο αρχείο pdf του τίτλου που επιλέξαμε.

Κεφάλαιο	Ενότητα	Τίτλος	Τύπος	Υλικό
16	4	Στο δρόμο για το Θεμελιώδες Θεώρημα του Λογισμού: Newton και το Θεώρημα του Διωνύμου	Έγγραφο	<a href="#">Προβολή</a>
17	4	Newton και το Θεμελιώδες Θεώρημα του Λογισμού, Leibniz	Έγγραφο	<a href="#">Προβολή</a>
18	4	Σύγκριση των Λογισμών του Newton και Leibniz, Κάποια βασικά ερωτήματα στην (Κλασσική) Άλγ...	Έγγραφο	<a href="#">Προβολή</a>
19	5	Το Θεμελιώδες Θεώρημα της Άλγεβρας και απαντήσεις σε κάποια βασικά ερωτήματα της Κλασ...	Έγγραφο	<a href="#">Προβολή</a>
20	5	Peacock και συμβολική Άλγεβρα, Euler, Gauss. Στο δρόμο για την Αφηρημένη/Μοντέρνα Άλγεβρ...	Έγγραφο	<a href="#">Προβολή</a>
21	5	Μία σύντομη Εισαγωγή στην Ιστορία της Πιθανότητας (διάλεξη: Καθηγ. Π. Μωυσιάδης)	Έγγραφο	<a href="#">Προβολή</a>
22	5	Αφηρημένη Άλγεβρα: Θεωρία Ομάδων στο έργο των Gauss, Galois, Μη αντιμεταθετικοί δακτύλι...	Έγγραφο	<a href="#">Προβολή</a>
1000	6	Ερατοσθένης ο Κυρηναίος	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1007	6	Πυθαγόρας	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1008	6	Το πείραμα του Ερατοσθένη	Διαδικτυακό βίντεο	<a href="#">Προβολή</a>
1009	7	Ευκλείδης	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1010	7	Αρχιμήδης	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1011	7	Παράδοξα του Ζήνωνα	Διαδικτυακός πόρος	<a href="#">Προβολή</a>

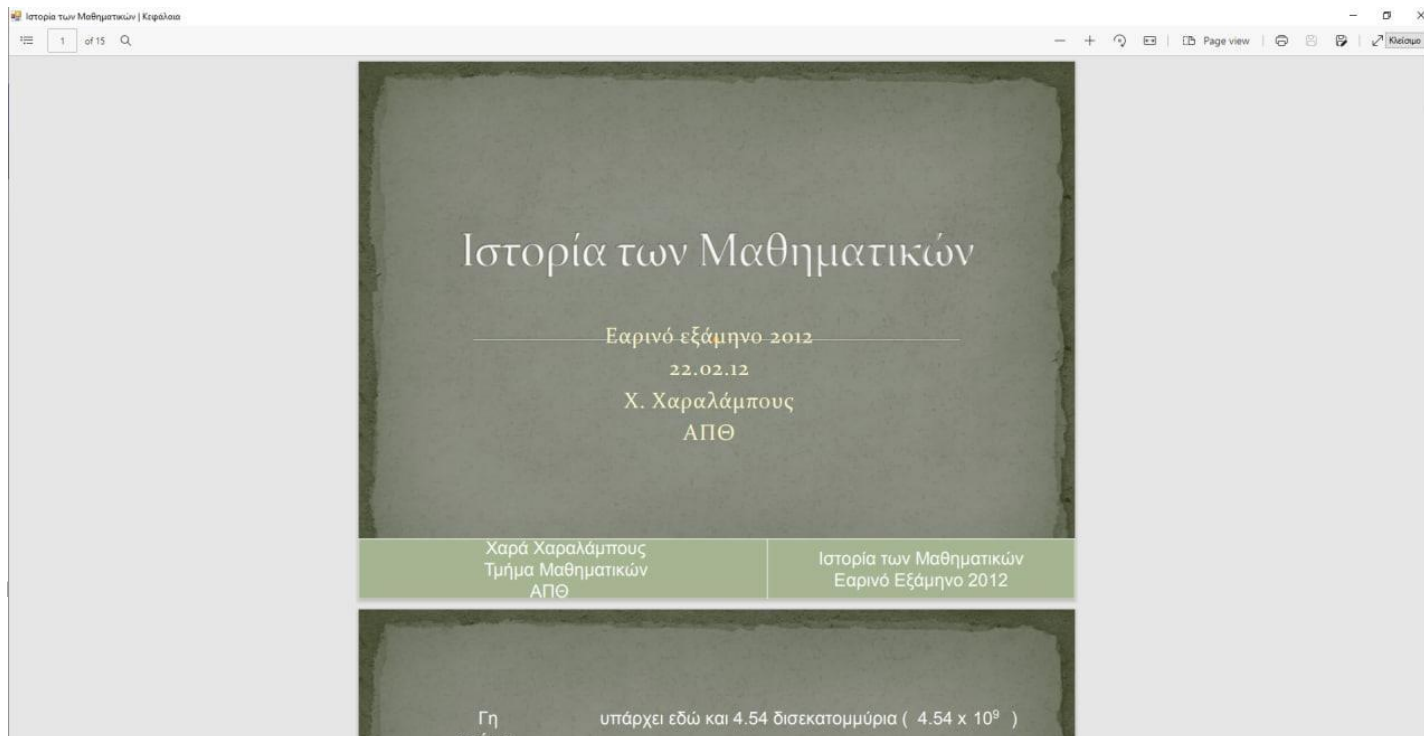
Με την ίδια διαδικασία για τα αρχεία τύπου «Διαδικτυακός

