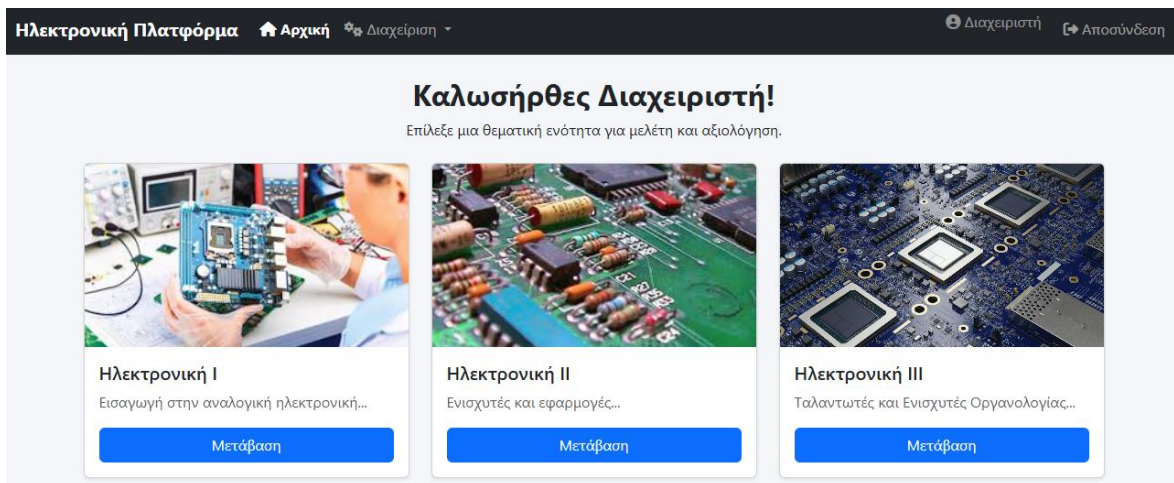


## ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

### ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«Διαδικτυακή πλατφόρμα μάθησης για την ηλεκτρονική»



Ηλεκτρονική Πλατφόρμα [Αρχική](#) [Διαχείριση](#) [Διαχειριστή](#) [Αποσύνδεση](#)

### Καλωσήρθες Διαχειριστή!

Επίλεξε μια θεματική ενότητα για μελέτη και αξιολόγηση.

**Ηλεκτρονική I**  
Εισαγωγή στην αναλογική ηλεκτρονική...  
[Μετάβαση](#)

**Ηλεκτρονική II**  
Ενισχυτές και εφαρμογές...  
[Μετάβαση](#)

**Ηλεκτρονική III**  
Ταλαντωτές και Ενισχυτές Οργανολογίας...  
[Μετάβαση](#)

**Του Φοιτητή**

ΠΑΣΧΑΛΗΣ ΖΗΣΟΠΟΥΛΟΣ - 514041

**Επιβλέπων**

Κυριάκος Τσιακμάκης  
Επίκουρος Καθηγητής

Σεπτέμβριος 2025

Διαδικτυακή πλατφόρμα μάθησης για την ηλεκτρονική

Κωδικός: 25157

Φοιτητής: Ζησόπουλος Πασχάλης

Εισηγητής: Δρ Κυριάκος Τσιακμάκης

Ημερομηνία ανάληψης Π.Ε. 08-03-2025

Ημερομηνία περάτωσης Π.Ε. 30-08-2025

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ζησόπουλου Πασχάλη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.



## Περίληψη

Η παρούσα εργασία αφορά τον σχεδιασμό και την υλοποίηση μιας διαδικτυακής πλατφόρμας μάθησης με αντικείμενο την Ηλεκτρονική, με στόχο την ενίσχυση της εκπαιδευτικής εμπειρίας των φοιτητών. Η πλατφόρμα παρέχει οργανωμένο εκπαιδευτικό υλικό, σημειώσεις, βίντεο διαλέξεων, ασκήσεις κυκλωμάτων. Υλοποιείται με χρήση της γλώσσας Python και του framework Flask, ενώ αξιοποιούνται επίσης HTML, CSS, Bootstrap και MySQL για την αποθήκευση των δεδομένων. Το σύστημα επιτρέπει την πρόσβαση σε μαθήματα μέσω φιλικού περιβάλλοντος, με δυνατότητα παρακολούθησης της προόδου του χρήστη. Μέσω αυτής της εφαρμογής, επιδιώκεται η σύνδεση θεωρίας και πράξης με τρόπο διαδραστικό και προσβάσιμο αξιοποιώντας τις δυνατότητες της τεχνολογίας στην εκπαίδευση.

## « Online learning platform for electronics»

### **Abstract**

This work concerns the design and implementation of an online learning platform for Electronics, with the aim of enhancing the educational experience of students. The platform provides organized educational material, notes, lecture videos, circuit exercises. It is implemented using the Python language and the Flask framework, while HTML, CSS, Bootstrap and MySQL are also used for data storage. The system allows access to courses through a friendly environment, with the ability to monitor the user's progress. Through this application, the connection between theory and practice is sought in an interactive and accessible way, utilizing the capabilities of technology in education.

## **Ευχαριστίες**

Να ευχαριστήσω όσους βοήθησαν και σαν ώθηση να τελειώσω την πτυχιακή αλλά και στον κώδικα και στις συμβουλές τους, τους φίλους μου και τον επιβλέποντα μου.

# Περιεχόμενα

Περίληψη .....	iv
Abstract .....	v
Ευχαριστίες .....	vi
Περιεχόμενα.....	vii
Κατάλογος Σχημάτων .....	viii
Κεφάλαιο 1ο: Εισαγωγή.....	9
1.1 Εισαγωγή.....	9
Κεφάλαιο 2ο: Παρόμοια συστήματα διαχείρισης εξετάσεων και δημιουργίας διαδικτυακών δοκιμασιών προόδου .....	11
2.1 Moodle: Ανοιχτού Κώδικα Πλατφόρμα Διαχείρισης Μάθησης.....	11
2.2 Chamilo .....	13
2.3 Open edX.....	15
2.4 Google Classroom .....	17
Κεφάλαιο 3ο: Λογισμικό και εργαλεία που χρησιμοποιήθηκαν .....	21
3.1 Flask.....	21
3.1.1 Python.....	21
3.2 Βάση δεδομένων MySQL.....	25
3.3 Bootstrap 5.....	29
Κεφάλαιο 4ο: Διαδικτυακή πλατφόρμα μάθησης για την ηλεκτρονική.....	31
4.1 Περιγραφή .....	31
4.2 Περιγραφή μέσω διαγραμμάτων.....	46
4.3 Η βάση που χρησιμοποιήθηκε .....	49
4.4 Προσέγγιση για την Ασφάλεια .....	51
Κεφάλαιο 5ο: Τα συμπεράσματα της εργασίας και κάποιες προτάσεις για βελτίωση .....	53
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	55
ΚΩΔΙΚΑΣ.....	56

## Κατάλογος Σχημάτων

Εικόνα 2.1: Moodle – ερωτήσεις quiz .....	11
Εικόνα 2.2: Chamilo.....	14
Εικόνα 2.3: Open edX.....	16
Εικόνα 2.4: Google Classroom.....	18
Εικόνα 3.1: Bootstrap Quiz.....	29
Εικόνα 4.1: Σελίδα σύνδεσης και ενεργοποίησης λογαριασμού.....	32
Εικόνα 4.2: Θεματικές ενότητες για μελέτη και αξιολόγηση .....	33
Εικόνα 4.3: Μενού του διαχειριστή .....	33
Εικόνα 4.4: Ο admin μπορεί να δει το σύνολο των σελίδων για τη κάθε θεματική.....	34
Εικόνα 4.5: Λίστα Θεματικών Ενότητων .....	34
Εικόνα 4.6: Σελίδα για Νέα Θεματική Ενότητα.....	35
Εικόνα 4.7: Σελίδα για Επεξεργασία Θεματικής Ενότητας .....	35
Εικόνα 4.8: Λίστα Όλων των Σελίδων Θεματικών Ενότητων .....	36
Εικόνα 4.9: Δημιουργία Νέας Σελίδας .....	36
Εικόνα 4.10: Επεξεργασία Σελίδας 1.....	37
Εικόνα 4.11: Επεξεργασία Σελίδας 2 – και με βίντεο και υπερσυνδέσμους .....	38
Εικόνα 4.12: Διαχείριση Quiz .....	38
Εικόνα 4.13: Δημιουργία Νέου Quiz .....	39
Εικόνα 4.14: Επεξεργασία Quiz .....	39
Εικόνα 4.15: Ερωτήσεις για Quiz στο θέμα: Ηλεκτρονική Ι .....	40
Εικόνα 4.16: Επεξεργασία Ερώτησης για Quiz - 1.....	40
Εικόνα 4.17: Επεξεργασία Ερώτησης για Quiz - 2.....	41
Εικόνα 4.18: Νέα Ερώτηση για Quiz: Ηλεκτρονική ΙΙ με επιλογή σελίδας.....	41
Εικόνα 4.19: Όλοι οι Χρήστες .....	42
Εικόνα 4.20: Σελίδα για εγγραφή χρήστη .....	42
Εικόνα 4.21: Αρχική σελίδα του χρήστη - βλέπει θεματικές μόνο του level που έχει πετύχει – το μενού έχει ιστορικό για τα quiz του.....	43
Εικόνα 4.22: Ιστοσελίδα που βλέπει το υλικό θεωρίας για να διαβάσει και να ετοιμαστεί για το Quiz θεματικής – όταν είναι έτοιμος πατάει το Ξεκίνα Quiz Αξιολόγησης .....	43
Εικόνα 4.23: Έναρξη για το «Αξιολόγηση Θεματικής Ενότητας: Ηλεκτρονική Ι» .....	44
Εικόνα 4.24: Ερωτήσεις που πρέπει να απαντήσει ο χρήστης με όποια σειρά θέλει και στο τέλος να πατήσει Υποβολή Quiz .....	44
Εικόνα 4.25: Αποτελέσματα Quiz.....	45
Εικόνα 4.26: Ιστορικό αποτελεσμάτων Quiz που έχει συμμετάσχει ο χρήστης .....	46
Εικόνα 4.27: Δομική διάταξη συστήματος.....	47
Εικόνα 4.28: Λειτουργίες διαχειριστή.....	48
Εικόνα 4.29: Δομική διάταξη συστήματος.....	48

# Κεφάλαιο 1ο: Εισαγωγή

## 1.1 Εισαγωγή

Τα τελευταία χρόνια, η τεχνολογία έχει αλλάξει τον τρόπο με τον οποίο πραγματοποιείται η εκπαιδευτική διαδικασία. Οι διαδικτυακές πλατφόρμες μάθησης έχουν γίνει βασικό εργαλείο σε πολλά επίπεδα εκπαίδευσης, από τα σχολεία και τα πανεπιστήμια μέχρι τη δια βίου μάθηση. Οι πλατφόρμες αυτές είναι ψηφιακά συστήματα που επιτρέπουν σε εκπαιδευτικούς και φοιτητές να επικοινωνούν, να ανταλλάσσουν υλικό, να οργανώνουν μαθήματα και να συμμετέχουν σε δραστηριότητες εξ αποστάσεως, χωρίς να απαιτείται η φυσική τους παρουσία στον ίδιο χώρο.

Μέσα από μία διαδικτυακή πλατφόρμα μάθησης, ο φοιτητής έχει τη δυνατότητα να παρακολουθεί θεωρία, να διαβάζει σημειώσεις, να απαντά σε ερωτήσεις, να συμμετέχει σε quiz και ασκήσεις αξιολόγησης, να βλέπει την πρόοδό του και να επανέρχεται σε οποιοδήποτε σημείο του μαθήματος οποιαδήποτε στιγμή. Από την πλευρά του, ο διδάσκων μπορεί να οργανώσει το μάθημά του με δομημένο τρόπο, να ανεβάζει υλικό, να δημιουργεί αξιολογήσεις και να βλέπει στατιστικά στοιχεία για τους φοιτητές του. Η επικοινωνία μεταξύ των δύο πλευρών γίνεται απρόσκοπτα, με τη βοήθεια τεχνολογικών εργαλείων που κάνουν το περιβάλλον απλό, λειτουργικό και ευέλικτο.

Η ανάγκη για τέτοιες πλατφόρμες αναδείχθηκε ακόμη περισσότερο κατά τη διάρκεια της πανδημίας, όταν η εξ αποστάσεως εκπαίδευση έγινε αναγκαστική για μεγάλο χρονικό διάστημα. Εκείνη την περίοδο, πολλές σχολικές και πανεπιστημιακές μονάδες στράφηκαν σε εργαλεία όπως το Moodle, το Google Classroom ή άλλες προσαρμοσμένες λύσεις. Η εμπειρία αυτή κατέδειξε ότι η τεχνολογία μπορεί να καλύψει με επιτυχία σημαντικό μέρος της εκπαιδευτικής εμπειρίας, εφόσον υπάρχει σωστή οργάνωση, εύχρηστο περιβάλλον και ποιοτικό περιεχόμενο.

Πέρα από την εξ αποστάσεως εκπαίδευση, όμως, οι πλατφόρμες αυτές προσφέρουν σημαντικά πλεονεκτήματα και σε συνδυασμό με τη διά ζώσης διδασκαλία. Χρησιμοποιούνται ως υποστηρικτικά εργαλεία στην εκπαιδευτική διαδικασία, δίνοντας στους φοιτητές τη δυνατότητα να επαναλάβουν την ύλη, να μελετήσουν με το δικό τους ρυθμό και να αξιολογήσουν τις γνώσεις τους μέσα από διαδραστικές δραστηριότητες. Με τον τρόπο αυτό, ενισχύεται η αυτονομία του μαθητή, η εξατομίκευση της μάθησης και η ενεργητική συμμετοχή του στην εκπαιδευτική διαδικασία.

Η τεχνολογία που χρησιμοποιείται πίσω από τις πλατφόρμες αυτές διαφέρει από περίπτωση σε περίπτωση, όμως βασίζεται σε αρχές που αφορούν τη λειτουργικότητα, την ασφάλεια, την ευκολία χρήσης και την επεκτασιμότητα. Από τη βάση δεδομένων που αποθηκεύει τους χρήστες και τα μαθήματα, μέχρι το γραφικό περιβάλλον που επιτρέπει την πλοήγηση και την αλληλεπίδραση κάθε κομμάτι συμβάλλει ώστε η εμπειρία του χρήστη να είναι θετική και αποτελεσματική.

Στην εργασία αυτή σχεδιάστηκε και υλοποιήθηκε μια τέτοια πλατφόρμα με επίκεντρο την εκπαιδευτική ενότητα της Ηλεκτρονικής. Η εφαρμογή δημιουργήθηκε με στόχο να προσφέρει στους φοιτητές ένα εργαλείο μάθησης οργανωμένο, ευχάριστο στη χρήση και παιδαγωγικά ουσιαστικό.

Η παρούσα εργασία εστιάζει στην υλοποίηση μιας διαδικτυακής πλατφόρμας μάθησης που έχει ως στόχο τη διδασκαλία και την αξιολόγηση φοιτητών στο αντικείμενο της Ηλεκτρονικής. Η επιλογή του θέματος δεν είναι τυχαία, καθώς η Ηλεκτρονική αποτελεί ένα σημαντικό και συχνά απαιτητικό γνωστικό πεδίο για φοιτητές τεχνικών και τεχνολογικών κατευθύνσεων. Το περιεχόμενο της συνδυάζει θεωρία, μαθηματική τεκμηρίωση και εφαρμογές που χρειάζονται εμπειριστατωμένη κατανόηση. Η ανάγκη για συνεχή εξάσκηση, η οπτικοποίηση εννοιών και η σταδιακή αξιολόγηση των γνώσεων καθιστούν το συγκεκριμένο αντικείμενο ιδανικό για ψηφιακή υποστήριξη μέσω εκπαιδευτικής πλατφόρμας.

Η εφαρμογή που αναπτύχθηκε δίνει τη δυνατότητα στους φοιτητές να έχουν πρόσβαση σε δομημένο εκπαιδευτικό υλικό, το οποίο περιλαμβάνει θεωρητικά κείμενα, παραδείγματα και εικόνες που βοηθούν στην κατανόηση των βασικών εννοιών της Ηλεκτρονικής. Επιπλέον περιλαμβάνει μηχανισμό αξιολόγησης με ερωτήσεις πολλαπλής επιλογής (quiz), τα οποία έχουν σχεδιαστεί έτσι ώστε να συνδέονται άμεσα με τις αντίστοιχες ενότητες της θεωρίας. Ο φοιτητής μπορεί να μελετήσει με τον δικό του ρυθμό, να επιστρέψει όσες φορές χρειάζεται στο υλικό και να αξιολογήσει τον εαυτό του, βλέποντας τα αποτελέσματά του αμέσως μετά την ολοκλήρωση κάθε quiz.

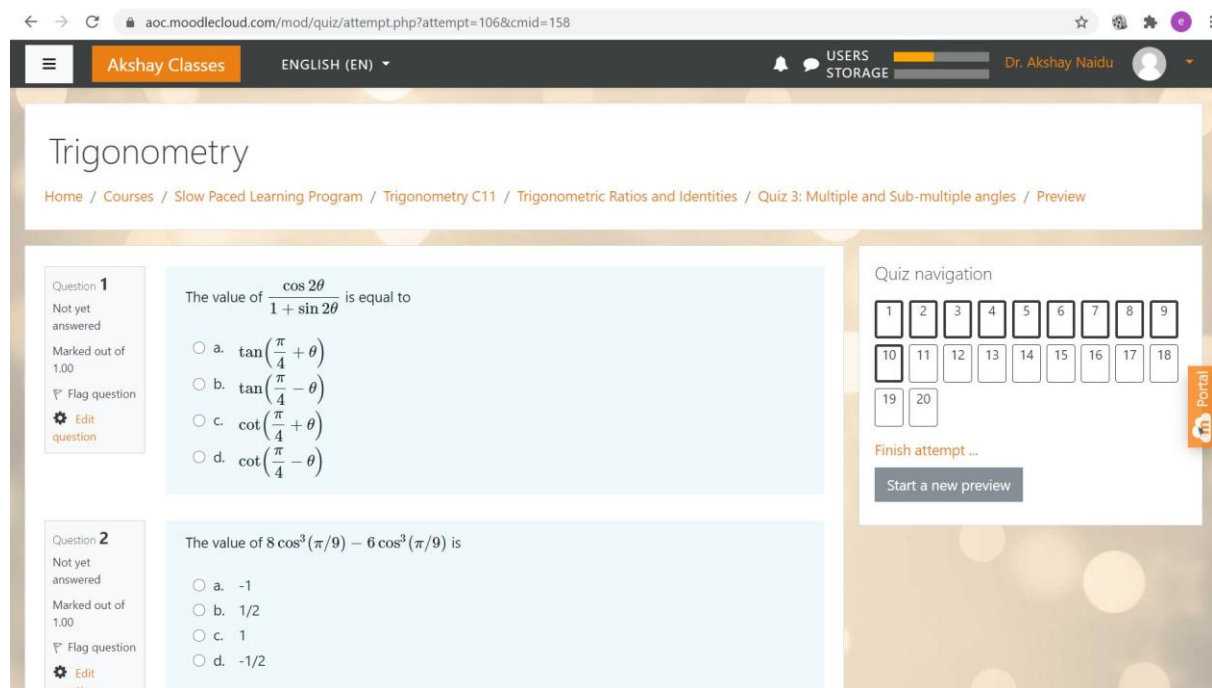
Η πλατφόρμα υποστηρίζει επίσης την έννοια της προοδευτικής μάθησης καθώς οι θεματικές ενότητες είναι οργανωμένες με τέτοιο τρόπο ώστε να ανοίγουν σταδιακά ανάλογα με την επίδοση του φοιτητή. Αυτό ενισχύει την προσπάθεια, προσφέρει ένα στοιχείο πρόκλησης και ανταμοιβής και βοηθά στην καλύτερη παρακολούθηση της προόδου. Ο διαχειριστής του συστήματος έχει τη δυνατότητα να παρακολουθεί την απόδοση των χρηστών, να διαχειρίζεται τις θεματικές ενότητες και τα quiz, να προσθέτει νέο υλικό και να επεμβαίνει όποτε χρειάζεται.