πόρος/βίντεο»  
μεταφερόμαστε στην  
αντίστοιχη ιστοσελίδα.

## Μεταπτυχιακή Διατριβή

Κεφάλαιο	Ενότητα	Τίτλος	Τύπος	Υλικό
1	1	Εισαγωγή στην ιστορία των μαθηματικών	Έγγραφο	<a href="#">Προβολή</a>
2	1	Τα Μαθηματικά των αρχαίων Αιγυπτίων, συστήματα αρίθμησης	Έγγραφο	<a href="#">Προβολή</a>
3	1	Τα Μαθηματικών των Βαβυλωνίων	Έγγραφο	<a href="#">Προβολή</a>
4	1	Τι είναι απόδειξη?	Έγγραφο	<a href="#">Προβολή</a>
5	2	Θαλής, Πυθαγόρας, Πλατωνικά Στερεά, άρρητα μεγέθη, παράδοξα του Ζήνωνα, τα περίφημα γε...	Έγγραφο	<a href="#">Προβολή</a>
6	2	Εύδοξος	Έγγραφο	<a href="#">Προβολή</a>
7	2	Τα Στοιχεία του Ευκλείδη, το πέμπτο αίτημα και οι μη Ευκλείδειες γεωμετρίες	Έγγραφο	<a href="#">Προβολή</a>
8	2	Αρχιμήδης: Ψαμμίτης, το βοεικό πρόβλημα και το εμβαδόν ανάμεσα στη παραβολή και τη τέμνο...	Έγγραφο	<a href="#">Προβολή</a>
9	2	Αρχιμήδης: Η μέθοδος: κύβος και κώνος, Διόφαντος	Έγγραφο	<a href="#">Προβολή</a>
10	3	Τι είναι Άλγεβρα? Δευτεροβάθμια εξίσωση από το 2000 π.Χ. -250 μ.Χ.	Έγγραφο	<a href="#">Προβολή</a>
11	3	Αραβικά μαθηματικά. Liber abaci. Ιταλοί και τριτοβάθμια εξίσωση	Έγγραφο	<a href="#">Προβολή</a>
12	3	Cardano και λύση της τριτοβάθμιας	Έγγραφο	<a href="#">Προβολή</a>
13	3	Τεταροβάθμια, Bombelli, Viète: οι μιγαδικοί αριθμοί είναι χρήσιμοι	Έγγραφο	<a href="#">Προβολή</a>
14	3	Ο καιρός των λογαρίθμων	Έγγραφο	<a href="#">Προβολή</a>

Μελέτη απο αρχείο τύπου «έγγραφο» (pdf)



Για να επιστρέψουμε στη κύρια σελίδα επιλέγουμε το πλήκτρο «Κλείσιμο» πάνω δεξιά.