Η τεχνολογική υλοποίηση βασίστηκε σε Python και Flask για το backend, σε συνδυασμό με MySQL για την αποθήκευση των δεδομένων και HTML/CSS/Bootstrap για το frontend. Το αποτέλεσμα είναι μια εφαρμογή πλήρως λειτουργική, φιλική στον χρήστη και κατάλληλη για εκπαιδευτικά περιβάλλοντα, είτε αυτόνομα είτε σε συνδυασμό με τη φυσική τάξη. Η εργασία αυτή δείχνει πώς μπορεί να αξιοποιηθεί η τεχνολογία στην πράξη, ώστε να ενισχυθεί η διδασκαλία ενός τεχνικού μαθήματος και να δημιουργηθούν προϋποθέσεις για πιο αποτελεσματική και ενδιαφέρουσα μάθηση.

## Κεφάλαιο 2ο: Παρόμοια συστήματα διαχείρισης εξετάσεων και δημιουργίας διαδικτυακών δοκιμασιών προόδου

### 2.1 Moodle: Ανοιχτού Κώδικα Πλατφόρμα Διαχείρισης Μάθησης

Το Moodle (Modular Object-Oriented Dynamic Learning Environment) αποτελεί μία από τις πιο διαδεδομένες πλατφόρμες διαχείρισης μάθησης (LMS) παγκοσμίως. Αναπτύχθηκε το 2002 από τον Αυστραλό προγραμματιστή Martin Dougiamas με στόχο την υποστήριξη της εξ αποστάσεως εκπαίδευσης και της μικτής μάθησης, παρέχοντας ένα ευέλικτο και προσαρμόσιμο περιβάλλον για εκπαιδευτικούς και μαθητές. [1-2]



Εικόνα 2.1: Moodle – ερωτήσεις quiz

[[https://fiverr-res.cloudinary.com/images/q\\_auto,f\\_auto/gigs/198097350/original/57be5388aed7035e6619f4c3576e985296db9cd/create-quizzes-on-moodle-or-any-other-lms.jpg](https://fiverr-res.cloudinary.com/images/q_auto,f_auto/gigs/198097350/original/57be5388aed7035e6619f4c3576e985296db9cd/create-quizzes-on-moodle-or-any-other-lms.jpg)]

Το Moodle είναι μια πλατφόρμα ανοιχτού κώδικα, γραμμένη σε PHP, και διανέμεται υπό την άδεια GNU General Public License. Αυτό σημαίνει ότι είναι ελεύθερα διαθέσιμη για λήψη, εγκατάσταση και προσαρμογή, χωρίς κόστος αδειοδότησης. Η ευελιξία αυτή επιτρέπει σε εκπαιδευτικά ιδρύματα και οργανισμούς να διαμορφώσουν την πλατφόρμα σύμφωνα με τις δικές τους ανάγκες και προδιαγραφές.

Η πλατφόρμα υποστηρίζει τη δημιουργία και διαχείριση μαθημάτων, την εγγραφή χρηστών, την παρακολούθηση της προόδου των μαθητών, καθώς και την ενσωμάτωση διαφόρων τύπων εκπαιδευτικού υλικού, όπως κείμενα, βίντεο, κουίζ και φόρουμ συζητήσεων. Επιπλέον, προσφέρει εργαλεία αξιολόγησης, όπως βαθμολόγηση, ανατροφοδότηση και αναλυτικά στατιστικά στοιχεία για την απόδοση των μαθητών.

Το Moodle χρησιμοποιείται ευρέως σε εκπαιδευτικά ιδρύματα όλων των βαθμίδων, από σχολεία και πανεπιστήμια έως οργανισμούς επαγγελματικής κατάρτισης και επιχειρήσεις. Η ευελιξία και η προσαρμοστικότητά του το καθιστούν κατάλληλο για διάφορα εκπαιδευτικά σενάρια, όπως:

- Εξ αποστάσεως εκπαίδευση: Παροχή μαθημάτων και εκπαιδευτικού υλικού σε μαθητές που δεν μπορούν να παρευρίσκονται φυσικά στην τάξη.
- Μικτή μάθηση: Συνδυασμός παραδοσιακής διδασκαλίας με διαδικτυακές δραστηριότητες και αξιολογήσεις.
- Επαγγελματική κατάρτιση: Εκπαίδευση εργαζομένων σε νέες δεξιότητες ή διαδικασίες εντός ενός οργανισμού.
- Αυτοκατευθυνόμενη μάθηση: Δυνατότητα στους μαθητές να μελετούν με τον δικό τους ρυθμό, επιλέγοντας τα μαθήματα και τις δραστηριότητες που τους ενδιαφέρουν.

Ένα από τα κύρια πλεονεκτήματα του Moodle είναι η ενεργή κοινότητα χρηστών και προγραμματιστών που συμβάλλει συνεχώς στη βελτίωση και την επέκταση της πλατφόρμας μέσω προσθηκών (plugins) και θεμάτων (themes). Αυτό επιτρέπει την προσαρμογή της πλατφόρμας σε συγκεκριμένες ανάγκες, όπως η ενσωμάτωση με άλλα συστήματα, η υποστήριξη πολλαπλών γλωσσών και η προσαρμογή της εμφάνισης.

Επιπλέον, το Moodle υποστηρίζει πρότυπα διαλειτουργικότητας όπως το SCORM και το LTI, επιτρέποντας την ενσωμάτωση εξωτερικών εργαλείων και περιεχομένου. Η πλατφόρμα διαθέτει επίσης εφαρμογή για κινητές συσκευές, επιτρέποντας στους χρήστες να έχουν πρόσβαση στο εκπαιδευτικό υλικό από οπουδήποτε και οποτεδήποτε.

Το Moodle αποτελεί μια ισχυρή και ευέλικτη λύση για τη διαχείριση της διαδικτυακής μάθησης, προσφέροντας ένα πλήρες σύνολο εργαλείων για τη δημιουργία, διανομή και αξιολόγηση εκπαιδευτικού περιεχομένου. Η ανοιχτή φύση του, σε συνδυασμό με την ενεργή κοινότητα και την ευρεία υιοθέτησή του, το καθιστούν μια αξιόπιστη επιλογή για εκπαιδευτικά ιδρύματα και οργανισμούς που επιδιώκουν να προσφέρουν ποιοτική διαδικτυακή εκπαίδευση.

Παρατηρώντας τη λειτουργικότητα του Moodle και συγκρίνοντάς τη με την πλατφόρμα που υλοποιήθηκε στο πλαίσιο της παρούσας εργασίας, αναδεικνύονται αρκετά κοινά χαρακτηριστικά. Και τα δύο συστήματα έχουν ως βασικό τους στόχο τη διάθεση εκπαιδευτικού υλικού με οργανωμένο τρόπο, τη διαχείριση μαθημάτων ανά θεματική ενότητα και την αξιολόγηση των χρηστών μέσω quiz.

Όπως και στο Moodle, έτσι και στο δικό μας σύστημα υπάρχει υποστήριξη διαφορετικών ρόλων χρηστών, όπως φοιτητής και διαχειριστής, με αντίστοιχες δυνατότητες πρόσβασης και ελέγχου. Επιπλέον, το στοιχείο της προόδου του φοιτητή και της παρακολούθησης των αποτελεσμάτων αποτελεί επίσης βασική λειτουργία και των δύο. Η δυνατότητα δημιουργίας quiz με ερωτήσεις πολλαπλής επιλογής, η αυτόματη διόρθωση και η αποθήκευση των αποτελεσμάτων στο ιστορικό του φοιτητή είναι παρούσες και στα δύο συστήματα. Αν και το Moodle προσφέρει πολύ περισσότερες δυνατότητες λόγω της ωριμότητας και του μεγέθους του, η πλατφόρμα που υλοποιήθηκε μοιράζεται την ίδια φιλοσοφία: την υποστήριξη της εκπαιδευτικής διαδικασίας μέσα από ένα ευέλικτο, επεκτάσιμο και προσβάσιμο διαδικτυακό περιβάλλον. Η απλότητα, η λειτουργικότητα και η προσαρμογή στις ανάγκες του μαθήματος της Ηλεκτρονικής καθιστούν το παρόν σύστημα μια πιο ελαφριά αλλά εξειδικευμένη εκδοχή αντίστοιχων πλατφορμών ευρείας χρήσης όπως το Moodle.

## 2.2 Chamilo

Το Chamilo είναι μια δωρεάν και ανοιχτού κώδικα πλατφόρμα διαχείρισης μάθησης (LMS), σχεδιασμένη για να υποστηρίξει την αποτελεσματική διαδικτυακή εκπαίδευση. Αναπτύχθηκε μέσω της συνεργασίας διαφόρων εταιρειών, οργανισμών και ατόμων, σύμφωνα με ένα μοντέλο γνωστό ως ελεύθερο λογισμικό, αλλά με αυστηρότερες ηθικές αξίες. Αυτό σημαίνει ότι μπορείτε να κατεβάσετε και να χρησιμοποιήσετε το Chamilo δωρεάν, υπό την προϋπόθεση ότι αποδέχεστε τους όρους της άδειάς του, που περιγράφονται λεπτομερώς στην άδεια GNU/GPLv3. Αυτό σας παρέχει τέσσερις βασικές ελευθερίες: την ελευθερία χρήσης, μελέτης, τροποποίησης και διανομής του λογισμικού. [3]

Το Chamilo προσφέρει ένα ευρύ φάσμα εργαλείων για τη δημιουργία και διαχείριση μαθημάτων, την εγγραφή χρηστών, την παρακολούθηση της προόδου των μαθητών και την ενσωμάτωση διαφόρων τύπων εκπαιδευτικού υλικού, όπως κείμενα, βίντεο, κουίζ και φόρουμ συζητήσεων. Επιπλέον, παρέχει εργαλεία αξιολόγησης, όπως βαθμολόγηση, ανατροφοδότηση και αναλυτικά στατιστικά στοιχεία για την απόδοση των μαθητών.

Το Chamilo χρησιμοποιείται ευρέως σε εκπαιδευτικά ιδρύματα όλων των βαθμίδων, από σχολεία και πανεπιστήμια έως οργανισμούς επαγγελματικής κατάρτισης και επιχειρήσεις. Η ευελιξία και η προσαρμοστικότητά του το καθιστούν κατάλληλο για διάφορα εκπαιδευτικά σενάρια, όπως:

- Εξ αποστάσεως εκπαίδευση: Παροχή μαθημάτων και εκπαιδευτικού υλικού σε μαθητές που δεν μπορούν να παρευρίσκονται φυσικά στην τάξη.
- Μικτή μάθηση: Συνδυασμός παραδοσιακής διδασκαλίας με διαδικτυακές δραστηριότητες και αξιολογήσεις.

- Επαγγελματική κατάρτιση: Εκπαίδευση εργαζομένων σε νέες δεξιότητες ή διαδικασίες εντός ενός οργανισμού.
- Αυτοκατευθυνόμενη μάθηση: Δυνατότητα στους μαθητές να μελετούν με τον δικό τους ρυθμό, επιλέγοντας τα μαθήματα και τις δραστηριότητες που τους ενδιαφέρουν.

The screenshot shows the 'Test Your English / Assessments' page in Chamilo. At the top, it indicates 'Total weight : 100 - Minimum certification score : 0'. Below this is a table with the following data:

Type	Name	Description	Weight	Edit
	<b>Test your English</b>	-	<b>100</b>	[Icons: pencil, eye, percentage, eye, X]
	Class presentation <b>Score</b>	-	20	[Icons: pencil, eye, X]
	A family dinner <b>Forum threads</b>	-	20	[Icons: pencil, eye, X]
	A1 Reading <b>Tests</b>	-	20	[Icons: pencil, eye, X]
	English assault course <b>Learning paths</b>	-	40	[Icons: pencil, eye, X]
-	<b>Subtotal</b>	-	<b>100 / 100 100</b>	
-	<b>Total</b>	-	<b>100 / 100 100</b>	

Below the table, there are buttons for 'Select all', 'Unselect all', and a 'Detail' dropdown menu.

Εικόνα 2.2: Chamilo

[[https://docs.chamilo.org/~gitbook/image?url=https%3A%2F%2F748165138-files.gitbook.io%2F%2Ffiles%2Fv0%2Fb%2Fgitbook-legacy-files%2Fo%2Fassets%252F-LH9CCYPqjHw1VKNcuiq%252F-LH9CGn0ncN\\_3G09lrQH%252F-LH9D1WEcDQexVr1znWw%252Fgraphics190.png%3Fgeneration%3D1531327443604764%26alt%3Dmedia&width=768&dpr=4&quality=100&sign=8bbcdf23&sv=2\]](https://docs.chamilo.org/~gitbook/image?url=https%3A%2F%2F748165138-files.gitbook.io%2F%2Ffiles%2Fv0%2Fb%2Fgitbook-legacy-files%2Fo%2Fassets%252F-LH9CCYPqjHw1VKNcuiq%252F-LH9CGn0ncN_3G09lrQH%252F-LH9D1WEcDQexVr1znWw%252Fgraphics190.png%3Fgeneration%3D1531327443604764%26alt%3Dmedia&width=768&dpr=4&quality=100&sign=8bbcdf23&sv=2])

Ένα από τα κύρια πλεονεκτήματα του Chamilo είναι η ενεργή κοινότητα χρηστών και προγραμματιστών που συμβάλλει συνεχώς στη βελτίωση και την επέκταση της πλατφόρμας μέσω προσθηκών (plugins) και θεμάτων (themes). Αυτό επιτρέπει την προσαρμογή της πλατφόρμας σε συγκεκριμένες ανάγκες, όπως η ενσωμάτωση με άλλα συστήματα, η υποστήριξη πολλαπλών γλωσσών και η προσαρμογή της εμφάνισης.

Το Chamilo υποστηρίζει πρότυπα διαλειτουργικότητας όπως το SCORM και το LTI, επιτρέποντας την ενσωμάτωση εξωτερικών εργαλείων και περιεχομένου. Η πλατφόρμα διαθέτει επίσης εφαρμογή για κινητές συσκευές, επιτρέποντας στους χρήστες να έχουν πρόσβαση στο εκπαιδευτικό υλικό από οπουδήποτε και οποτεδήποτε.

Το Chamilo αποτελεί μια ισχυρή και ευέλικτη λύση για τη διαχείριση της διαδικτυακής μάθησης, προσφέροντας ένα πλήρες σύνολο εργαλείων για τη δημιουργία, διανομή και αξιολόγηση εκπαιδευτικού περιεχομένου. Η ανοιχτή φύση του, σε συνδυασμό με την ενεργή κοινότητα και την

ευρεία υιοθέτησή του, το καθιστούν μια αξιόπιστη επιλογή για εκπαιδευτικά ιδρύματα και οργανισμούς που επιδιώκουν να προσφέρουν ποιοτική διαδικτυακή εκπαίδευση.

Σε σχέση με το σύστημα που υλοποιήθηκε στο πλαίσιο της παρούσας εργασίας, το Chamilo παρουσιάζει αρκετές ομοιότητες τόσο σε επίπεδο δομής όσο και σε φιλοσοφία. Και οι δύο πλατφόρμες στοχεύουν στην υποστήριξη της μαθησιακής διαδικασίας μέσω ενός οργανωμένου ψηφιακού περιβάλλοντος, στο οποίο ο φοιτητής έχει πρόσβαση σε εκπαιδευτικό υλικό και αξιολογείται μέσα από δραστηριότητες όπως quiz. Όπως και στο δικό μας σύστημα, έτσι και στο Chamilo γίνεται διαχωρισμός χρηστών σε εκπαιδευόμενους και διαχειριστές, με τον δεύτερο να έχει τη δυνατότητα δημιουργίας μαθημάτων, διαχείρισης περιεχομένου και παρακολούθησης της προόδου των φοιτητών. Επιπλέον, η χρήση δομής τύπου «μάθημα με σελίδες και quiz» είναι κοινή στα δύο περιβάλλοντα, ενώ η αξιολόγηση βασίζεται σε ερωτήσεις πολλαπλής επιλογής με αυτόματη βαθμολόγηση και καταγραφή αποτελεσμάτων. Παρόλο που το Chamilo διαθέτει ευρύτερη υποστήριξη σε χαρακτηριστικά, όπως φόρουμ, chat και SCORM συμβατότητα, η δική μας πλατφόρμα αποτελεί μια ελαφριά, ειδικά σχεδιασμένη λύση προσαρμοσμένη στο μάθημα της Ηλεκτρονικής, δίνοντας έμφαση στην καθαρότητα, την απλότητα και την παιδαγωγική στόχευση κάθε ενότητας.

## 2.3 Open edX

Το Open edX είναι μια ισχυρή, ανοιχτού κώδικα πλατφόρμα διαχείρισης μάθησης (LMS), σχεδιασμένη για την παροχή διαδικτυακής εκπαίδευσης σε μεγάλη κλίμακα. Αναπτύχθηκε αρχικά από το MIT και το Harvard το 2013, με σκοπό την υποστήριξη των Massive Open Online Courses (MOOCs), και σήμερα χρησιμοποιείται από κορυφαία πανεπιστήμια και οργανισμούς παγκοσμίως. [4]

Το Open edX αποτελείται από δύο βασικά συστατικά:

- LMS (Learning Management System): Η διεπαφή που χρησιμοποιούν οι εκπαιδευόμενοι για πρόσβαση σε μαθήματα, συμμετοχή σε συζητήσεις, υποβολή εργασιών και παρακολούθηση της προόδου τους.
- Studio: Ένα εργαλείο δημιουργίας περιεχομένου που επιτρέπει στους εκπαιδευτές να σχεδιάζουν και να διαχειρίζονται μαθήματα με ευκολία, ενσωματώνοντας κείμενα, βίντεο, κουίζ και διαδραστικά στοιχεία.

Η πλατφόρμα υποστηρίζει επίσης:

- XBlocks: Μονάδες περιεχομένου που επιτρέπουν την προσθήκη διαδραστικών στοιχείων, όπως προσομοιώσεις και παιχνίδια, ενισχύοντας την εμπειρία μάθησης.

- Insights: Ένα εργαλείο ανάλυσης που παρέχει δεδομένα σχετικά με την απόδοση των μαθητών, βοηθώντας τους εκπαιδευτές να βελτιώσουν το περιεχόμενο και τη διδασκαλία τους.
- Πιστοποιήσεις: Δυνατότητα έκδοσης πιστοποιητικών ολοκλήρωσης μαθημάτων, ενισχύοντας το κίνητρο των μαθητών.
- Χρήσεις και Εφαρμογές

**Multiple Choice**  
0.25/1 point (graded)

Which of the following countries has the largest population?

Brazil \*

Germany

Indonesia

Russia

**Submit**

---

\* Partially correct (0.25/1 point)

Εικόνα 2.3: Open edX

[[https://edx.readthedocs.io/projects/open-edx-building-and-running-a-course/en/open-release-koa.master/\\_images/partial\\_credit\\_multiple\\_choice.png](https://edx.readthedocs.io/projects/open-edx-building-and-running-a-course/en/open-release-koa.master/_images/partial_credit_multiple_choice.png)]

Το Open edX χρησιμοποιείται ευρέως σε:

- Πανεπιστήμια και Εκπαιδευτικά Ιδρύματα: Για την παροχή διαδικτυακών μαθημάτων και προγραμμάτων σπουδών.
- Επιχειρήσεις: Για την εκπαίδευση και κατάρτιση εργαζομένων.
- Κυβερνητικούς και Μη Κερδοσκοπικούς Οργανισμούς: Για την παροχή εκπαιδευτικών προγραμμάτων σε ευρύ κοινό.
- Πλεονεκτήματα και Δυνατότητες

Το Open edX προσφέρει:

- **Επεκτασιμότητα:** Δυνατότητα υποστήριξης μεγάλου αριθμού χρηστών ταυτόχρονα.
- **Προσαρμοστικότητα:** Ευκολία προσαρμογής της πλατφόρμας στις ανάγκες του οργανισμού.
- **Ανοιχτός Κώδικας:** Δυνατότητα τροποποίησης και επέκτασης της πλατφόρμας από την κοινότητα.
- **Πολυγλωσσική Υποστήριξη:** Δυνατότητα παροχής περιεχομένου σε πολλές γλώσσες.

Το Open edX αποτελεί μια ολοκληρωμένη λύση για την παροχή διαδικτυακής εκπαίδευσης, προσφέροντας εργαλεία για τη δημιουργία, διανομή και αξιολόγηση εκπαιδευτικού περιεχομένου. Η ανοιχτή φύση του, σε συνδυασμό με την ενεργή κοινότητα και την ευρεία υιοθέτησή του, το καθιστούν μια αξιόπιστη επιλογή για οργανισμούς που επιδιώκουν να προσφέρουν ποιοτική διαδικτυακή εκπαίδευση.

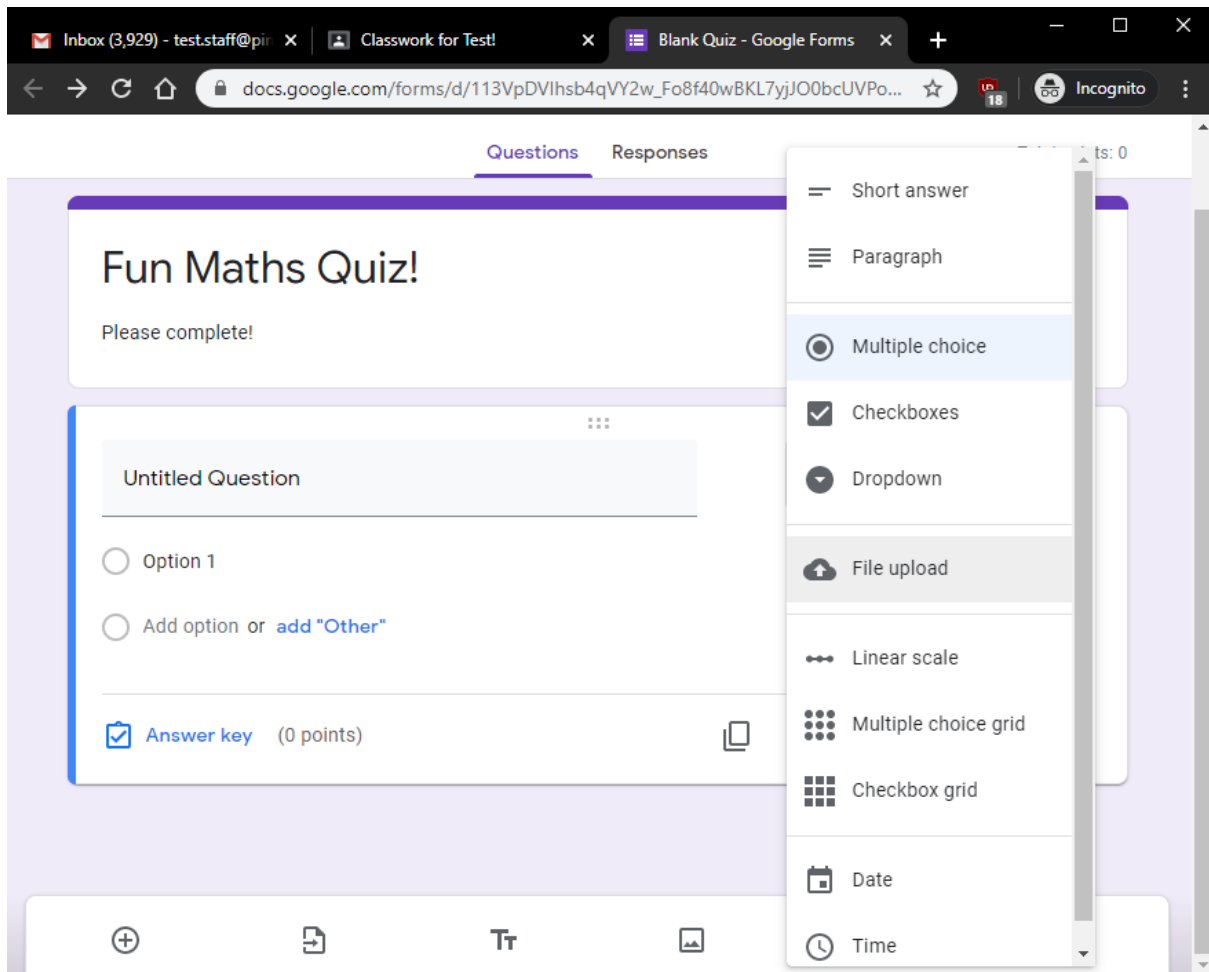
Η πλατφόρμα που υλοποιήθηκε στο πλαίσιο της παρούσας εργασίας παρουσιάζει αρκετές ομοιότητες με το Open edX, ιδιαίτερα ως προς τη φιλοσοφία και τη βασική λειτουργικότητα. Και οι δύο πλατφόρμες στοχεύουν στην παροχή οργανωμένου εκπαιδευτικού περιεχομένου, τη διαχείριση μαθημάτων και την αξιολόγηση των μαθητών μέσω κουίζ. Η χρήση εργαλείων για την παρακολούθηση της προόδου των μαθητών και η δυνατότητα προσαρμογής του περιεχομένου στις ανάγκες των εκπαιδευόμενων είναι επίσης κοινά χαρακτηριστικά.

## 2.4 Google Classroom

Το Google Classroom είναι μια δωρεάν διαδικτυακή πλατφόρμα μάθησης που αναπτύχθηκε από την Google το 2014, με στόχο την απλοποίηση της δημιουργίας, διανομής και αξιολόγησης εκπαιδευτικού υλικού σε εκπαιδευτικά ιδρύματα. Αποτελεί μέρος του Google Workspace for Education και ενσωματώνει εργαλεία όπως το Google Drive, τα Έγγραφα, τα Φύλλα, τις Παρουσιάσεις, το Gmail και το Google Meet, προσφέροντας ένα ολοκληρωμένο ψηφιακό περιβάλλον μάθησης.

Το Google Classroom επιτρέπει στους εκπαιδευτικούς να δημιουργούν εικονικές τάξεις, να αναθέτουν εργασίες, να παρέχουν ανατροφοδότηση και να παρακολουθούν την πρόοδο των μαθητών. Οι μαθητές μπορούν να υποβάλλουν τις εργασίες τους, να συμμετέχουν σε συζητήσεις και να λαμβάνουν ειδοποιήσεις για προθεσμίες και ανακοινώσεις. Η πλατφόρμα υποστηρίζει επίσης τη δημιουργία κουίζ μέσω Google Forms και την ενσωμάτωση πολυμεσικού περιεχομένου, όπως βίντεο από το YouTube.

[5]



Εικόνα 2.4: Google Classroom

[<https://pinewood.herts.sch.uk/wp-content/uploads/2020/03/Capture-61.png>]

Το Google Classroom χρησιμοποιείται ευρέως σε σχολεία και πανεπιστήμια για την υποστήριξη της εξ αποστάσεως και μικτής μάθησης. Επιτρέπει την ευέλικτη οργάνωση μαθημάτων, την εύκολη επικοινωνία μεταξύ εκπαιδευτικών και μαθητών και την αποτελεσματική διαχείριση των εκπαιδευτικών δραστηριοτήτων. Επιπλέον, η δυνατότητα πρόσβασης από διάφορες συσκευές (υπολογιστές, tablets, smartphones) καθιστά την πλατφόρμα ιδιαίτερα προσιτή.

- **Απλότητα και Ευχρηστία:** Η διεπαφή του Google Classroom είναι φιλική προς τον χρήστη, διευκολύνοντας την ταχεία υιοθέτηση από εκπαιδευτικούς και μαθητές.
- **Ενσωμάτωση με Εργαλεία της Google:** Η πλατφόρμα συνεργάζεται άψογα με άλλα εργαλεία του Google Workspace, επιτρέποντας την εύκολη διαχείριση και κοινή χρήση αρχείων.
- **Διαχείριση Βαθμολογίας:** Οι εκπαιδευτικοί μπορούν να παρακολουθούν και να διαχειρίζονται τις βαθμολογίες των μαθητών, παρέχοντας ανατροφοδότηση και αξιολογήσεις.

- Ειδοποιήσεις και Υπενθυμίσεις: Οι μαθητές λαμβάνουν ειδοποιήσεις για νέες αναθέσεις, προθεσμίες και σχόλια, διατηρώντας τους ενήμερους για τις υποχρεώσεις τους.

Το Google Classroom αποτελεί ένα ισχυρό εργαλείο για την υποστήριξη της σύγχρονης εκπαιδευτικής διαδικασίας, προσφέροντας ευελιξία, συνεργατικότητα και αποτελεσματικότητα. Η ενσωμάτωσή του με τα εργαλεία του Google Workspace και η δυνατότητα πρόσβασης από οποιαδήποτε συσκευή το καθιστούν ιδανική επιλογή για εκπαιδευτικά ιδρύματα που επιδιώκουν την ψηφιοποίηση της διδασκαλίας και της μάθησης.

Η πλατφόρμα που υλοποιήθηκε στο πλαίσιο της παρούσας εργασίας παρουσιάζει ομοιότητες με το Google Classroom, ιδιαίτερα ως προς τη δομή και τη λειτουργικότητα. Και οι δύο πλατφόρμες επιτρέπουν τη δημιουργία μαθημάτων, την ανάρτηση εκπαιδευτικού υλικού και την αξιολόγηση των μαθητών μέσω κουίζ. Επιπλέον, υποστηρίζουν την παρακολούθηση της προόδου των μαθητών και την παροχή ανατροφοδότησης. Ωστόσο, η δική μας πλατφόρμα είναι ειδικά σχεδιασμένη για το μάθημα της Ηλεκτρονικής, προσφέροντας εξειδικευμένες λειτουργίες και περιεχόμενο που ανταποκρίνονται στις ανάγκες του συγκεκριμένου γνωστικού αντικείμενου. Ενώ το Google Classroom παρέχει μια γενική λύση για διάφορα εκπαιδευτικά περιβάλλοντα, η δική μας πλατφόρμα εστιάζει στην παροχή μιας προσαρμοσμένης και εστιασμένης εμπειρίας μάθησης για τους φοιτητές της Ηλεκτρονικής.

Αξιολογώντας τα τέσσερα δημοφιλή συστήματα διαχείρισης μάθησης, το Moodle, το Chamilo, το Open edX και το Google Classroom, διαπιστώνεται ότι κάθε ένα από αυτά έχει σχεδιαστεί με συγκεκριμένες εκπαιδευτικές ανάγκες κατά νου και προσφέρει διαφορετικό εύρος δυνατοτήτων. Το Moodle είναι ίσως η πιο γνωστή και πληρέστερη πλατφόρμα ανοιχτού κώδικα, με τεράστια κοινότητα, απεριόριστες δυνατότητες επεκτασιμότητας και υποστήριξη σχεδόν κάθε εκπαιδευτικού σεναρίου. Χρησιμοποιείται ευρέως στην τριτοβάθμια εκπαίδευση και δίνει τη δυνατότητα πλήρους ελέγχου, παραμετροποίησης και αυτοφιλοξενίας. Το Chamilo, επίσης ανοιχτού κώδικα, προσφέρει παρόμοια λειτουργικότητα με το Moodle αλλά σε ένα πιο απλό και ελαφρύ περιβάλλον. Είναι ιδανικό για φορείς που χρειάζονται μια εύχρηστη και ταχύτατη λύση με λιγότερες τεχνικές απαιτήσεις, διατηρώντας όμως βασικά χαρακτηριστικά όπως μαθήματα, quiz, παρακολούθηση προόδου και εργαλεία συνεργασίας.

Το Open edX διαφέρει κυρίως λόγω της κλίμακας που υποστηρίζει. Απευθύνεται σε πανεπιστήμια και οργανισμούς που θέλουν να προσφέρουν μαθήματα μαζικής συμμετοχής (MOOCs) και διαθέτει πανίσχυρα εργαλεία ανάλυσης, σχεδίασης μαθημάτων και διαχείρισης χιλιάδων χρηστών ταυτόχρονα. Η πολυπλοκότητα της εγκατάστασης και διαχείρισης του Open edX είναι αυξημένη, όμως προσφέρει δυνατότητες που δεν συναντώνται εύκολα αλλού, όπως η παραμετροποίηση της εμπειρίας μάθησης σε μεγάλο βάθος και η διασύνδεση με εξωτερικές τεχνολογίες μέσω XBlocks. Από την άλλη πλευρά, το Google Classroom δεν είναι ανοιχτού κώδικα, αλλά παρέχεται δωρεάν μέσω του Google Workspace

for Education. Ξεχωρίζει για την απλότητά του και τη στενή του ενσωμάτωση με άλλα εργαλεία της Google, καθιστώντας το ιδανικό για σχολικές μονάδες και διδάσκοντες που θέλουν να δημιουργήσουν γρήγορα μια ψηφιακή τάξη με ελάχιστο τεχνικό κόπο.

Σε σύγκριση με τα παραπάνω, η πλατφόρμα που υλοποιήθηκε στην παρούσα εργασία αποτελεί μια πιο εξειδικευμένη και εστιασμένη λύση. Εστιάζει αποκλειστικά στο γνωστικό αντικείμενο της Ηλεκτρονικής και υλοποιεί έναν πλήρη κύκλο μάθησης: θεωρητική μελέτη, αξιολόγηση μέσω quiz, καταγραφή αποτελεσμάτων και παρακολούθηση προόδου. Αν και δεν διαθέτει την κλίμακα και τα πολύπλοκα χαρακτηριστικά των μεγάλων συστημάτων, προσφέρει ευκολία στη χρήση, στοχευμένη εμπειρία για τον φοιτητή και πλήρη έλεγχο από τον διαχειριστή, χωρίς να απαιτείται εξωτερική πλατφόρμα ή υπηρεσία.

## Κεφάλαιο 3ο: Λογισμικό και εργαλεία που χρησιμοποιήθηκαν

### 3.1 Flask

#### 3.1.1 Python

Η Python είναι μία από τις πιο δημοφιλείς και διαδεδομένες γλώσσες προγραμματισμού παγκοσμίως. Δημιουργήθηκε στις αρχές της δεκαετίας του 1990 από τον Ολλανδό προγραμματιστή Guido van Rossum και εξελίχθηκε με στόχο να είναι μια απλή, καθαρή και ευανάγνωστη γλώσσα που διευκολύνει τον προγραμματιστή να επικεντρώνεται στην επίλυση προβλημάτων. [6-10]

Η Python χαρακτηρίζεται από καθαρή σύνταξη, που μοιάζει πολύ με την αγγλική γλώσσα, γεγονός που την καθιστά ιδανική για αρχάριους αλλά και για έμπειρους προγραμματιστές που αναζητούν ευελιξία και ταχύτητα στην ανάπτυξη. Είναι γλώσσα γενικής χρήσης, πράγμα που σημαίνει ότι μπορεί να χρησιμοποιηθεί για την υλοποίηση ευρείας γκάμας εφαρμογών και συστημάτων, από απλά scripts μέχρι πολύπλοκα πληροφοριακά συστήματα, συστήματα τεχνητής νοημοσύνης και επεξεργασία δεδομένων.

Στη σύγχρονη εποχή, η Python χρησιμοποιείται σε πάρα πολλούς τομείς. Στον ακαδημαϊκό χώρο, είναι η γλώσσα που επιλέγεται συχνότερα για τη διδασκαλία βασικών μαθημάτων προγραμματισμού, καθώς και για έργα που σχετίζονται με ανάλυση δεδομένων, επεξεργασία εικόνας και επιστημονικούς υπολογισμούς. Στον επαγγελματικό τομέα, αποτελεί βασικό εργαλείο για την ανάπτυξη web εφαρμογών, αυτοματοποίηση διαδικασιών, ανάπτυξη APIs και δημιουργία διαδραστικών περιβαλλόντων. Με τη βοήθεια frameworks όπως το Flask ή το Django, η Python μπορεί να χρησιμοποιηθεί για την ανάπτυξη ολοκληρωμένων ιστοσελίδων και υπηρεσιών backend. Επίσης, με εργαλεία όπως το Pandas και το NumPy, είναι ιδανική για την ανάλυση δεδομένων, τη στατιστική και τη μηχανική μάθηση.

Η γλώσσα αυτή χρησιμοποιείται επίσης εκτεταμένα στην επιστήμη δεδομένων (data science), στην τεχνητή νοημοσύνη και στη ρομποτική, ενώ αποτελεί βασικό συστατικό λογισμικού σε εφαρμογές αυτοματισμού, στο cloud και στο Internet of Things (IoT). Οι κοινότητες ανάπτυξης ανοιχτού κώδικα τη στηρίζουν έντονα, με χιλιάδες διαθέσιμες βιβλιοθήκες και παραδείγματα που καλύπτουν σχεδόν κάθε ανάγκη.

Το γεγονός ότι η Python είναι δυναμική, διερμηνευόμενη και υψηλού επιπέδου τη διαφοροποιεί από άλλες γλώσσες όπως η C ή η Java. Ο προγραμματιστής δεν χρειάζεται να διαχειριστεί λεπτομέρειες μνήμης ή τύπων δεδομένων σε βάθος, κάτι που επιτρέπει την ταχύτερη υλοποίηση πρωτοτύπων και εφαρμογών. Η κοινότητα είναι εξαιρετικά δραστήρια και παρέχει συνεχή υποστήριξη μέσω ντοκουμέντων, tutorial, forum και βιβλιοθηκών.

Η επιλογή της Python για την υλοποίηση της παρούσας πλατφόρμας βασίστηκε σε όλα τα παραπάνω χαρακτηριστικά. Η ευκολία στη χρήση, η καθαρότητα του κώδικα και η δυνατότητα γρήγορης ανάπτυξης web εφαρμογών με τη χρήση του Flask καθιστούν την Python ιδανική για την κατασκευή εκπαιδευτικών εφαρμογών, όπου η σταθερότητα, η επεκτασιμότητα και η απλότητα είναι απαραίτητες. Μέσω της Python, υλοποιήθηκαν όλες οι βασικές λειτουργίες της πλατφόρμας, όπως η διαχείριση των χρηστών, η λειτουργία των quiz, η αποθήκευση και προβολή των αποτελεσμάτων και η επικοινωνία με τη βάση δεδομένων.

Είναι φανερό ότι η Python έχει καθιερωθεί ως μία από τις σημαντικότερες τεχνολογίες του σύγχρονου προγραμματιστικού τοπίου και η χρήση της αποτελεί πολύτιμη γνώση για κάθε φοιτητή, εκπαιδευτικό και προγραμματιστή.

Η Python είναι μια γλώσσα προγραμματισμού που έχει αναγνωριστεί παγκοσμίως για τα πολλά της πλεονεκτήματα, τα οποία την καθιστούν πρώτη επιλογή σε πολλούς τομείς της πληροφορικής. Ένα από τα βασικότερα πλεονεκτήματά της είναι η ευκολία μάθησης και χρήσης. Η σύνταξή της είναι καθαρή, διαβάζεται σχεδόν σαν φυσική γλώσσα και απαιτεί λιγότερες γραμμές κώδικα για την επίτευξη ενός αποτελέσματος σε σχέση με άλλες γλώσσες. Αυτό καθιστά την Python ιδιαίτερα κατάλληλη για αρχάριους, φοιτητές και γενικότερα για όσους ξεκινούν την ενασχόλησή τους με τον προγραμματισμό. Ένα ακόμη πλεονέκτημα είναι η τεράστια βιβλιοθήκη έτοιμων εργαλείων που διαθέτει. Για σχεδόν κάθε τομέα – από την τεχνητή νοημοσύνη και την επιστήμη δεδομένων, μέχρι το web development και τις επιστημονικές προσομοιώσεις – υπάρχουν έτοιμες βιβλιοθήκες που επιτρέπουν στον προγραμματιστή να επιταχύνει την ανάπτυξη και να επικεντρωθεί στην επίλυση του προβλήματος.

Η Python υποστηρίζει πολλαπλά προγραμματιστικά παραδείγματα, όπως αντικειμενοστραφή προγραμματισμό, διαδικαστικό και λειτουργικό, δίνοντας ευελιξία στην προσέγγιση κάθε εφαρμογής. Είναι επίσης γλώσσα ανοιχτού κώδικα, πράγμα που σημαίνει ότι είναι δωρεάν για όλους και εξελίσσεται διαρκώς μέσα από τη συνεισφορά μιας παγκόσμιας κοινότητας προγραμματιστών. Η δυνατότητα εκτέλεσης της ίδιας Python εφαρμογής σε διάφορα λειτουργικά συστήματα χωρίς αλλαγές αποτελεί ένα ακόμα σημαντικό πλεονέκτημα. Επιπλέον, η γλώσσα είναι κατάλληλη τόσο για γρήγορη ανάπτυξη πρωτοτύπων όσο και για μεγάλης κλίμακας έργα που απαιτούν σταθερότητα και επεκτασιμότητα.