Κεφάλαιο	Ενότητα	Τίτλος	Τύπος	Υλικό
15	4	Σκέφτομαι άρα υπάρχω...Μέθοδοι παραγώγισης και τετραγωνισμού: Descartes, Fermat	Έγγραφο	<a href="#">Προβολή</a>
16	4	Στο δρόμο για το Θεμελιώδες Θεώρημα του Λογισμού: Newton και το Θεώρημα του Διωνύμου	Έγγραφο	<a href="#">Προβολή</a>
17	4	Newton και το Θεμελιώδες Θεώρημα του Λογισμού, Leibniz	Έγγραφο	<a href="#">Προβολή</a>
18	4	Σύγκριση των Λογισμών του Newton και Leibniz, Κάποια βασικά ερωτήματα στην (Κλασική) Άλγ...	Έγγραφο	<a href="#">Προβολή</a>
19	5	Το Θεμελιώδες Θεώρημα της Άλγεβρας και απαντήσεις σε κάποια βασικά ερωτήματα της Κλασ...	Έγγραφο	<a href="#">Προβολή</a>
20	5	Reascock και συμβολική Άλγεβρα, Euler, Gauss. Στο δρόμο για την Αφηρημένη/Μοντέρνα Άλγεβρ...	Έγγραφο	<a href="#">Προβολή</a>
21	5	Μία σύντομη Εισαγωγή στην Ιστορία της Πιθανότητας (διάλεξη: Καθηγ. Π. Μωυσιάδης)	Έγγραφο	<a href="#">Προβολή</a>
22	5	Αφηρημένη Άλγεβρα: Θεωρία Ομάδων στο έργο των Gauss, Galois, Μη αντιμεταθετικοί δακτύλι...	Έγγραφο	<a href="#">Προβολή</a>
1000	6	Ερατοσθένης ο Κυρηναίος	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1007	6	Πυθαγόρας	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
▶ 1008	6	Το πείραμα του Ερατοσθένη	Διαδικτυακό βίντεο	<a href="#">Προβολή</a>
1009	7	Ευκλείδης	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1010	7	Αρχιμήδης	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1011	7	Παράδοξα του Ζήνωνα	Διαδικτυακός πόρος	<a href="#">Προβολή</a>

Μελέτη απο αρχείο τύπου «Διαδικτυακό Βίντεο»



Κεφάλαιο	Ενότητα	Τίτλος	Τύπος	Υλικό
15	4	Σκέφτομαι άρα υπάρχω...Μέθοδοι παραγωγής και τετραγωνισμού: Descartes, Fermat	Έγγραφο	<a href="#">Προβολή</a>
16	4	Στο δρόμο για το Θεμελιώδες Θεώρημα του Λογισμού: Newton και το Θεώρημα του Διωνύμου	Έγγραφο	<a href="#">Προβολή</a>
17	4	Newton και το Θεμελιώδες Θεώρημα του Λογισμού, Leibniz	Έγγραφο	<a href="#">Προβολή</a>
18	4	Σύγκριση των Λογισμών του Newton και Leibniz, Κάποια βασικά ερωτήματα στην (Κλασική) Άλγ...	Έγγραφο	<a href="#">Προβολή</a>
19	5	Το Θεμελιώδες Θεώρημα της Άλγεβρας και απαντήσεις σε κάποια βασικά ερωτήματα της Κλασ...	Έγγραφο	<a href="#">Προβολή</a>
20	5	Reasock και συμβολική Άλγεβρα, Euler, Gauss. Στο δρόμο για την Αφηρημένη/Μοντέρνα Άλγεβρ...	Έγγραφο	<a href="#">Προβολή</a>
21	5	Μία σύντομη Εισαγωγή στην Ιστορία της Πιθανότητας (διάλεξη: Καθηγ. Π. Μωυσιάδης)	Έγγραφο	<a href="#">Προβολή</a>
22	5	Αφηρημένη Άλγεβρα: Θεωρία Ομάδων στο έργο των Gauss, Galois, Μη αντιμεταθετικοί δακτύλι...	Έγγραφο	<a href="#">Προβολή</a>
▶ 1000	6	Ερατοσθένης ο Κυρηναίος	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1007	6	Πυθαγόρας	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1008	6	Το πείραμα του Ερατοσθένη	Διαδικτυακό βίντεο	<a href="#">Προβολή</a>
1009	7	Ευκλείδης	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1010	7	Αρχιμήδης	Διαδικτυακός πόρος	<a href="#">Προβολή</a>
1011	7	Παράδοξα του Ζήνωνα	Διαδικτυακός πόρος	<a href="#">Προβολή</a>