Παρόλα αυτά, η Python δεν είναι χωρίς περιορισμούς. Ένα από τα πιο συχνά αναφερόμενα μειονεκτήματα είναι η σχετικά χαμηλή της ταχύτητα εκτέλεσης σε σχέση με γλώσσες όπως η C ή η Java. Καθώς είναι διερμηνευόμενη γλώσσα και όχι μεταγλωττισμένη, ο κώδικας της Python τείνει να είναι πιο αργός, κάτι που γίνεται αισθητό σε εφαρμογές που απαιτούν επεξεργασία μεγάλου όγκου δεδομένων σε πραγματικό χρόνο. Αν και υπάρχουν τρόποι βελτιστοποίησης ή ακόμα και υβριδικές λύσεις, η απόδοση μπορεί να αποτελέσει εμπόδιο σε συγκεκριμένες περιπτώσεις. Ένα ακόμα μειονέκτημα είναι η κακή της σχέση με τις εφαρμογές για κινητά τηλέφωνα. Αν και υπάρχουν εργαλεία

που υποστηρίζουν ανάπτυξη mobile εφαρμογών με Python, η γλώσσα δεν έχει επικρατήσει σε αυτόν τον τομέα, σε αντίθεση με άλλες τεχνολογίες όπως η Java, το Kotlin ή το Flutter.

Ο δυναμικός τύπος μεταβλητών που χρησιμοποιεί, ενώ διευκολύνει την ανάπτυξη, μπορεί να οδηγήσει σε σφάλματα που δεν εντοπίζονται πριν την εκτέλεση, γεγονός που την καθιστά λιγότερο ασφαλή σε εφαρμογές όπου απαιτείται αυστηρός έλεγχος δεδομένων και δομών. Παρ' όλα αυτά, τα μειονεκτήματα της Python δεν αναιρούν τη μεγάλη της αξία, ιδιαίτερα όταν αξιοποιείται στο κατάλληλο πλαίσιο. Ο συνδυασμός απλότητας, εκφραστικής δύναμης και τεράστιου οικοσυστήματος εργαλείων την καθιστά μία από τις πιο σημαντικές γλώσσες προγραμματισμού σήμερα, και η παρουσία της σε εκπαιδευτικά και ερευνητικά περιβάλλοντα θεωρείται καθιερωμένη.

Η γλώσσα προγραμματισμού Python έχει καθιερωθεί ως ένα από τα πιο ισχυρά και ευέλικτα εργαλεία στην εκπαιδευτική τεχνολογία, ιδιαίτερα στην ανάπτυξη διαδραστικών κουίζ και εφαρμογών αξιολόγησης. Η ευκολία στη σύνταξη, η πληθώρα διαθέσιμων βιβλιοθηκών και η δυνατότητα δημιουργίας τόσο απλών όσο και σύνθετων εφαρμογών την καθιστούν ιδανική για εκπαιδευτικούς και μαθητές όλων των επιπέδων.

Χρήσεις της Python στην Εκπαίδευση και τη Δημιουργία Κουίζ [11-12]

1. Ανάπτυξη Διαδραστικών Κουίζ: Η Python επιτρέπει τη δημιουργία κουίζ που μπορούν να λειτουργούν σε περιβάλλοντα γραμμής εντολών ή με γραφικό περιβάλλον χρήστη (GUI). Για παράδειγμα, χρησιμοποιώντας τη βιβλιοθήκη Tkinter, μπορούμε να αναπτύξουμε εφαρμογές με πολλαπλές ερωτήσεις και απαντήσεις, παρέχοντας άμεση ανατροφοδότηση στον χρήστη.
2. Ενσωμάτωση σε Εκπαιδευτικές Πλατφόρμες: Η Python μπορεί να χρησιμοποιηθεί για την ανάπτυξη και ενσωμάτωση κουίζ σε εκπαιδευτικές πλατφόρμες, επιτρέποντας την παρακολούθηση της προόδου των μαθητών και την προσαρμογή του περιεχομένου ανάλογα με τις ανάγκες τους .
3. Αυτοματοποιημένη Δημιουργία Ερωτήσεων: Με εργαλεία όπως το pygiftgenerator, είναι δυνατή η αυτόματη δημιουργία μεγάλου αριθμού ερωτήσεων για κουίζ, τα οποία μπορούν να εισαχθούν σε πλατφόρμες όπως το Moodle, διευκολύνοντας τους εκπαιδευτικούς στη δημιουργία ποικίλου και προσαρμοσμένου περιεχομένου.
4. Ενίσχυση της Μάθησης μέσω Παιχνιδιών: Η Python επιτρέπει τη δημιουργία εκπαιδευτικών παιχνιδιών που ενσωματώνουν κουίζ, καθιστώντας τη μάθηση πιο ελκυστική και διαδραστική για τους μαθητές .
5. Προσαρμοσμένες Εκπαιδευτικές Εφαρμογές: Η ευελιξία της Python επιτρέπει την ανάπτυξη εφαρμογών που προσαρμόζονται στις ανάγκες διαφορετικών μαθησιακών επιπέδων, παρέχοντας εξατομικευμένη εμπειρία μάθησης.

## Πλεονεκτήματα της Χρήσης Python στην Εκπαίδευση

- **Ευκολία Μάθησης:** Η απλή και κατανοητή σύνταξη της Python την καθιστά ιδανική για αρχάριους προγραμματιστές και μαθητές.
- **Ευρεία Υποστήριξη:** Η μεγάλη κοινότητα χρηστών και η πληθώρα διαθέσιμων βιβλιοθηκών διευκολύνουν την ανάπτυξη εκπαιδευτικών εφαρμογών.
- **Διαλειτουργικότητα:** Η Python μπορεί να ενσωματωθεί με άλλες τεχνολογίες και πλατφόρμες, επιτρέποντας την ανάπτυξη ολοκληρωμένων εκπαιδευτικών λύσεων.
- **Ανοιχτός Κώδικας:** Η φύση του ανοιχτού κώδικα επιτρέπει την προσαρμογή και επέκταση των εφαρμογών ανάλογα με τις ανάγκες των εκπαιδευτικών και των μαθητών.

Η Python αποτελεί ένα πολύτιμο εργαλείο στην εκπαιδευτική τεχνολογία, προσφέροντας ευελιξία, προσβασιμότητα και δυνατότητες προσαρμογής που ενισχύουν τη μαθησιακή εμπειρία. Η χρήση της στη δημιουργία κουίζ και εκπαιδευτικών εφαρμογών συμβάλλει στην ανάπτυξη διαδραστικών και αποτελεσματικών μεθόδων διδασκαλίας, προσαρμοσμένων στις ανάγκες των μαθητών.

Το Flask είναι ένα δημοφιλές και ευρέως χρησιμοποιούμενο framework για την ανάπτυξη διαδικτυακών εφαρμογών με τη γλώσσα προγραμματισμού Python. Δημιουργήθηκε το 2010 από τον Armin Ronacher και σχεδιάστηκε ώστε να προσφέρει ευελιξία, απλότητα και επεκτασιμότητα. Ανήκει στην κατηγορία των micro-frameworks, πράγμα που σημαίνει ότι παρέχει μόνο τα βασικά εργαλεία για τη δημιουργία μιας web εφαρμογής, δίνοντας στον προγραμματιστή τον πλήρη έλεγχο ως προς το πώς θα επεκτείνει και θα διαμορφώσει το σύστημά του.

Το Flask προσφέρει βασικές λειτουργίες για τη δημιουργία διαδρομών (routes), τη διαχείριση αιτημάτων (requests), την απόκριση (response), την απόδοση HTML μέσω templates (συχνά με τη μηχανή Jinja2), καθώς και τη σύνδεση με βάσεις δεδομένων. Αν και ελαφρύ από προεπιλογή, μπορεί να επεκταθεί εύκολα με βιβλιοθήκες για επαλήθευση χρηστών, διαχείριση αρχείων, αποστολή email, χρήση APIs και άλλες λειτουργίες που χρειάζεται μια σύγχρονη web εφαρμογή.

Ένα από τα σημαντικά πλεονεκτήματα του Flask είναι η απλότητά του. Η φιλοσοφία του βασίζεται στην αρχή "keep it simple", δίνοντας στον προγραμματιστή την ευκαιρία να ξεκινήσει γρήγορα με ένα απλό αρχείο Python και να προσθέτει λειτουργίες σταδιακά. Αυτή η προσέγγιση είναι ιδανική για φοιτητές, ερευνητές, μικρές ομάδες ανάπτυξης ή όσους θέλουν να δημιουργήσουν μία εφαρμογή χωρίς το βάρος μιας πολύπλοκης υποδομής. Ταυτόχρονα, δεν περιορίζει τον χρήστη από το να δημιουργήσει πολύπλοκα έργα, καθώς είναι πλήρως επεκτάσιμο και μπορεί να συνδυαστεί με ORM εργαλεία όπως

SQLAlchemy για σύνθετες βάσεις δεδομένων ή βιβλιοθήκες όπως το Flask-Login για διαχείριση ταυτοποίησης χρηστών.

Το Flask είναι ιδιαίτερα χρήσιμο στην εκπαίδευση, καθώς επιτρέπει στους φοιτητές να κατανοήσουν σε βάθος την αρχιτεκτονική μιας web εφαρμογής, να διαχειριστούν τις διαδρομές, τα δεδομένα και τις απαντήσεις με διαφάνεια, και να εξοικειωθούν με έννοιες όπως HTTP requests, templates, φόρμες και συνεδρίες. Επιπλέον, χρησιμοποιείται συχνά σε μαθήματα πληροφορικής για την ανάπτυξη RESTful APIs και μικρών διαδικτυακών εφαρμογών που μπορούν να φιλοξενηθούν τοπικά ή σε υπηρεσίες όπως το Heroku και το Render.

Στην παρούσα εργασία, το Flask αποτέλεσε τη βάση για την υλοποίηση της διαχειριστικής λογικής της πλατφόρμας μάθησης. Μέσα από αυτό, ορίστηκαν όλες οι βασικές λειτουργίες, όπως η δημιουργία θεματικών ενοτήτων, η διαχείριση των χρηστών, η αποστολή και επεξεργασία δεδομένων quiz, η καταγραφή των απαντήσεων και η εμφάνιση των αποτελεσμάτων. Η απλότητα με την οποία το Flask διαχειρίζεται τις φόρμες, τα session cookies, τη σύνδεση με τη βάση δεδομένων και τα templates HTML το καθιστά ιδανικό για την ανάπτυξη ενός προσαρμοσμένου εκπαιδευτικού συστήματος, όπως αυτό που υλοποιήθηκε. Επιπλέον, το γεγονός ότι πρόκειται για open-source framework, ενισχύει τις δυνατότητες μελλοντικής επέκτασης και προσαρμογής της εφαρμογής ανάλογα με τις ανάγκες.

Το Flask είναι ένα ισχυρό και ευέλικτο εργαλείο για την ανάπτυξη web εφαρμογών, ιδιαίτερα κατάλληλο για φοιτητικά έργα και εξειδικευμένες εφαρμογές όπου απαιτείται έλεγχος, απλότητα και ταχύτητα στην υλοποίηση. Η χρήση του στο πλαίσιο της παρούσας εργασίας απέδειξε στην πράξη την αποτελεσματικότητά του και επιβεβαίωσε ότι η Python σε συνδυασμό με το Flask αποτελεί έναν σύγχρονο και αξιόπιστο συνδυασμό για την ανάπτυξη διαδικτυακών συστημάτων εκπαίδευσης.

### **3.2 Βάση δεδομένων MySQL**

Η MySQL είναι ένα από τα πιο δημοφιλή συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων παγκοσμίως και χρησιμοποιείται ευρέως σε διαδικτυακές εφαρμογές, εμπορικά πληροφοριακά συστήματα, πλατφόρμες περιεχομένου και σε κάθε είδους έργο που απαιτεί αξιόπιστη αποθήκευση και ανάκτηση δεδομένων. Αναπτύχθηκε τη δεκαετία του 1990 από τη σουηδική εταιρεία MySQL AB και στη συνέχεια αποκτήθηκε από τη Sun Microsystems και αργότερα από την Oracle. Είναι λογισμικό ανοιχτού κώδικα, γεγονός που έχει συμβάλει στη μεγάλη του διάδοση και στην ενσωμάτωσή του σε χιλιάδες έργα σε όλο τον κόσμο. [13]

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, το οποίο σημαίνει ότι τα δεδομένα οργανώνονται σε πίνακες και οι συσχετίσεις μεταξύ τους γίνονται μέσω σχέσεων (relations). Οι πίνακες είναι δομημένοι με γραμμές και στήλες, όπου κάθε στήλη αντιστοιχεί σε ένα

χαρακτηριστικό και κάθε γραμμή σε μία εγγραφή. Η αλληλεπίδραση με τη βάση γίνεται μέσω της γλώσσας SQL (Structured Query Language), με την οποία μπορεί κανείς να κάνει εισαγωγή, ανάκτηση, επεξεργασία και διαγραφή δεδομένων, αλλά και να ορίσει σχέσεις, περιορισμούς και δείκτες (indexes).

Ένα από τα βασικά πλεονεκτήματα της MySQL είναι η αξιοπιστία και η ταχύτητα. Είναι γνωστή για την υψηλή της απόδοση ακόμη και σε εφαρμογές με μεγάλο όγκο δεδομένων ή υψηλό αριθμό χρηστών. Υποστηρίζει πολλαπλούς χρήστες ταυτόχρονα, μηχανισμούς ασφάλειας, διαχείριση εξουσιοδοτήσεων και δυνατότητες αντιγράφων ασφαλείας. Επιπλέον, είναι πλήρως συμβατή με πολλές γλώσσες προγραμματισμού και frameworks, μεταξύ αυτών και η Python μέσω συνδέσμων όπως το `mysql-connector-python` ή η βιβλιοθήκη `PyMySQL`.

Η χρήση της MySQL στην παρούσα εργασία υπήρξε κομβικής σημασίας για την υποστήριξη της λειτουργικότητας της πλατφόρμας. Στη βάση δεδομένων αποθηκεύονται όλα τα απαραίτητα δεδομένα της εφαρμογής: πληροφορίες χρηστών, θεματικές ενότητες, σελίδες θεωρίας, ερωτήσεις quiz, απαντήσεις χρηστών και αποτελέσματα αξιολόγησης. Μέσα από κατάλληλους πίνακες και σχέσεις, δημιουργήθηκε μια δομή που υποστηρίζει την οργάνωση των μαθημάτων και την αλληλεπίδραση του χρήστη με το σύστημα. Η σύνδεση της βάσης με την εφαρμογή Flask επιτρέπει την ανάκτηση και αποθήκευση δεδομένων σε πραγματικό χρόνο, διασφαλίζοντας την ομαλή λειτουργία της πλατφόρμας.

Η MySQL επιλέχθηκε τόσο για την απόδοσή της όσο και για την ευκολία με την οποία ενσωματώνεται σε εκπαιδευτικά έργα. Η απλή της σύνταξη σε συνδυασμό με τη σταθερότητά της την καθιστούν ιδανική για εφαρμογές όπου η ακρίβεια, η συνοχή και η ακεραιότητα των δεδομένων είναι κρίσιμες. Επίσης, η δυνατότητα εξαγωγής και ελέγχου των δεδομένων μέσω εργαλείων όπως το `phpMyAdmin` ή η γραμμή εντολών προσφέρουν σημαντική ευκολία κατά την ανάπτυξη, τη δοκιμή και τη συντήρηση του έργου.

Η MySQL είναι μια ισχυρή, σταθερή και καθιερωμένη λύση για διαχείριση βάσεων δεδομένων, η οποία εξυπηρέτησε με επιτυχία τις ανάγκες της παρούσας εργασίας. Η συμβολή της στη διαχείριση των δεδομένων της πλατφόρμας υπήρξε καθοριστική και αποτελεί πρότυπο για παρόμοιες εκπαιδευτικές εφαρμογές.

*Ο παρακάτω κώδικας συνδέεται με μια βάση δεδομένων MySQL, αποθηκεύει νέους χρήστες μέσω φόρμας και εμφανίζει όλους τους καταχωρημένους χρήστες στην ίδια σελίδα. Αποτελεί ένα απλό αλλά χαρακτηριστικό παράδειγμα χρήσης του Flask με backend βάση δεδομένων.*

```
# Εισαγωγή βασικών modules του Flask
from flask import Flask, request, render_template

# Εισαγωγή της βιβλιοθήκης mysql.connector για σύνδεση με MySQL
import mysql.connector
```

```

# Δημιουργία αντικειμένου εφαρμογής Flask
app = Flask(__name__)

# Συνάρτηση που επιστρέφει σύνδεση με τη βάση δεδομένων MySQL
def get_connection():
    return mysql.connector.connect(
        host="localhost",    # Όνομα του server βάσης (τοπικά)
        user="root",        # Χρήστης της βάσης
        password="your_password", # Κωδικός πρόσβασης (να αλλάξει αναλόγως)
        database="mydb"     # Όνομα της βάσης που χρησιμοποιείται
    )

# Ορισμός διαδρομής για την αρχική σελίδα
@app.route('/', methods=['GET', 'POST'])
def index():
    # Δημιουργία σύνδεσης με τη βάση και αρχικοποίηση του cursor
    conn = get_connection()
    cursor = conn.cursor()

    # Αν η φόρμα υποβληθεί με POST, αποθηκεύονται τα δεδομένα
    if request.method == 'POST':
        name = request.form['name'] # Λήψη του ονόματος από τη φόρμα
        email = request.form['email'] # Λήψη του email από τη φόρμα

        # Εισαγωγή νέας εγγραφής στον πίνακα users
        cursor.execute("INSERT INTO users (fullname, email) VALUES (%s, %s)", (name, email))
        conn.commit() # Οριστικοποίηση αλλαγών στη βάση

    # Εκτέλεση ερωτήματος για εμφάνιση όλων των χρηστών
    cursor.execute("SELECT fullname, email FROM users")
    users = cursor.fetchall() # Αποθήκευση των αποτελεσμάτων σε λίστα

    # Κλείσιμο της σύνδεσης με τη βάση
    conn.close()

    # Επιστροφή του HTML template και εμφάνιση των χρηστών
    return render_template('users.html', users=users)

# Εκκίνηση της εφαρμογής Flask σε debug mode

```

```
if __name__ == '__main__':  
    app.run(debug=True)
```

και το αρχείο users.html με αναλυτικά σχόλια σε κάθε σημείο

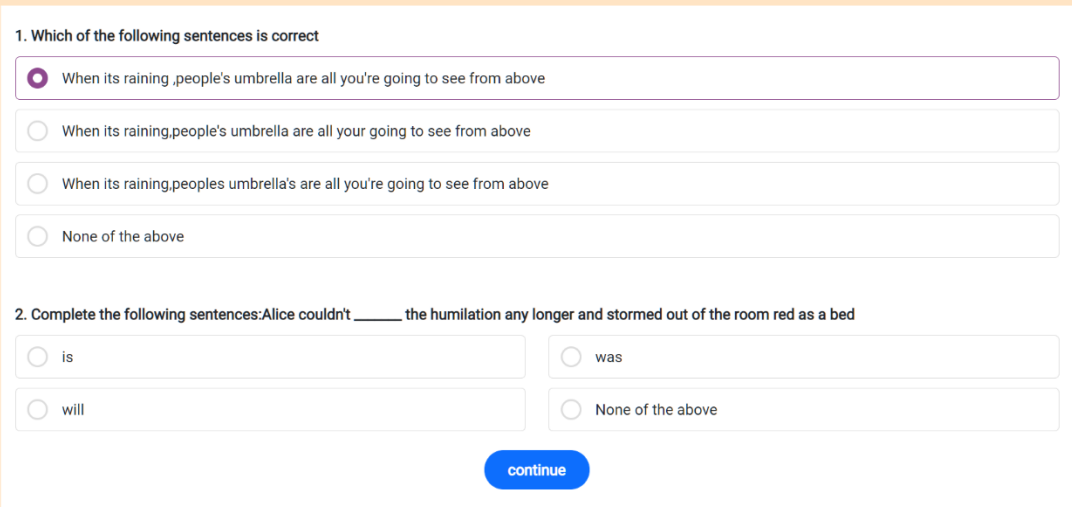
```
<!DOCTYPE html>  
<html lang="el">  
<head>  
    <meta charset="UTF-8">  
    <title>Λίστα Χρηστών</title>  
</head>  
<body>  
  
    <!-- Τίτλος σελίδας -->  
    <h1>Καταχωρημένοι Χρήστες</h1>  
  
    <!-- Λίστα που εμφανίζει τους χρήστες από τη βάση -->  
    <ul>  
        {% for user in users %}  
            <!-- Εμφανίζεται το ονοματεπώνυμο και το email κάθε χρήστη -->  
            <li>{{ user[0] }} - {{ user[1] }}</li>  
        {% endfor %}  
    </ul>  
  
    <!-- Φόρμα καταχώρησης νέου χρήστη -->  
    <h2>Προσθήκη Χρήστη</h2>  
    <form method="POST" action="/">  
        <label>Ονοματεπώνυμο:</label>  
        <input type="text" name="name" required><br><br>  
  
        <label>Email:</label>  
        <input type="email" name="email" required><br><br>  
  
        <input type="submit" value="Αποθήκευση">  
    </form>  
</body>
```

</html>

Η σελίδα users.html προβάλλει όλους τους χρήστες που υπάρχουν στη βάση (users), και ταυτόχρονα περιλαμβάνει μια φόρμα μέσω της οποίας ο επισκέπτης μπορεί να προσθέσει νέον χρήστη. Όλα τα δεδομένα περνούν από το Flask backend.

### 3.3 Bootstrap 5

Το Bootstrap είναι ένα από τα πιο διαδεδομένα και ευρέως χρησιμοποιούμενα frameworks για την ανάπτυξη διεπαφών χρήστη (User Interfaces – UI) σε ιστοσελίδες και web εφαρμογές. Η έκδοση 5, που κυκλοφόρησε επίσημα το 2021, έφερε σημαντικές βελτιώσεις τόσο σε απόδοση όσο και σε ευελιξία. Πρόκειται για ένα ανοιχτού κώδικα εργαλείο, το οποίο δημιουργήθηκε από την ομάδα του Twitter το 2011 και έκτοτε εξελίσσεται συνεχώς, παρέχοντας πρότυπα εμφάνισης, σχεδίασης και συμπεριφοράς για κουμπιά, φόρμες, πίνακες, πλοηγήσεις, ειδοποιήσεις και άλλα βασικά στοιχεία μιας ιστοσελίδας. [14]



1. Which of the following sentences is correct

When its raining ,people's umbrella are all you're going to see from above

When its raining,people's umbrella are all your going to see from above

When its raining,peoples umbrella's are all you're going to see from above

None of the above

2. Complete the following sentences:Alice couldn't \_\_\_\_\_ the humiliation any longer and stormed out of the room red as a bed

is

was

will

None of the above

continue

Εικόνα 3.1: Bootstrap Quiz

[<https://bbbootstrap.github.io/bbimages/images/CjZzE30.png>]

Το Bootstrap προσφέρει ένα ολοκληρωμένο σύνολο προκαθορισμένων κλάσεων CSS και έτοιμων JavaScript συστατικών (components), τα οποία επιτρέπουν στον προγραμματιστή να κατασκευάσει όμορφες και λειτουργικές ιστοσελίδες χωρίς να απαιτείται προηγμένη γνώση γραφικών σχεδιαστικών

τεχνικών. Το πιο σημαντικό ίσως χαρακτηριστικό του Bootstrap είναι ότι υποστηρίζει με φυσικό τρόπο τη δημιουργία responsive ιστοσελίδων. Αυτό σημαίνει ότι η διάταξη και το περιεχόμενο προσαρμόζονται αυτόματα σε διαφορετικές διαστάσεις οθόνης, από υπολογιστές και tablet έως κινητά τηλέφωνα.

Η έκδοση 5 του Bootstrap διευρύνει τις δυνατότητες αυτές, αφαιρώντας την εξάρτηση από την βιβλιοθήκη jQuery και επιτρέποντας την ευκολότερη ενσωμάτωση με σύγχρονες JavaScript εφαρμογές. Περιλαμβάνει επίσης βελτιώσεις στη διαχείριση πλεγμάτων (grid system), πιο ευέλικτη τυπογραφία, προσαρμοσμένα χρώματα και μεταβλητές CSS, καθώς και νέες κλάσεις για πιο λεπτομερή έλεγχο της διάταξης και της εμφάνισης στοιχείων.

Στην παρούσα εργασία, το Bootstrap 5 χρησιμοποιήθηκε για να προσδώσει στην πλατφόρμα μάθησης μια καθαρή και λειτουργική εμφάνιση. Όλα τα βασικά στοιχεία της διεπαφής, όπως η κεφαλίδα, τα κουμπιά, οι πίνακες αποτελεσμάτων, οι φόρμες σύνδεσης και οι λίστες θεματικών ενοτήτων, έχουν διαμορφωθεί με χρήση κλάσεων του Bootstrap. Με αυτόν τον τρόπο επιτυγχάνεται ενιαία αισθητική, χωρίς την ανάγκη για χειροκίνητη σχεδίαση με CSS από την αρχή.

Η χρήση του Bootstrap επιτρέπει επίσης την ταχύτερη ανάπτυξη και ευκολότερη συντήρηση του κώδικα HTML. Αντί να γράφονται πολύπλοκα styles για κάθε στοιχείο, ο προγραμματιστής μπορεί να χρησιμοποιεί έτοιμες κλάσεις όπως btn, container, form-control, row, col, table, κ.ά. Αυτό καθιστά την ανάπτυξη πιο αποδοτική, ενώ η καθολική υποστήριξη από όλους τους σύγχρονους browsers εγγυάται την ομαλή λειτουργία της πλατφόρμας.

Επιπλέον, μέσω των utility classes που παρέχει το Bootstrap, η μορφοποίηση γίνεται πιο εύκολη και επαναχρησιμοποιήσιμη. Για παράδειγμα, μπορούν να προστεθούν περιθώρια (mt-3, mb-2) ή χρώματα (bg-primary, text-white) με μία μόνο κλάση, χωρίς να γραφτεί καθόλου επιπλέον CSS. Αυτή η λογική βοηθά σημαντικά σε φοιτητικά έργα, όπου η ευκολία και η συνέπεια στον σχεδιασμό είναι εξίσου σημαντικές με τη λειτουργικότητα.

Το Bootstrap προσφέρει εξαιρετική τεκμηρίωση, με παραδείγματα, οδηγούς και ζωντανά preview, γεγονός που το καθιστά ιδανικό για φοιτητές και εκπαιδευτικούς που θέλουν να αναπτύξουν γρήγορα μια web εφαρμογή χωρίς να ασχοληθούν με τη δημιουργία γραφικών από την αρχή. Στην πλατφόρμα που παρουσιάζεται σε αυτή την εργασία, το Bootstrap 5 αποτέλεσε βασικό εργαλείο σχεδίασης του frontend και συνέβαλε στην κατασκευή μιας εύχρηστης, αισθητικά συνεπούς και πλήρως responsive εφαρμογής μάθησης.

# Κεφάλαιο 4ο: Διαδικτυακή πλατφόρμα μάθησης για την ηλεκτρονική

## 4.1 Περιγραφή

Το σύστημα που παρουσιάζεται αποτελεί μια διαδικτυακή πλατφόρμα μάθησης με αντικείμενο την Ηλεκτρονική, σχεδιασμένη με στόχο την υποστήριξη της εκπαιδευτικής διαδικασίας σε περιβάλλοντα ανώτατης εκπαίδευσης. Η πλατφόρμα προσφέρει οργανωμένο και πολυεπίπεδο περιεχόμενο, το οποίο περιλαμβάνει θεωρητικό υλικό, διαγράμματα, εκπαιδευτικά βίντεο, καθώς και αξιολογήσεις μέσω quiz. Ο χρήστης έχει τη δυνατότητα να μελετήσει, να προετοιμαστεί και να ελεγχθεί με αυτοματοποιημένο τρόπο, μέσα από ένα φιλικό και λειτουργικό περιβάλλον.

Στην πλατφόρμα υπάρχουν δύο βασικοί ρόλοι: ο διαχειριστής (ή εκπαιδευτικός) και ο φοιτητής. Ο διαχειριστής είναι υπεύθυνος για τη δημιουργία θεματικών ενοτήτων, τη διαχείριση των σελίδων θεωρίας, των quiz και των χρηστών. Μπορεί να οργανώνει το υλικό, να δημιουργεί νέες ερωτήσεις και να παρακολουθεί την πρόοδο των συμμετεχόντων. Ο φοιτητής από την άλλη πλευρά έχει πρόσβαση σε θεματικές ενότητες ανάλογα με το επίπεδο προόδου του, μελετά το υλικό, συμμετέχει σε quiz και βλέπει τα αποτελέσματά του μέσα από ένα ξεκάθαρο ιστορικό.

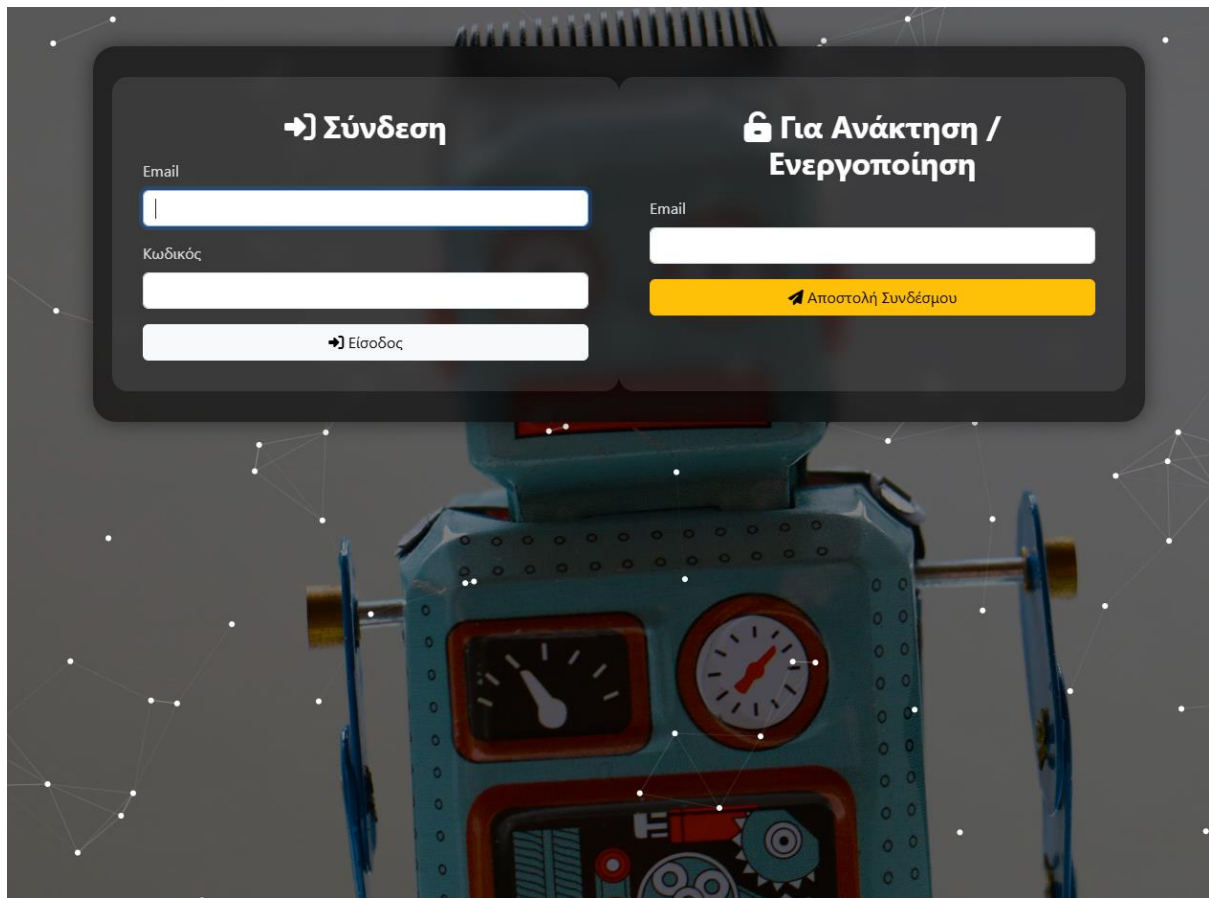
Κάθε θεματική ενότητα περιλαμβάνει οργανωμένες σελίδες με θεωρία, οι οποίες μπορεί να περιλαμβάνουν απλό κείμενο, πολυμέσα, συνδέσμους και ασκήσεις. Ο φοιτητής διαβάζει το υλικό και, όταν νιώσει έτοιμος, προχωρά στην αξιολόγηση μέσω quiz. Τα αποτελέσματα αποθηκεύονται και παρουσιάζονται τόσο με γραφική όσο και με αναλυτική μορφή, ώστε να ενισχύεται η εκπαιδευτική ανατροφοδότηση.

Η πλατφόρμα υλοποιήθηκε με χρήση της γλώσσας προγραμματισμού Python και του framework Flask, ενώ αξιοποιήθηκαν HTML, CSS και Bootstrap για το frontend, καθώς και MySQL για την αποθήκευση των δεδομένων. Η δομή της επιτρέπει την επέκταση με νέες λειτουργίες και την προσαρμογή της σε διαφορετικά γνωστικά αντικείμενα.

Η δημιουργία μιας τέτοιας πλατφόρμας ενισχύει τη δυνατότητα αυτοκατευθυνόμενης μάθησης, επιτρέπει στους φοιτητές να προοδεύουν με τον δικό τους ρυθμό και προσφέρει στον εκπαιδευτικό πλήρη εικόνα της συμμετοχής και της προόδου κάθε φοιτητή.

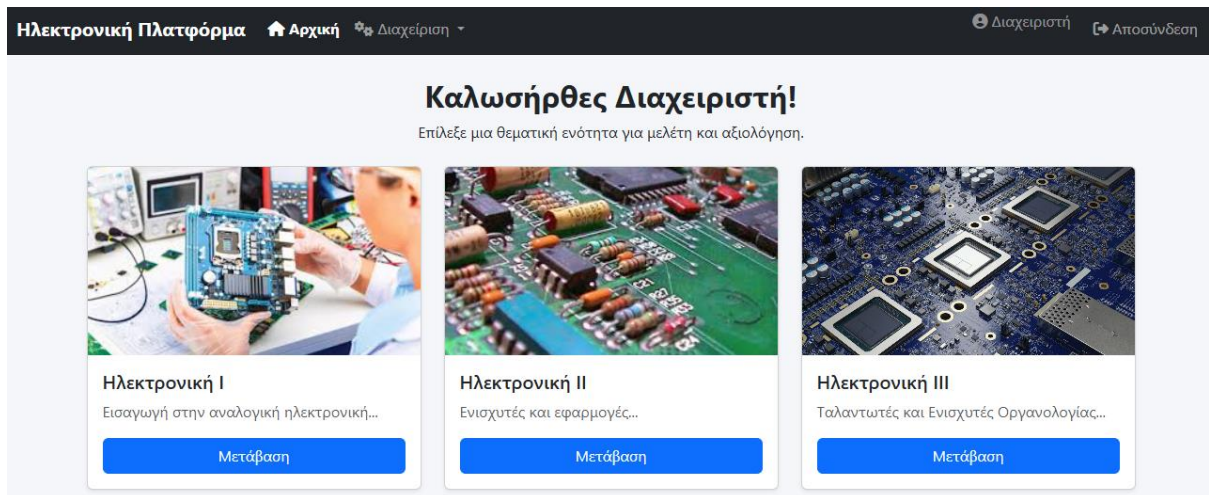
Στην εικόνα 4.1 φαίνεται η αρχική σελίδα σύνδεσης και ενεργοποίησης λογαριασμού της πλατφόρμας. Ο χρήστης καλείται να εισάγει τα στοιχεία του προκειμένου να αποκτήσει πρόσβαση στις λειτουργίες του συστήματος. Η σελίδα περιλαμβάνει πεδία για την εισαγωγή email και κωδικού πρόσβασης, ενώ δίνεται η δυνατότητα ενεργοποίησης του λογαριασμού μέσω επιβεβαίωσης. Ενσωματώνει βασικά στοιχεία ασφάλειας και επαλήθευσης, ώστε να διασφαλίζεται η πρόσβαση μόνο σε εξουσιοδοτημένους

φοιτητές ή εκπαιδευτικούς. Η λειτουργικότητα αυτής της σελίδας είναι κρίσιμη για την αυθεντικοποίηση των χρηστών και την ασφαλή χρήση του συστήματος.



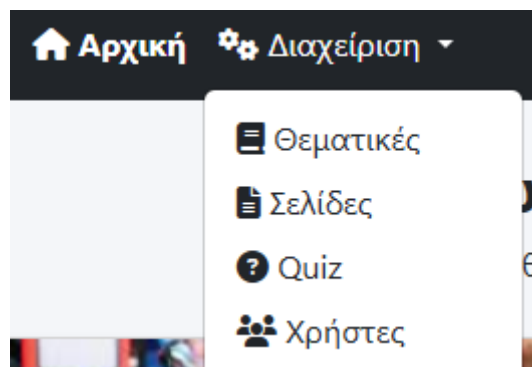
Εικόνα 4.1: Σελίδα σύνδεσης και ενεργοποίησης λογαριασμού

Στην εικόνα 4.2 φαίνεται η κεντρική σελίδα των θεματικών ενοτήτων, τις οποίες μπορεί να μελετήσει ο χρήστης και να αξιολογηθεί πάνω στο περιεχόμενό τους. Κάθε ενότητα παρουσιάζεται με τίτλο και συνοπτική περιγραφή, δίνοντας μια πρώτη εικόνα του περιεχομένου που καλύπτει. Από εδώ ξεκινά ο χρήστης την πλοήγηση στο υλικό που σχετίζεται με κάθε επιμέρους θέμα, όπως θεωρία, ασκήσεις, ή quiz. Η παρουσίαση των θεματικών είναι ομαδοποιημένη και εύκολα προσβάσιμη, ώστε να εξυπηρετεί διαφορετικά επίπεδα γνώσεων ή προόδου. Η συγκεκριμένη σελίδα αποτελεί βασικό σημείο πλοήγησης μέσα στην πλατφόρμα και βοηθά τον χρήστη να οργανώσει το διάβασμά του ανάλογα με την πορεία του στην ύλη.



Εικόνα 4.2: Θεματικές ενότητες για μελέτη και αξιολόγηση

Στην εικόνα 4.3 φαίνεται το μενού του διαχειριστή, το οποίο του επιτρέπει να έχει πρόσβαση σε όλες τις βασικές λειτουργίες διαχείρισης της πλατφόρμας. Μέσα από αυτό το μενού, ο admin μπορεί να πλοηγηθεί στις ενότητες των μαθημάτων, να διαχειριστεί χρήστες, quiz, σελίδες περιεχομένου και άλλες ρυθμίσεις.



Εικόνα 4.3: Μενού του διαχειριστή

Στην εικόνα 4.4 παρουσιάζεται η δυνατότητα του διαχειριστή να βλέπει συγκεντρωτικά όλες τις σελίδες που ανήκουν σε κάθε θεματική ενότητα. Η λειτουργία αυτή είναι ιδιαίτερα χρήσιμη για την παρακολούθηση και την αναδιοργάνωση του υλικού ανά θέμα. Ο admin μπορεί να επιλέξει μια θεματική και να δει ακριβώς ποιες σελίδες έχουν δημιουργηθεί, ποιο είναι το περιεχόμενό τους και να προχωρήσει σε ενέργειες όπως επεξεργασία ή διαγραφή.