Μελέτη απο αρχείο τύπου «Διαδικτυακός Πόρος»

Ιστορία των Μαθηματικών | Κεφάλαια
Χωρίς Σύνδεση Σχόληση για αυτή την IP Συνεισφορές Δημιουργία λογαριασμού Σ Κλείσιμο

**ΒΙΚΙΠΑΙΔΕΙΑ**  
Η ελεύθερη εγκυκλοπαίδεια

Κύρια πύλη  
Θεματικές πύλες  
Προβλεπόμενα λήμματα  
Τρέχοντα γεγονότα  
Τυχαίο λήμμα

Συμμετοχή  
Βοήθεια  
Πύλη Κοινότητας  
Αγορά  
Πρόσφατες αλλαγές  
Επικοινωνία  
Δωρεές

Εργαλεία  
Συνδέσεις προς εδώ  
Σχετικές αλλαγές  
Ειδικές σελίδες  
Σταθερός σύνδεσμος  
Πληροφορίες σελίδας  
Παραπομπή  
Αντικείμενο Wikidata

**Ionian Wikithon 2022**  
14-23 Μαΐου 2022 στο Ιόνιο Πανεπιστήμιο (Κέρκυρα)

Από τις 21 Μαρτίου έως τις 31 Μαΐου, λάβετε μέρος στον διαγωνισμό σύνταξης λημμάτων για τις χώρες της Κεντρικής και Ανατολικής Ευρώπης.

Συνεισφέρετε γράφοντας στο CEE Spring 2022!

## Ερατοσθένης ο Κυρηναίος

Από τη Βικιπαίδεια, την ελεύθερη εγκυκλοπαίδεια

*Για άλλες χρήσεις, δείτε: Ερατοσθένης.*

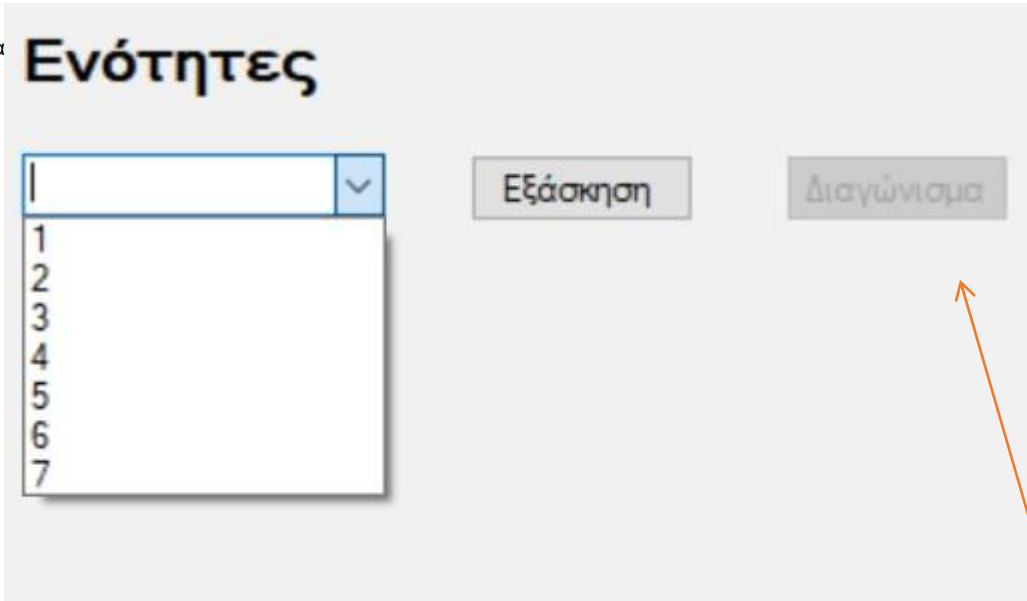
Ο **Ερατοσθένης ο Κυρηναίος** (Κυρήνη, 276 π.Χ. – Αλεξάνδρεια, 194 π.Χ.) ήταν αρχαίος Έλληνας μαθηματικός, γεωγράφος, αστρονόμος, γεωδαίτης, μουσικός, ποιητής, ιστορικός, φιλόλογος και συγγραφέας, ο οποίος θεωρείται ο πρώτος άνθρωπος στην ιστορία που υπολόγισε το μέγεθος της Γης και κατασκεύασε ένα σύστημα συντεταγμένων με παράλληλους και μεσημβρινούς. Επίσης, κατασκεύασε και έναν χάρτη του κόσμου, όπως τον θεωρούσε.

**Πίνακας περιεχομένων** [Απόκρυψη]

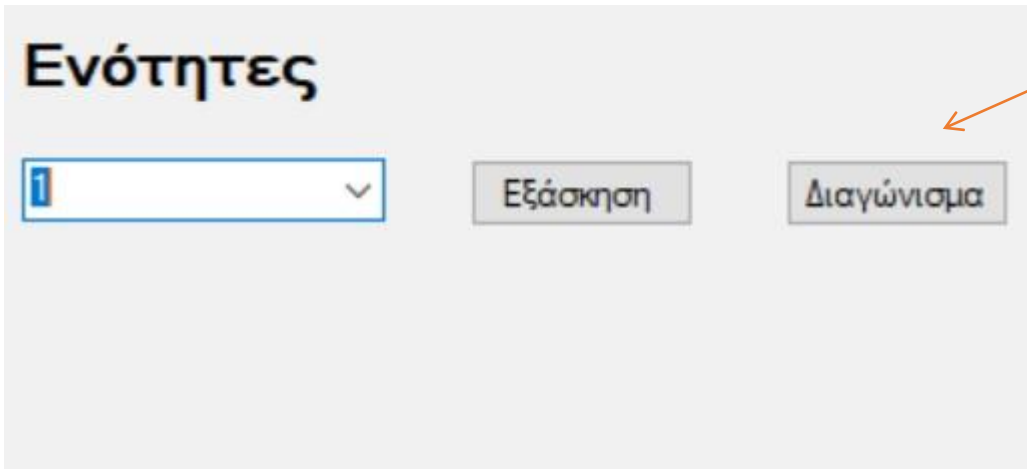
- 1 Βιογραφία
- 2 Το έργο του
- 3 Η μέτρηση της περιφέρειας της Γης από τον Ερατοσθένη
- 4 Παραπομπές
- 5 Δείτε επίσης

**Ερατοσθένης ο Κυρηναίος**





Πατώντας στη μπάρα μπορούμε να επιλέξουμε μια ενότητα και με το πλήκτρο «Εξάσκηση» να εξασκηθούμε σε ένα δοκιμαστικό τεστ ερωτήσεων ενώ με το πλήκτρο «Διαγώνισμα» να εξεταστούμε στο διαγώνισμα της ενότητας.



Σημείωση : Για να ξεκλειδώσουμε το πλήκτρο «Διαγώνισμα» για κάποια ενότητα πρέπει να έχουμε μελετήσει μέσω της εφαρμογής όλα τα αρχεία της αντίστοιχης

ενότητας τουλάχιστον μια φορά και να έχουμε ολοκληρώσει τουλάχιστον ένα (1) τεστ εξάσκησης.

Επιλέγοντας το πλήκτρο «Εξάσκηση» μεταφερόμαστε στο παρακάτω παράθυρο. Στη κορυφή εμφανίζονται οι ερωτήσεις οι οποίες δύναται να είναι σε μορφή πολλαπλής επιλογής, συμπλήρωση κενού, σωστό-λάθος ή αντιστοίχιση. Το πλήκτρο «Τυχαία απάντηση» συμπληρώνει τυχαία την ερώτηση, το πλήκτρο «Προβολή σωστού» εμφανίζει τη σωστή απάντηση και το κεφάλαιο από το οποίο αντλήθηκε η ερώτηση, ενώ με το «Επόμενο» περνάμε στην επόμενη ερώτηση μέχρις ότου να ολοκληρώσουμε το τεστ εξάσκησης. Αν θέλουμε να σταματήσουμε το τεστ πατάμε το πλήκτρο «Πίσω».