Σελίδα 1

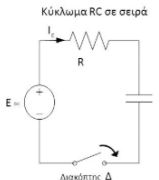
Σελίδα 2

Σελίδα 3

Σελίδα 4

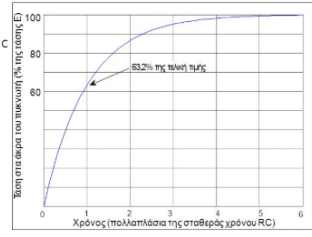
Ξεκίνα Quiz Αξιολόγησης

### Κύκλωμα συνεχούς ρεύματος RC - Φόρτιση πυκνωτή (3/4)



Κύκλωμα RC σε σειρά

Τάση στα άκρα του πυκνωτή:  
 $\rightarrow V_c(t) = E \times (1 - e^{-\frac{t}{RC}})$









Σελίδα 1  
RC κύκλωμα φόρτισης πυκνωτή.

Εικόνα 4.4: Ο admin μπορεί να δει το σύνολο των σελίδων για τη κάθε θεματική

Στην εικόνα 4.5 απεικονίζεται η λίστα όλων των θεματικών ενοτήτων που έχουν καταχωρηθεί στο σύστημα. Κάθε θεματική εμφανίζεται με τίτλο και επιμέρους στοιχεία, ενώ δίπλα της υπάρχουν επιλογές για διαχείριση όπως επεξεργασία, διαγραφή ή προσθήκη νέου περιεχομένου. Η προβολή της λίστας επιτρέπει την εποπτεία του συνόλου των διαθέσιμων θεματικών.

### Λίστα Θεματικών Ενοτήτων

+ Νέα Θεματική

ID	Τίτλος	Επίπεδο	Απαιτούμενο %	Ενέργειες
1	Ηλεκτρονική I	1	70.0%	 
2	Ηλεκτρονική II	2	70.0%	 
3	Ηλεκτρονική III	3	70.0%	 

Εικόνα 4.5: Λίστα Θεματικών Ενοτήτων

Στην εικόνα 4.6 εμφανίζεται η φόρμα δημιουργίας νέας θεματικής ενότητας. Ο διαχειριστής μπορεί να καταχωρήσει τίτλο και περιγραφή για την ενότητα που επιθυμεί να προσθέσει στην πλατφόρμα. Μέσω αυτής της λειτουργίας, το σύστημα επεκτείνεται συνεχώς με νέο υλικό. Η διαδικασία είναι απλή και επικεντρώνεται στο να προσφέρει ευελιξία και ταχύτητα κατά την προσθήκη νέου περιεχομένου.

### Νέα Θεματική Ενότητα

Τίτλος

Περιγραφή

URL Εικόνας

Επίπεδο (Level)

Βαθμολογία Ξεκλειδώματος (%)

**Δημιουργία**

Εικόνα 4.6: Σελίδα για Νέα Θεματική Ενότητα

Στην εικόνα 4.7 φαίνεται η σελίδα που χρησιμοποιείται για την επεξεργασία ήδη υπάρχουσας θεματικής ενότητας. Ο διαχειριστής έχει τη δυνατότητα να αλλάξει τον τίτλο, την περιγραφή ή άλλες βασικές πληροφορίες της θεματικής. Η δυνατότητα επεξεργασίας επιτρέπει την επικαιροποίηση και τη διόρθωση του περιεχομένου, ώστε αυτό να ανταποκρίνεται στις ανάγκες του μαθήματος και των φοιτητών.

### Επεξεργασία Θεματική Ενότητα

Τίτλος

Περιγραφή

URL Εικόνας

Επίπεδο (Level)

Βαθμολογία Ξεκλειδώματος (%)

**Αποθήκευση Αλλαγών**

Εικόνα 4.7: Σελίδα για Επεξεργασία Θεματικής Ενότητας

Στην εικόνα 4.8 παρουσιάζεται η συνολική λίστα των σελίδων που έχουν δημιουργηθεί για τις διάφορες θεματικές ενότητες της πλατφόρμας. Κάθε γραμμή αντιστοιχεί σε μια σελίδα θεωρίας ή ασκήσεων και συνοδεύεται από βασικές πληροφορίες, όπως ο τίτλος και η θεματική στην οποία ανήκει. Ο

διαχειριστής μπορεί να επεξεργαστεί, να διαγράψει ή να δει προεπισκόπηση της σελίδας. Η προβολή αυτής της λίστας βοηθά στη γρήγορη διαχείριση μεγάλου όγκου υλικού.

Λίστα Όλων των Σελίδων Θεματικών Ενότητων

+ Νέα Σελίδα

ID	Τopic	Σελίδα	Ενέργεια
1	Ηλεκτρονική I (ID 1)	Σελίδα 1	<a href="#">Επεξεργασία</a>
2	Ηλεκτρονική I (ID 1)	Σελίδα 2	<a href="#">Επεξεργασία</a>
3	Ηλεκτρονική I (ID 1)	Σελίδα 3	<a href="#">Επεξεργασία</a>
7	Ηλεκτρονική I (ID 1)	Σελίδα 4	<a href="#">Επεξεργασία</a>
4	Ηλεκτρονική II (ID 2)	Σελίδα 1	<a href="#">Επεξεργασία</a>
5	Ηλεκτρονική II (ID 2)	Σελίδα 2	<a href="#">Επεξεργασία</a>
6	Ηλεκτρονική II (ID 2)	Σελίδα 3	<a href="#">Επεξεργασία</a>

Εικόνα 4.8: Λίστα Όλων των Σελίδων Θεματικών Ενότητων

Στην εικόνα 4.9 φαίνεται η φόρμα που χρησιμοποιείται για τη δημιουργία νέας σελίδας εντός κάποιας θεματικής ενότητας. Ο διαχειριστής έχει τη δυνατότητα να εισαγάγει τίτλο, κείμενο περιεχομένου και να επισυνάψει αρχεία ή άλλους πόρους. Η προσθήκη μιας νέας σελίδας επιτρέπει την ανάπτυξη θεωρητικού υλικού, οδηγιών ή ασκήσεων. Η διαδικασία είναι σχεδιασμένη ώστε να μπορεί εύκολα να επαναληφθεί για κάθε νέα ανάγκη.

Δημιουργία Νέας Σελίδας

Θεματική Ενότητα  
Ηλεκτρονική I

Αριθμός Σελίδας

URL Εικόνας (προαιρετικό)

Κείμενο / Περιεχόμενο (HTML)

← → Paragraph **B** *I*

Εικόνα 4.9: Δημιουργία Νέας Σελίδας

Στην εικόνα 4.10 απεικονίζεται η πρώτη από τις διαθέσιμες φόρμες επεξεργασίας μιας σελίδας θεματικής ενότητας. Ο διαχειριστής μπορεί να τροποποιήσει το υπάρχον κείμενο, να προσθέσει ή να αφαιρέσει περιεχόμενο και να ανανεώσει την παρουσίαση της σελίδας. Η δυνατότητα αυτή εξασφαλίζει



## Επεξεργασία Σελίδας 3

Αριθμός Σελίδας

3

URL Εικόνας

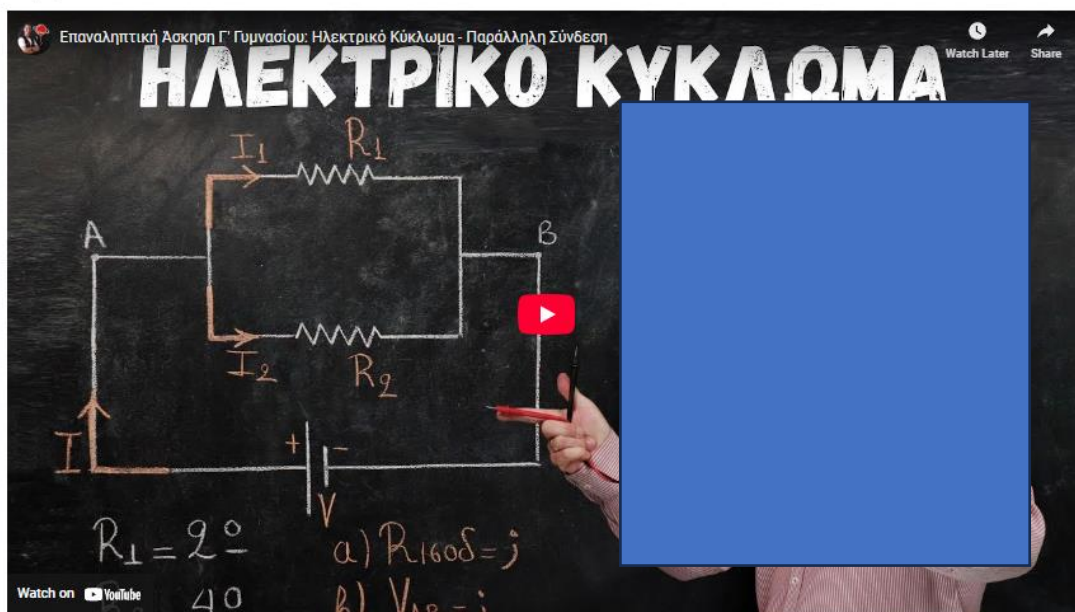
https://users.sch.gr/jabatzo/files/yilko/live%20ebooks/arxes\_elektrologias\_A\_2018\_final/arxes\_kef4-1.png

Κείμενο / Περιεχόμενο (HTML)

Heading 2 **B** *I*     

### Νόμος του Ohm

Ο νόμος του Ohm δίνεται από τον τύπο:  $V = I \times R$



Επαναληπτική Άσκηση Γ' Γυμνασίου: Ηλεκτρικό Κύκλωμα - Παράλληλη Σύνδεση

**ΗΛΕΚΤΡΙΚΟ ΚΥΚΛΩΜΑ**

Watch Later Share

Watch on YouTube







- V: Τάση (Volt)
- I: Ρεύμα (Ampere)
- R: Αντίσταση (Ohm)

[Μάθε περισσότερα στη Wikipedia](#)

Εικόνα 4.11: Επεξεργασία Σελίδας 2 – και με βίντεο και υπερσυνδέσμους

## Διαχείριση Quiz

+ Νέο Quiz

ID	Θεματική	Χρόνος	Ενεργό	Ενέργειες
1	Ηλεκτρονική I (ID 1)	5'	Ναι	  Ερωτήσεις  Export Αποτελεσμάτων
2	Ηλεκτρονική II (ID 2)	5'	Ναι	  Ερωτήσεις  Export Αποτελεσμάτων

Εικόνα 4.12: Διαχείριση Quiz

**Δημιουργία Νέου Quiz**

Θεματική Ενότητα  
Ηλεκτρονική Ι

Χρόνος (σε λεπτά)

Ενεργό Quiz

Εικόνα 4.13: Δημιουργία Νέου Quiz

Στην εικόνα 4.13 φαίνεται η φόρμα δημιουργίας νέου quiz, η οποία επιτρέπει στον διαχειριστή να ορίσει βασικές πληροφορίες όπως ο τίτλος του quiz και η θεματική ενότητα στην οποία θα ενταχθεί. Μέσα από αυτή τη σελίδα ξεκινά η διαδικασία κατασκευής αξιολογήσεων με σκοπό την εξέταση των φοιτητών πάνω στο αντίστοιχο υλικό. Ο διαχειριστής μπορεί επίσης να ρυθμίσει παραμέτρους που σχετίζονται με τη διαθεσιμότητα ή τη σειρά εμφάνισης των ερωτήσεων, προσαρμόζοντας το quiz στις ανάγκες κάθε ενότητας.

**Επεξεργασία Quiz**

Θεματική Ενότητα  
Ηλεκτρονική Ι

Χρόνος (σε λεπτά)  
5





Ενεργό Quiz

Εικόνα 4.14: Επεξεργασία Quiz

Στην εικόνα 4.14 απεικονίζεται η λειτουργία επεξεργασίας ενός υπάρχοντος quiz. Ο διαχειριστής έχει τη δυνατότητα να τροποποιήσει το όνομα, τη συσχέτιση με θεματική ενότητα, καθώς και να προσθέσει ή να αφαιρέσει ερωτήσεις.

## Ερωτήσεις για Quiz στο θέμα: Ηλεκτρονική Ι

+ Νέα Ερώτηση

ID	Σελίδα	Ερώτηση	Σωστή	Ενέργειες
1	Σελίδα 1	Ποιο είναι το αποτέλεσμα της φόρτισης σε RC κύκλωμ...	Εκθετική αύξηση	 
2	Σελίδα 2	Ποιος είναι ο τύπος για διαίρεση τάσης...	$V_{out} = V_{in} * R_2 / (R_1 + R_2)$	 

Εικόνα 4.15: Ερωτήσεις για Quiz στο θέμα: Ηλεκτρονική Ι

Στην εικόνα 4.15 εμφανίζονται οι ερωτήσεις που έχουν καταχωρηθεί για το quiz με τίτλο "Ηλεκτρονική Ι". Ο διαχειριστής μπορεί να δει τη λίστα των ερωτήσεων, να ελέγξει τις διατυπώσεις, τις απαντήσεις και τη σωστή επιλογή. Επίσης, από εδώ παρέχεται πρόσβαση σε λειτουργίες επεξεργασίας ή διαγραφής κάθε ερώτησης. Η σελίδα αυτή αποτελεί σημαντικό εργαλείο για την επισκόπηση του συνόλου των ερωτήσεων που θα κληθεί να απαντήσει ο φοιτητής κατά την αξιολόγησή του.

### Επεξεργασία Ερώτηση για Quiz:

Σελίδα (προέλευση ερώτησης)

Σελίδα 2

Ερώτηση

Ποιος είναι ο τύπος για διαίρεση τάσης;

Επιλογή 1

$V_{out} = V_{in} * R_1 / (R_1 + R_2)$

Επιλογή 2

$V_{out} = V_{in} * R_2 / (R_1 + R_2)$

Επιλογή 3

$V_{out} = V_{in} / R$

Επιλογή 4

$V_{out} = R_1 - R_2$

Σωστή Επιλογή (1-4)

2

Αποθήκευση

Εικόνα 4.16: Επεξεργασία Ερώτησης για Quiz - 1

Στην εικόνα 4.16 φαίνεται η φόρμα επεξεργασίας μιας συγκεκριμένης ερώτησης σε quiz. Ο διαχειριστής μπορεί να αλλάξει τη διατύπωση της ερώτησης, να τροποποιήσει τις διαθέσιμες απαντήσεις και να ορίσει εκ νέου ποια είναι η σωστή. Επιτρέπει τη συνεχή βελτίωση των ερωτήσεων και την προσαρμογή τους στις ανάγκες των φοιτητών.

### Επεξεργασία Ερώτηση για Quiz:

Σελίδα (προέλευση ερώτησης)

Σελίδα 2

Ερώτηση

Σε τι χρησιμεύει ο ενισχυτής κοινού εκπομπού;

Επιλογή 1

Αντιστροφική φάσης και ενίσχυση τάσης

Επιλογή 2

Μείωση θορύβου

Επιλογή 3

Σύζευξη AC-DC

Επιλογή 4

Ενίσχυση ρεύματος

Σωστή Επιλογή (1-4)

1

Αποθήκευση

Εικόνα 4.17: Επεξεργασία Ερώτησης για Quiz - 2

Στην εικόνα 4.17 παρουσιάζεται μια δεύτερη εκδοχή της φόρμας επεξεργασίας ερώτησης για quiz. Εκτός από την κλασική διατύπωση ερώτησης και επιλογών εδώ δίνεται η δυνατότητα προσθήκης επιπλέον χαρακτηριστικών, όπως είναι η αντιστοίχιση της ερώτησης με συγκεκριμένη σελίδα θεωρίας ή η παροχή υποδείξεων.

### Νέα Ερώτηση για Quiz: Ηλεκτρονική II

Σελίδα (προέλευση ερώτησης)

Σελίδα 1

Ερώτηση

Επιλογή 1

Επιλογή 2

Επιλογή 3

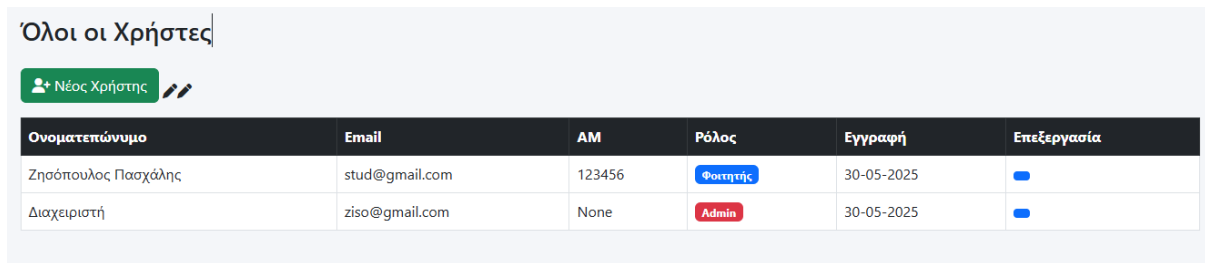
Επιλογή 4

Σωστή Επιλογή (1-4)

Αποθήκευση

Εικόνα 4.18: Νέα Ερώτηση για Quiz: Ηλεκτρονική II με επιλογή σελίδας

Στην εικόνα 4.18 φαίνεται η φόρμα για την προσθήκη νέας ερώτησης στο quiz της θεματικής ενότητας "Ηλεκτρονική ΙΙ". Εκτός από τα βασικά πεδία για τη διατύπωση της ερώτησης και των επιλογών απάντησης, υπάρχει δυνατότητα σύνδεσης της ερώτησης με συγκεκριμένη σελίδα της θεωρίας.



Όνοματεπώνυμο	Email	AM	Ρόλος	Εγγραφή	Επεξεργασία
Ζησόπουλος Πασχάλης	stud@gmail.com	123456	Φοιτητής	30-05-2025	<input type="checkbox"/>
Διαχειριστή	ziso@gmail.com	None	Admin	30-05-2025	<input type="checkbox"/>

Εικόνα 4.19: Όλοι οι Χρήστες

Στην εικόνα 4.19 παρουσιάζεται η σελίδα με τη λίστα όλων των χρηστών που είναι καταχωρημένοι στο σύστημα. Περιλαμβάνονται στοιχεία όπως το όνομα, το email και ο ρόλος κάθε χρήστη, ενώ ο διαχειριστής μπορεί να προβεί σε ενέργειες όπως τροποποίηση ή διαγραφή χρηστών.



**Νέος Χρήστης**

Όνοματεπώνυμο

Email

Αριθμός Μητρώου (AM)

Ρόλος  
Φοιτητής

Κωδικός

Εικόνα 4.20: Σελίδα για εγγραφή χρήστη

Στην εικόνα 4.20 απεικονίζεται η φόρμα εγγραφής νέου χρήστη στην πλατφόρμα. Ο ενδιαφερόμενος καλείται να συμπληρώσει βασικές πληροφορίες όπως το όνομα, το email και τον επιθυμητό κωδικό πρόσβασης. Η διαδικασία είναι απλή επιτρέποντας την εύκολη ένταξη νέων φοιτητών ή εκπαιδευτικών στο σύστημα.

## Καλωσήρθες Ζησόπουλος Πασχάλης!

Επίλεξε μια θεματική ενότητα για μελέτη και αξιολόγηση.



### Ηλεκτρονική I

Εισαγωγή στην αναλογική ηλεκτρονική...

Μετάβαση

Εικόνα 4.21: Αρχική σελίδα του χρήστη - βλέπει θεματικές μόνο του level που έχει πετύχει – το μενού έχει ιστορικό για τα quiz του

Στην εικόνα 4.21 εμφανίζεται η αρχική σελίδα του χρήστη μετά τη σύνδεση. Ο φοιτητής βλέπει μόνο τις θεματικές ενότητες που αντιστοιχούν στο επίπεδο προόδου του, το οποίο έχει επιτύχει βάσει των quiz που έχει ολοκληρώσει. Στο μενού της πλατφόρμας περιλαμβάνεται και σύνδεσμος για την προβολή του ιστορικού των quiz, δίνοντας στον φοιτητή τη δυνατότητα να παρακολουθεί την πορεία του.