Ιστορία των Μαθηματικών | Εξάσκηση

Πίσω

**Ερώτηση 2 από 15**

Για την αρίθμηση στη Βαβυλωνία χρησιμοποιούσαν ως βάση το:  
a)100 b)10 c)60 d)50

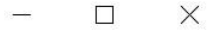
Τυχαία απάντηση

Επόμενο

Προβολή σωστού

b  
Κεφάλαιο 3

Εάν πατήσουμε το πλήκτρο «Διαγώνισμα» θα μεταφερθούμε σε ένα παράθυρο, ίδιο με το τεστ εξάσκησης, με τη διαφορά ότι δεν υπάρχει το πλήκτρο «Προβολή σωστού». Με την ολοκλήρωση του διαγωνίσματος εμφανίζονται τα αποτελέσματα, ακριβώς όπως και στο τέλος του τεστ εξάσκησης.

[Πίσω](#)**Ερώτηση 1 από 15**

Ο πάπυρος του Rhind που περιέχει προβλήματα μαθηματικής φύσης βρέθηκε στην  
\_ (1650 π.Χ.).

[Τυχαία απάντηση](#)[Επόμενο](#)

**Ερώτηση 10 από 25**

Αντιστοιχίστε τους Μαθηματικούς με τις χρονολογίες.  
1)Parelman 2)John Wiles 3)Fermat 4)Riemann  
a)1826-1866 b)1966- c)1953- d)1601-1665

1	a
2	b
3	
4	a b c d

Παράδειγμα ερώτησης τύπου αντιστοίχισης.

Επιλέγουμε στα αριστερά τον αριθμό (1, 2, 3, 4) και αντιστοιχίζουμε στα δεξιά το κατάλληλο γράμμα (a, b, c, d)

**Ερώτηση 11 από 25**

Κάποια από τα μαθηματικά αποτελέσματα στα βιβλία 5,12 των "Στοιχείων" του Ευκλείδη αποδίδονται στον Πυθαγόρα.

Παράδειγμα τύπου «Σωστό-Λάθος»

Επιλέγουμε το πλήκτρο «Σωστό» ή «Λάθος»



Ιστορία των Μαθηματικών | Αποτελέσματα

[Πίσω](#)

## Αποτελέσματα

Τύπος: Εξάσκηση

Ενότητα: 1

Συνολικές ερωτήσεις: 15

Απαντημένες ερωτήσεις: 14

Σωστές ερωτήσεις: 11

Βαθμός: 11/15 (Περάσατε)

Ολοκληρώνοντας το τεστ εξάσκησης ή το διαγώνισμα εμφανίζονται τα αποτελέσματα, εκ των οποίων και ο βαθμός του φοιτητή στο συγκεκριμένο τεστ και εάν πέρασε ή κόπηκε.

Ο βαθμός επιτυχίας ορίζεται μεγαλύτερος ή ίσος του 50%.

Στο τέλος επιστρέφουμε στη κύρια σελίδα της εφαρμογής

επιλέγοντας το πλήκτρο  
«Πίσω».

## Στατιστικά

Αναγνώσεις κεφαλαίων: 12  
Μέσος χρόνος ανάγνωσης: 5.09 δεπτ.

Εξασκήσεις: 8

Διαγωνίσματα: 3

Τελευταίος βαθμός: 3/15 (Κοπήκατε)

Πίσω στη κύρια σελίδα της εφαρμογής στα «Στατιστικά» εμφανίζονται οι αναγνώσεις κεφαλαίων και ο μέσος χρόνος

ανάγνωσης του κάθε κεφαλαίου.

Στην εικόνα παραπάνω φαίνονται οι συνολικές εξασκήσεις και τα συνολικά διαγωνίσματα στα οποία εξετάστηκε ο φοιτητής.

Τέλος απο κάτω μπορεί να δει ο φοιτητής τον βαθμό του στο τελευταίο διαγώνισμα στο οποίο εξετάστηκε.