Σελίδα 1

Σελίδα 2

Σελίδα 3

Σελίδα 4

Ξεκίνα Quiz Αξιολόγησης

### Κύκλωμα συνεχούς ρεύματος RC - Φόρτιση πυκνωτή (3/4)

Κύκλωμα RC σε σειρά

Διακόπτης Δ

Τάση στα άκρα του πυκνωτή:

$$V_c(t) = E \times \left(1 - e^{-\frac{t}{RC}}\right)$$

**Σελίδα 1**  
RC κύκλωμα φόρτισης πυκνωτή.

Εικόνα 4.22: Ιστοσελίδα που βλέπει το υλικό θεωρίας για να διαβάσει και να ετοιμαστεί για το Quiz θεματικής – όταν είναι έτοιμος πατάει το Ξεκίνα Quiz Αξιολόγησης

Στην εικόνα 4.22 φαίνεται η σελίδα που περιλαμβάνει το θεωρητικό υλικό μιας θεματικής ενότητας, το οποίο ο φοιτητής μπορεί να μελετήσει πριν ξεκινήσει το quiz. Το περιεχόμενο είναι οργανωμένο σε

κείμενο, εικόνες και πιθανώς πολυμέσα, ώστε να υποστηρίζει την καλύτερη κατανόηση των εννοιών. Στο τέλος της σελίδας υπάρχει κουμπί που επιτρέπει την έναρξη της αξιολόγησης.

The screenshot shows a light blue header with the title "Αξιολόγηση Θεματικής Ενότητας: Ηλεκτρονική Ι" in bold black text. Below the title, there is a paragraph of text: "Η αξιολόγηση περιλαμβάνει ερωτήσεις σχετικές με τις σελίδες του μαθήματος. Έχεις συνολικό χρόνο 5 λεπτά. Μπορείς να περιηγηθείς στις ερωτήσεις, αλλά δεν μπορείς να αποθηκεύσεις ενδιάμεσα. Μόλις πατήσεις "Έναρξη", ξεκινά η χρονομέτρηση." At the bottom center, there is a green button with a white play icon and the text "Έναρξη Quiz".

Εικόνα 4.23: Έναρξη για το «Αξιολόγηση Θεματικής Ενότητας: Ηλεκτρονική Ι»

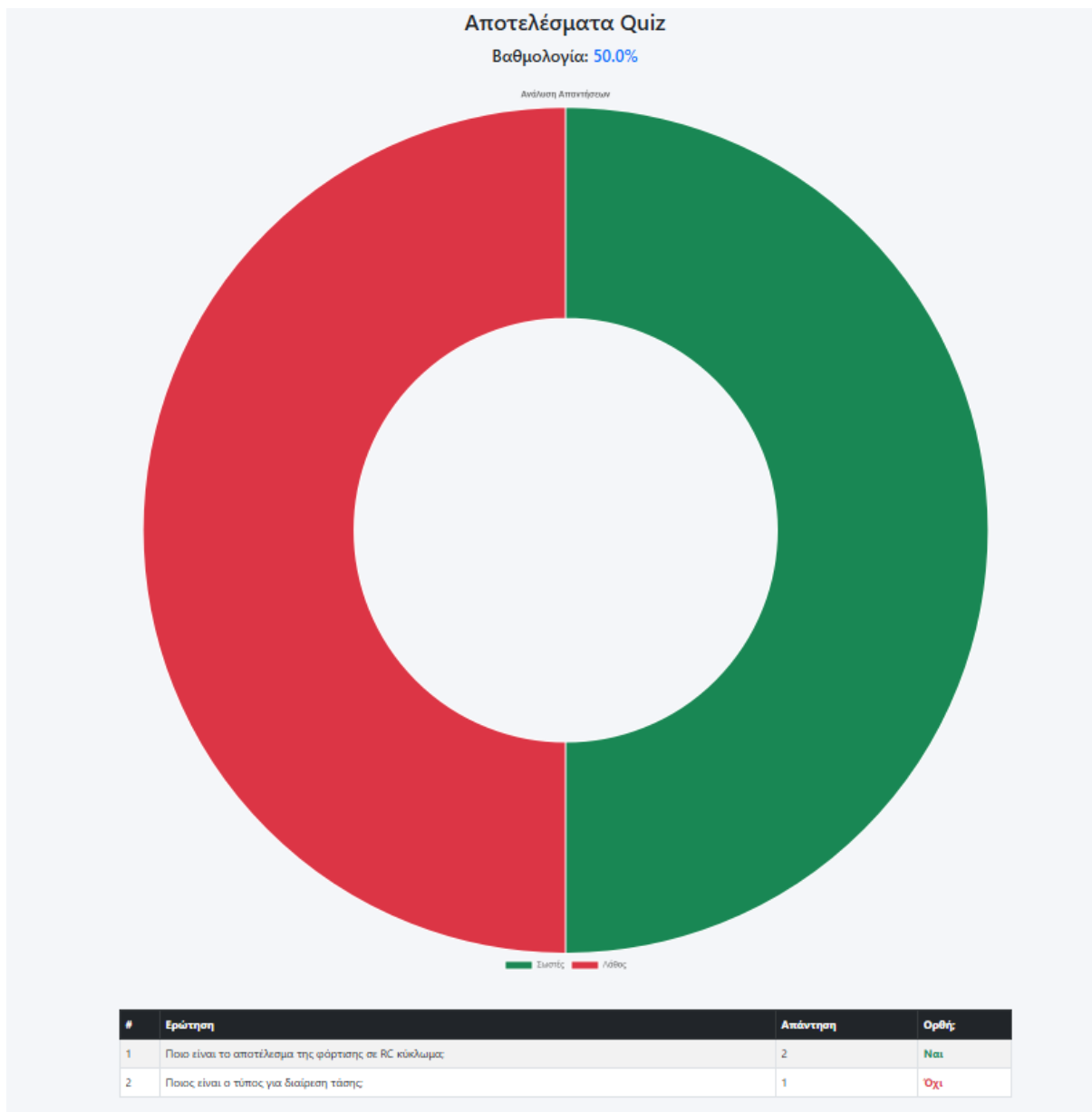
Στην εικόνα 4.23 παρουσιάζεται η αρχική σελίδα του quiz για τη θεματική ενότητα «Ηλεκτρονική Ι». Ο χρήστης βλέπει τον τίτλο του quiz και βασικές πληροφορίες για την αξιολόγηση που πρόκειται να ξεκινήσει. Η σελίδα λειτουργεί ως εισαγωγική, προετοιμάζοντας τον φοιτητή για τη διαδικασία που ακολουθεί. Από εδώ μπορεί να ξεκινήσει το quiz όταν νιώσει έτοιμος, γνωρίζοντας ότι οι απαντήσεις του θα καταγραφούν και θα αξιολογηθούν.

The screenshot shows the quiz interface. At the top left, it says "Quiz: Ηλεκτρονική Ι" and at the top right, "Χρόνος: 4:56". The main area is titled "Ερώτηση 1" and contains the question: "Ποιο είναι το αποτέλεσμα της φόρτισης σε RC κύκλωμα;". Below the question are four radio button options: "Σταθερή τάση", "Εκθετική αύξηση", "Γραμμική αύξηση", and "Μηδενισμός". At the bottom right, there is a blue button labeled "Υποβολή Quiz".

The screenshot shows the quiz interface with the second question selected. At the top left, it says "Quiz: Ηλεκτρονική Ι" and at the top right, "Χρόνος: 4:37". The main area is titled "Ερώτηση 2" and contains the question: "Ποιος είναι ο τύπος για διαίρεση τάσης;". Below the question are four radio button options: "Vout = Vin \* R1 / (R1+R2)", "Vout = Vin \* R2 / (R1+R2)", "Vout = Vin / R", and "Vout = R1 - R2".

Εικόνα 4.24: Ερωτήσεις που πρέπει να απαντήσει ο χρήστης με όποια σειρά θέλει και στο τέλος να πατήσει Υποβολή Quiz

Στην εικόνα 4.24 φαίνεται η κεντρική σελίδα του quiz, όπου παρουσιάζονται όλες οι ερωτήσεις προς απάντηση. Ο φοιτητής έχει τη δυνατότητα να απαντήσει τις ερωτήσεις με όποια σειρά επιθυμεί, κάτι που του δίνει μεγαλύτερη ευελιξία κατά τη διάρκεια της αξιολόγησης. Κάθε ερώτηση συνοδεύεται από τις επιλογές απάντησης, και στο τέλος της σελίδας υπάρχει το κουμπί "Υποβολή Quiz" που ολοκληρώνει τη διαδικασία.



Εικόνα 4.25: Αποτελέσματα Quiz

Στην εικόνα 4.25 φαίνονται τα αναλυτικά αποτελέσματα ενός quiz στο οποίο έχει συμμετάσχει ο φοιτητής. Στο επάνω μέρος παρουσιάζεται ένας κυκλικός γραφικός δείκτης που αποτυπώνει οπτικά το ποσοστό επιτυχίας, με πράσινο χρώμα για τις σωστές και κόκκινο για τις λανθασμένες απαντήσεις. Στην περίπτωση της εικόνας, ο φοιτητής έχει επιτύχει βαθμολογία 50%, απαντώντας σωστά στη μία

από τις δύο ερωτήσεις. Στο κάτω μέρος της σελίδας εμφανίζεται πίνακας με τις ερωτήσεις, τις απαντήσεις του χρήστη και τις σωστές απαντήσεις.

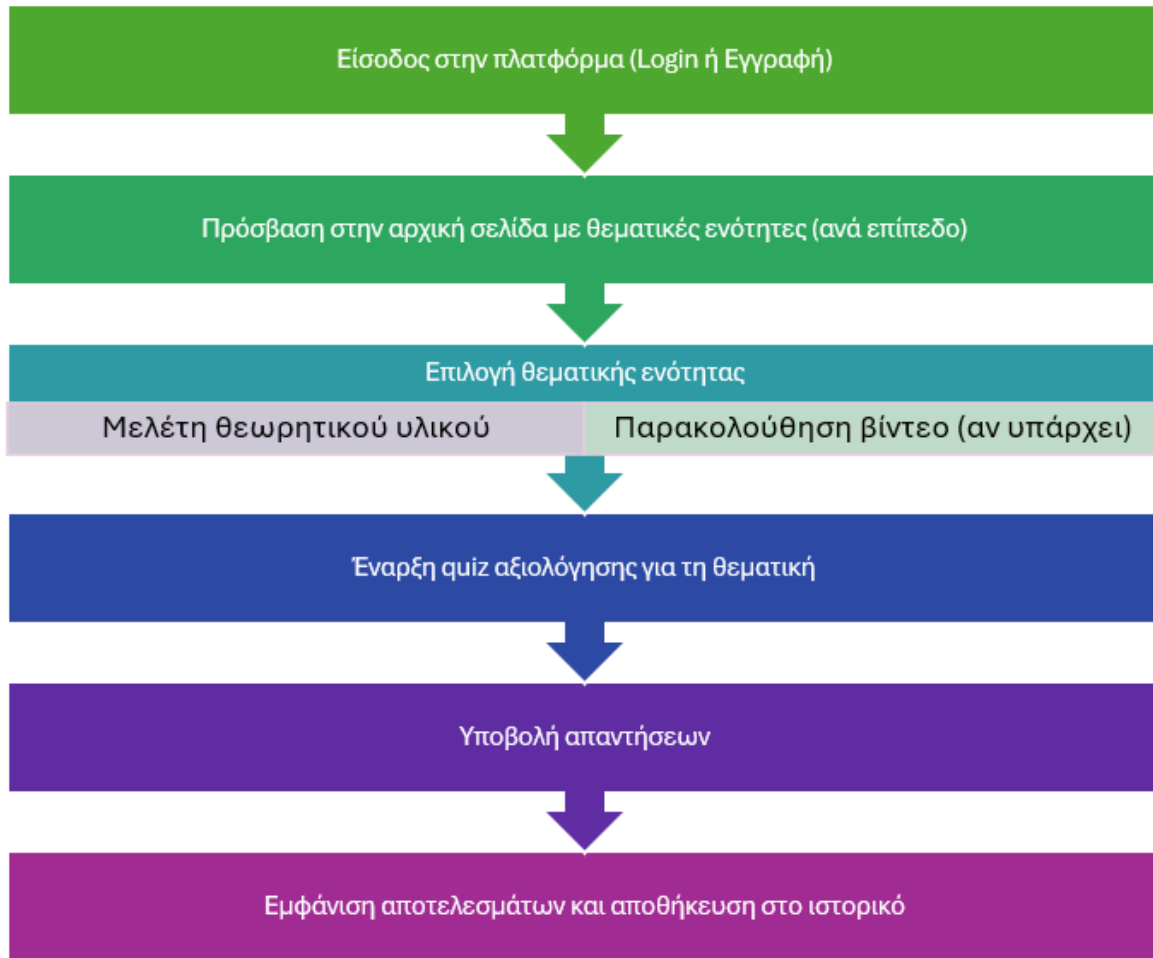
Τα Quiz που έχεις συμμετάσχει		
Θεματική Ενότητα	Βαθμολογία	Ημερομηνία
Ηλεκτρονική I	50.0%	31-05-2025 13:30

Εικόνα 4.26: Ιστορικό αποτελεσμάτων Quiz που έχει συμμετάσχει ο χρήστης

Στην εικόνα 4.26 απεικονίζεται η σελίδα με το ιστορικό των quiz που έχει ολοκληρώσει ο φοιτητής. Κάθε εγγραφή περιλαμβάνει τη θεματική ενότητα στην οποία ανήκε το quiz, τη συνολική βαθμολογία και την ακριβή ημερομηνία και ώρα συμμετοχής. Αυτή η προβολή επιτρέπει στον χρήστη να παρακολουθεί την πρόοδό του με την πάροδο του χρόνου και να αξιολογεί την πορεία του σε σχέση με τα διάφορα θέματα.

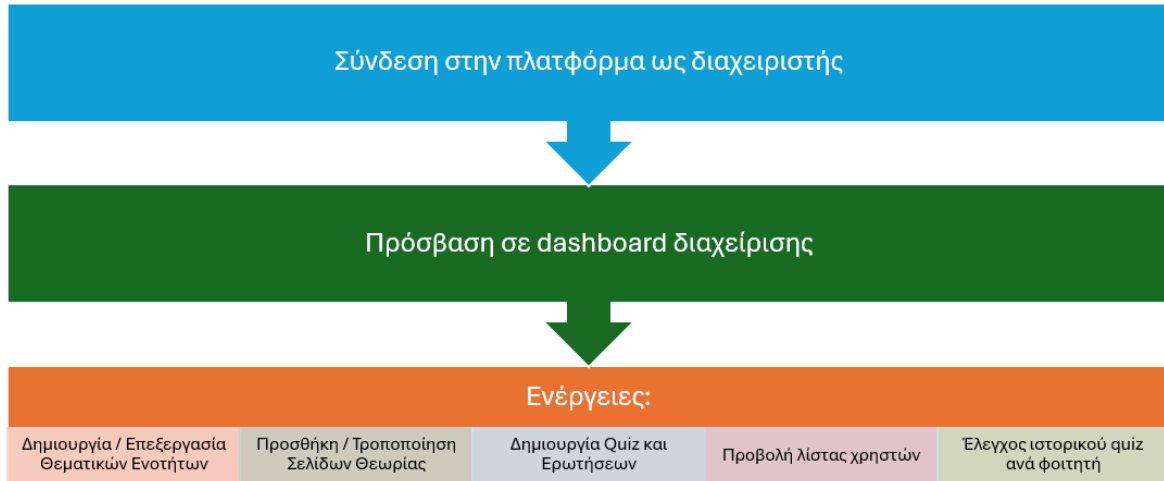
## 4.2 Περιγραφή μέσω διαγραμμάτων

Το πρώτο διάγραμμα απεικονίζει τη βασική ροή του απλού χρήστη (φοιτητή) μέσα στην πλατφόρμα. Αρχικά ο φοιτητής εισέρχεται στο σύστημα μέσω της σελίδας σύνδεσης ή αν δεν διαθέτει λογαριασμό, πραγματοποιεί εγγραφή. Μετά τη σύνδεση, μεταφέρεται στην αρχική του σελίδα, όπου εμφανίζονται οι θεματικές ενότητες που είναι διαθέσιμες ανάλογα με την πρόοδό του. Επιλέγοντας μια θεματική ενότητα, έχει πρόσβαση σε σχετικό θεωρητικό υλικό, το οποίο μπορεί να περιλαμβάνει κείμενα, εικόνες, συνδέσμους ή βίντεο. Όταν ολοκληρώσει τη μελέτη, έχει τη δυνατότητα να ξεκινήσει το αντίστοιχο quiz αξιολόγησης. Απαντά στις ερωτήσεις του quiz, υποβάλλει τις απαντήσεις του και στη συνέχεια βλέπει τα αποτελέσματα και την ανατροφοδότηση. Η απόδοσή του καταγράφεται στο προσωπικό του ιστορικό, δίνοντάς του τη δυνατότητα να παρακολουθεί την πορεία του μέσα στο σύστημα. Αυτή η ροή υποστηρίζει μια αυτοκατευθυνόμενη και προοδευτική μαθησιακή εμπειρία.



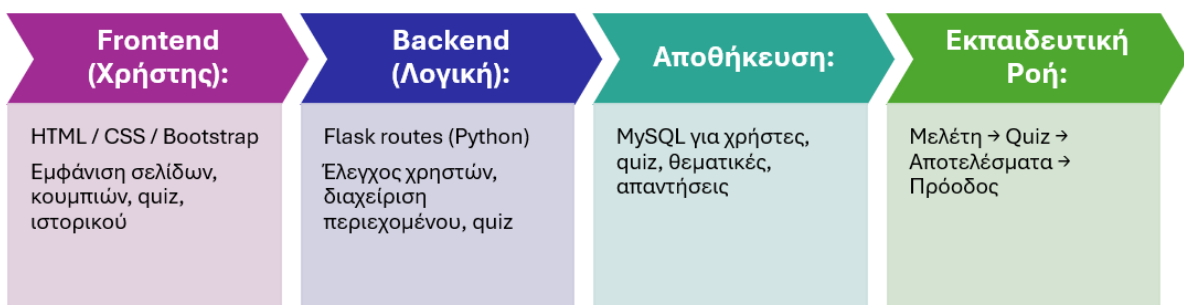
Εικόνα 4.27: Δομική διάταξη συστήματος

Το δεύτερο διάγραμμα περιγράφει τη ροή και τις δυνατότητες που διαθέτει ο διαχειριστής ή εκπαιδευτικός μέσα στην πλατφόρμα. Ο διαχειριστής συνδέεται μέσω της ειδικής φόρμας σύνδεσης και αποκτά πρόσβαση σε ένα κεντρικό μενού διαχείρισης. Από εκεί μπορεί να δημιουργήσει νέες θεματικές ενότητες, να προσθέσει ή να επεξεργαστεί σελίδες με θεωρητικό υλικό και να οργανώσει το εκπαιδευτικό περιεχόμενο με βάση τη διδακτέα ύλη. Επιπλέον έχει τη δυνατότητα να δημιουργεί quiz για κάθε θεματική, να προσθέτει ή να επεξεργάζεται ερωτήσεις και να ορίζει τις σωστές απαντήσεις. Μέσω της πλατφόρμας μπορεί να διαχειρίζεται όλους τους χρήστες (φοιτητές), να παρακολουθεί τη δραστηριότητά τους, να βλέπει τα quiz που έχουν υποβληθεί και να εξάγει συμπεράσματα για την πρόοδό τους. Οι λειτουργίες αυτές εξασφαλίζουν πλήρη έλεγχο του περιεχομένου και της εκπαιδευτικής διαδικασίας, καθιστώντας τον διαχειριστή υπεύθυνο για την ομαλή λειτουργία του συστήματος.



Εικόνα 4.28: Λειτουργίες διαχειριστή

Το τρίτο διάγραμμα παρουσιάζει τη συνολική δομική αρχιτεκτονική του συστήματος, διαχωρισμένη σε βασικά υποσυστήματα. Στο επίπεδο του χρήστη (frontend), η πλατφόρμα βασίζεται σε τεχνολογίες όπως HTML, CSS και Bootstrap για να προσφέρει μια αισθητικά ευχάριστη και λειτουργική εμπειρία πλοήγησης. Ο χρήστης αλληλεπιδρά με τις σελίδες, βλέπει περιεχόμενο, απαντά σε quiz και λαμβάνει ανατροφοδότηση. Στο backend επίπεδο, η γλώσσα Python σε συνδυασμό με το framework Flask διαχειρίζεται όλη τη λογική της εφαρμογής, περιλαμβάνοντας τη διαχείριση χρηστών, τις λειτουργίες quiz, την αυθεντικοποίηση, τη δυναμική δημιουργία σελίδων και τη διασύνδεση με τη βάση δεδομένων. Η MySQL χρησιμοποιείται ως αποθηκευτικός μηχανισμός, όπου καταγράφονται οι πληροφορίες των χρηστών, το υλικό, οι θεματικές, τα quiz και τα αποτελέσματα.



Εικόνα 4.29: Δομική διάταξη συστήματος

### 4.3 Η βάση που χρησιμοποιήθηκε

Η βάση δεδομένων είναι υπεύθυνη για την αποθήκευση, οργάνωση και ανάκτηση όλων των κρίσιμων πληροφοριών του συστήματος. Χρησιμοποιήθηκε η MySQL λόγω της ευκολίας διαχείρισης, της απόδοσης και της ευρείας υποστήριξης σε περιβάλλοντα ανάπτυξης με Flask.

Περιγραφή των βασικών πινάκων

Η βάση περιλαμβάνει πίνακες που σχετίζονται με:

- τους χρήστες,
- τις θεματικές ενότητες,
- το εκπαιδευτικό υλικό (σελίδες),
- τα quiz,
- τις ερωτήσεις και τις απαντήσεις,
- τις προσπάθειες και τα αποτελέσματα.

Παρακάτω παρουσιάζεται αναλυτικά ο ρόλος κάθε πίνακα.

Πίνακας users

Ο πίνακας users περιλαμβάνει όλα τα δεδομένα που αφορούν τους χρήστες του συστήματος. Υποστηρίζονται δύο ρόλοι: student και admin. Κάθε χρήστης έχει μοναδικό id, διεύθυνση email (χρησιμοποιείται ως username), κρυπτογραφημένο κωδικό πρόσβασης και πεδίο fullname για το ονοματεπώνυμο. Το am μπορεί να περιέχει τον αριθμό μητρώου φοιτητή, ενώ το created\_at καταγράφει την ημερομηνία εγγραφής.

Πίνακας topics

Ο πίνακας topics αντιπροσωπεύει τις θεματικές ενότητες της πλατφόρμας. Κάθε εγγραφή περιλαμβάνει τίτλο (title), περιγραφή (description), προαιρετική εικόνα (image\_url), καθώς και το επίπεδο δυσκολίας (level). Επιπλέον, υπάρχει το πεδίο unlocked\_by\_score που καθορίζει αν ένας φοιτητής έχει πρόσβαση στη θεματική ανάλογα με τη βαθμολογία που πέτυχε σε προηγούμενα quiz. Κάθε θεματική δημιουργείται από κάποιον διαχειριστή, του οποίου καταγράφεται το created\_by.

### Πίνακας topic\_pages

Κάθε θεματική ενότητα αποτελείται από μία ή περισσότερες σελίδες (topic\_pages), οι οποίες περιέχουν εκπαιδευτικό υλικό. Ο πίνακας συνδέεται με τον topic\_id, περιλαμβάνει αριθμό σελίδας (page\_number), εικόνα (image\_url) και επεξηγηματικό κείμενο (explanation). Ο διαχωρισμός των θεματικών σε σελίδες επιτρέπει την καλύτερη οργάνωση της πληροφορίας, διευκολύνει τη μελέτη και υποστηρίζει την σταδιακή παρουσίαση της ύλης.

### Πίνακας quizzes

Ο πίνακας quizzes περιέχει τις αξιολογήσεις που σχετίζονται με κάθε θεματική. Κάθε quiz συνδέεται με έναν topic\_id, έχει χρονικό περιορισμό (total\_time\_minutes) και μπορεί να είναι ενεργό ή όχι μέσω του πεδίου active. Η ευελιξία αυτή επιτρέπει στον διαχειριστή να ενεργοποιεί ή να απενεργοποιεί quiz ανάλογα με την περίοδο διδασκαλίας ή την πρόοδο των φοιτητών.

### Πίνακας questions

Ο πίνακας questions περιλαμβάνει τις ερωτήσεις κάθε quiz. Κάθε ερώτηση συνδέεται με συγκεκριμένο quiz\_id, και έχει επίσης το page\_id που τη συσχετίζει με τη σελίδα από την οποία προέρχεται η γνώση. Υποστηρίζονται έως και τέσσερις επιλογές απάντησης (option1 έως option4), ενώ το correct\_option δηλώνει ποια είναι η σωστή απάντηση. Η δομή αυτή δίνει τη δυνατότητα σύνδεσης κάθε ερώτησης με συγκεκριμένο εκπαιδευτικό περιεχόμενο, ενισχύοντας τη στοχευμένη αξιολόγηση.

### Πίνακας quiz\_attempts

Ο πίνακας quiz\_attempts καταγράφει κάθε προσπάθεια συμμετοχής φοιτητή σε quiz. Για κάθε απόπειρα αποθηκεύεται ο user\_id, το quiz\_id, η τελική score και η ημερομηνία υποβολής (submitted\_at). Μέσω αυτού του πίνακα διατηρείται το ιστορικό των συμμετοχών του φοιτητή και υπολογίζεται η πρόοδος του στο σύστημα.

### Πίνακας quiz\_answers

Ο πίνακας quiz\_answers αποθηκεύει κάθε απάντηση που δόθηκε από έναν φοιτητή κατά τη διάρκεια ενός quiz. Κάθε εγγραφή περιλαμβάνει attempt\_id για την αντιστοίχιση με την προσπάθεια, question\_id για την ερώτηση, την επιλεγμένη απάντηση (selected\_option) και αν αυτή ήταν σωστή (correct). Η καταγραφή αυτών των δεδομένων επιτρέπει την αναλυτική παρουσίαση των αποτελεσμάτων, τη δημιουργία γραφημάτων απόδοσης και την αξιολόγηση της κατανόησης του φοιτητή σε κάθε επιμέρους θέμα.

## Σχέσεις και αλληλεπίδραση πινάκων

Αν και δεν χρησιμοποιούνται ρητά FOREIGN KEY constraints, η σχεδίαση της βάσης στηρίζεται σε λογικές συσχετίσεις μεταξύ πινάκων:

- Οι users συνδέονται με quiz\_attempts
- Οι quiz\_attempts συνδέονται με quiz\_answers
- Οι quizzes συνδέονται με topics και questions
- Οι questions συνδέονται με topic\_pages για να δώσουν νόημα στην αξιολόγηση

Η αλληλεπίδραση αυτή εξασφαλίζει ότι το σύστημα μπορεί να διαχειρίζεται πολύπλοκες ενέργειες, όπως η παρουσίαση προσαρμοσμένου περιεχομένου, η ανάλυση της απόδοσης, η ενεργοποίηση quiz και η παρακολούθηση της συνολικής εξέλιξης του κάθε φοιτητή.

## 4.4 Προσέγγιση για την Ασφάλεια

Η ασφάλεια αποτελεί βασική προτεραιότητα σε κάθε διαδικτυακή πλατφόρμα εκπαίδευσης, ιδιαίτερα όταν διαχειρίζεται προσωπικά δεδομένα φοιτητών και εκπαιδευτικών, καθώς και αποτελέσματα αξιολόγησης. Το παρόν σύστημα έχει υλοποιηθεί με γνώμονα την προστασία των πληροφοριών και την αποτροπή κακόβουλων ενεργειών τόσο κατά την είσοδο όσο και κατά την περιήγηση και την αποθήκευση δεδομένων.

Η διαδικασία ταυτοποίησης των χρηστών στηρίζεται σε σύστημα σύνδεσης με έλεγχο email και κωδικού πρόσβασης. Οι κωδικοί αποθηκεύονται στη βάση δεδομένων σε κρυπτογραφημένη μορφή, με χρήση ασφαλούς αλγορίθμου hashing, ώστε να μην είναι αναγνώσιμοι ακόμη και σε περίπτωση παραβίασης της βάσης. Η αυθεντικοποίηση των χρηστών πραγματοποιείται σε κάθε ευαίσθητο σημείο της εφαρμογής, αποτρέποντας την πρόσβαση σε περιεχόμενο που δεν τους ανήκει.

Η πρόσβαση στις λειτουργίες διαχείρισης επιτρέπεται αποκλειστικά σε χρήστες με ρόλο διαχειριστή. Ο έλεγχος των ρόλων γίνεται σε κάθε route του συστήματος, ώστε να αποκλείεται οποιαδήποτε μη εξουσιοδοτημένη ενέργεια, είτε μέσω διεπαφής είτε μέσω απευθείας κλήσης διευθύνσεων. Επιπλέον χρησιμοποιούνται session cookies για τη διατήρηση της σύνδεσης του χρήστη, τα οποία προστατεύονται από μηχανισμούς expiration και έλεγχο εγκυρότητας.

Κατά την αποστολή δεδομένων από τους χρήστες (όπως quiz απαντήσεις ή προσωπικά στοιχεία), εφαρμόζονται έλεγχοι εγκυρότητας τόσο από την πλευρά του client όσο και του server. Με αυτόν τον τρόπο αποφεύγονται τυχόν κακόβουλες εισαγωγές, όπως SQL injection ή XSS. Όλα τα δεδομένα που

εμφανίζονται στις σελίδες του χρήστη περνούν από φιλτράρισμα και escaping, ώστε να διασφαλίζεται ότι δεν εκτελείται επικίνδυνος κώδικας στον browser.

## Κεφάλαιο 5ο: Τα συμπεράσματα της εργασίας και κάποιες προτάσεις για βελτίωση

Η παρούσα εργασία έδειξε τη δυνατότητα σχεδίασης και υλοποίησης μιας σύγχρονης διαδικτυακής πλατφόρμας μάθησης για την Ηλεκτρονική με σκοπό την ενίσχυση της εκπαιδευτικής διαδικασίας. Η υλοποίηση πραγματοποιήθηκε καλύπτοντας ένα ευρύ φάσμα λειτουργιών όπως η εγγραφή και η διαχείριση χρηστών, η δημιουργία και οργάνωση θεματικών ενοτήτων, η παροχή εκπαιδευτικού υλικού με δομημένο τρόπο και η ανάπτυξη συστήματος αξιολόγησης με τη μορφή quiz. Η εμπειρία χρήσης που προσφέρεται είναι ικανοποιητική και λειτουργική τόσο για τον φοιτητή όσο και για τον διαχειριστή ενώ η διάκριση ρόλων και οι ελεγχόμενες προσβάσεις συμβάλλουν στην ομαλή και ασφαλή λειτουργία της εφαρμογής.

Ένα από τα βασικά θετικά της πλατφόρμας είναι η ευκολία στη χρήση καθώς οι διεπαφές είναι καθαρές και απλές γεγονός που διευκολύνει ιδιαίτερα τους φοιτητές που δεν είναι εξοικειωμένοι με πολύπλοκα ψηφιακά περιβάλλοντα. Επιπλέον, το σύστημα προσφέρει έναν ξεκάθαρο διαχωρισμό θεωρίας και αξιολόγησης, δίνοντας τη δυνατότητα στο χρήστη να προετοιμαστεί πλήρως πριν εξεταστεί. Ιδιαίτερα χρήσιμη αποδείχθηκε η δυνατότητα συσχέτισης κάθε ερώτησης με σελίδες θεωρίας, καθώς αυτό ενισχύει την παιδαγωγική συνοχή και διευκολύνει την αναστοχαστική μάθηση. Ένα ακόμη σημαντικό πλεονέκτημα της εφαρμογής είναι η δυνατότητα αυτόματης καταγραφής της προόδου του χρήστη, προσφέροντάς του σαφή εικόνα για τις επιδόσεις του μέσω ιστορικού και στατιστικών αποτελεσμάτων. Η επιλογή τεχνολογιών όπως Flask και MySQL αποδείχθηκε ιδιαίτερα αποτελεσματική για την υλοποίηση ενός ελαφριού αλλά ευέλικτου συστήματος, το οποίο μπορεί να λειτουργήσει ακόμη και σε απλά περιβάλλοντα χωρίς μεγάλο υπολογιστικό κόστος.

Παρά τα θετικά στοιχεία, προέκυψαν ορισμένα σημεία που θα μπορούσαν να βελτιωθούν σε μελλοντικές εκδόσεις της πλατφόρμας. Πρώτον, θα ήταν χρήσιμο να ενσωματωθεί δυνατότητα υποστήριξης πολυγλωσσικού περιβάλλοντος, ώστε η πλατφόρμα να είναι πιο προσβάσιμη σε χρήστες διαφορετικής γλωσσικής προέλευσης. Δεύτερον, η διεπαφή χρήστη, αν και λειτουργική, μπορεί να βελτιωθεί αισθητικά με τη χρήση πιο σύγχρονων σχεδιαστικών προτύπων και δυναμικών στοιχείων, βελτιώνοντας έτσι την εμπειρία πλοήγησης. Επιπλέον, η διαδικασία αξιολόγησης μέσω quiz, αν και πλήρως λειτουργική, θα μπορούσε να εμπλουτιστεί με περισσότερους τύπους ερωτήσεων όπως αντιστοίχιση, συμπλήρωση κενού ή διαδραστικές ασκήσεις, ώστε να υποστηρίζονται διαφορετικά μαθησιακά στυλ και να ενισχυθεί η παρακίνηση των φοιτητών.

Η εργασία έδειξε πως με σωστή σχεδίαση και αξιοποίηση κατάλληλων τεχνολογιών είναι δυνατόν να αναπτυχθεί μια πλήρως λειτουργική και αποτελεσματική πλατφόρμα μάθησης η οποία μπορεί να χρησιμοποιηθεί τόσο σε ακαδημαϊκό πλαίσιο όσο και σε ανεξάρτητα εκπαιδευτικά περιβάλλοντα. Η περαιτέρω εξέλιξη της πλατφόρμας μπορεί να βασιστεί στην ανατροφοδότηση των χρηστών, στην τεχνολογική αναβάθμιση με στόχο τη συνεχή βελτίωση της ποιότητας και της χρησιμότητάς της. Επίσης, να τονισθεί ότι για την σύνταξη της εργασία χρησιμοποιήθηκαν τεχνολογίες τεχνητής νοημοσύνης για τη διόρθωση και συμβουλές στο συντακτικό του κειμένου.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] <https://www.learningeverest.com/moodle-lms-features-and-benefits>
- [2] <https://en.wikipedia.org/wiki/Moodle>
- [3] [https://docs.chamilo.org/teacher-guide/getting-to-know-chamilo/what\\_is\\_chamilo](https://docs.chamilo.org/teacher-guide/getting-to-know-chamilo/what_is_chamilo)
- [4] <https://openedx.org/blog/working-with-in-video-quizzes-in-the-open-edx-platform/>
- [5] <https://support.google.com/edu/classroom/answer/9095575?hl=en>
- [6] <https://www.w3schools.com/python/>
- [7] [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- [8] <https://www.learnpython.org/>
- [9] <https://www.online-python.com/>
- [10] <https://serokell.io/blog/python-pros-and-cons>
- [11] <https://serokell.io/blog/python-pros-and-cons>
- [12] [https://arxiv.org/abs/2005.00910?utm\\_source=chatgpt.com](https://arxiv.org/abs/2005.00910?utm_source=chatgpt.com)
- [13] <https://www.w3schools.com/MySQL/default.asp>
- [14] [https://www.w3schools.com/bootstrap5/bootstrap\\_get\\_started.php](https://www.w3schools.com/bootstrap5/bootstrap_get_started.php)

# ΚΩΔΙΚΑΣ

Το κεντρικό αρχείο app.py

```
from flask import Flask, render_template, request, redirect, session, url_for, flash
from db_config import get_db_connection
import hashlib
import csv
from io import StringIO
from flask import Response

app = Flask(__name__)
app.secret_key = 'supersecretkey'

@app.route('/')
def home():
    return redirect('/login')

@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password'] #hashlib.sha256(request.form['password'].encode()).hexdigest()

        conn = get_db_connection()
        cursor = conn.cursor(dictionary=True)
        cursor.execute("SELECT * FROM users WHERE email = %s AND password = %s", (email, password))
        user = cursor.fetchone()
        cursor.close()
        conn.close()

        if user:
            session['user_id'] = user['id']
            session['fullname'] = user['fullname']
            session['role'] = user['role']
            return redirect('/dashboard')
        else:
            flash("Λανθασμένα στοιχεία σύνδεσης", "danger")

    return render_template('welcome.html')

@app.route('/logout')
def logout():
```

```

session.clear()
return redirect('/login')

@app.route('/topic/<int:topic_id>')
@app.route('/topic/<int:topic_id>/page/<int:page_number>')
def view_topic(topic_id, page_number=1):
    if 'user_id' not in session:
        return redirect('/login')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    # Λήψη στοιχείων topic
    cursor.execute("SELECT * FROM topics WHERE id = %s", (topic_id,))
    topic = cursor.fetchone()

    if not topic:
        cursor.close()
        conn.close()
        flash("Το θέμα δεν βρέθηκε", "danger")
        return redirect('/dashboard')

    # Όλες οι σελίδες του topic για sidebar
    cursor.execute("SELECT * FROM topic_pages WHERE topic_id = %s ORDER BY page_number", (topic_id,))
    pages = cursor.fetchall()

    # Περιεχόμενο της τρέχουσας σελίδας
    cursor.execute("SELECT * FROM topic_pages WHERE topic_id = %s AND page_number = %s", (topic_id,
    page_number))
    page_content = cursor.fetchone()

    cursor.close()
    conn.close()

    return render_template("topic_view.html",
        topic=topic,
        pages=pages,
        page_content=page_content,
        current_page=page_number)

@app.route('/quiz/start/<int:topic_id>')
def quiz_start(topic_id):

```

```

if 'user_id' not in session:
    return redirect('/login')

conn = get_db_connection()
cursor = conn.cursor(dictionary=True)

cursor.execute("SELECT * FROM topics WHERE id = %s", (topic_id,))
topic = cursor.fetchone()

cursor.execute("SELECT * FROM quizzes WHERE topic_id = %s AND active = 1", (topic_id,))
quiz = cursor.fetchone()

cursor.close()
conn.close()

if not quiz:
    flash("Δεν υπάρχει ενεργό quiz για αυτό το topic.", "warning")
    return redirect('/dashboard')

return render_template("quiz_start.html", topic=topic, quiz=quiz)

@app.route('/quiz/run/<int:quiz_id>', methods=['GET'])
def quiz_run(quiz_id):
    if 'user_id' not in session:
        return redirect('/login')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    cursor.execute("SELECT * FROM quizzes WHERE id = %s", (quiz_id,))
    quiz = cursor.fetchone()

    if not quiz:
        flash("Το quiz δεν βρέθηκε.", "danger")
        return redirect('/dashboard')

    cursor.execute("SELECT * FROM topics WHERE id = %s", (quiz['topic_id'],))
    topic = cursor.fetchone()

    cursor.execute("SELECT * FROM questions WHERE quiz_id = %s", (quiz_id,))
    questions = cursor.fetchall()

    cursor.close()

```

```

conn.close()

return render_template("quiz_run.html", quiz=quiz, topic=topic, questions=questions)

@app.route('/quiz/submit/<int:quiz_id>', methods=['POST'])
def quiz_submit(quiz_id):
    if 'user_id' not in session:
        return redirect('/login')

    user_id = session['user_id']
    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    # Φέρνουμε όλες τις ερωτήσεις του quiz
    cursor.execute("SELECT * FROM questions WHERE quiz_id = %s", (quiz_id,))
    questions = cursor.fetchall()

    correct_count = 0
    total = len(questions)

    # Δημιουργία νέου attempt
    cursor.execute("INSERT INTO quiz_attempts (user_id, quiz_id, score) VALUES (%s, %s, 0)", (user_id, quiz_id))
    attempt_id = cursor.lastrowid

    for q in questions:
        qid = q['id']
        selected = int(request.form.get(f'q{qid}', 0)) # Αν δεν απαντήθηκε
        is_correct = int(selected == q['correct_option'])

        if is_correct:
            correct_count += 1

        cursor.execute(
            "INSERT INTO quiz_answers (attempt_id, question_id, selected_option, correct) VALUES (%s, %s, %s, %s)",
            (attempt_id, qid, selected, is_correct)
        )

    # Υπολογισμός τελικής βαθμολογίας
    score = round((correct_count / total) * 100, 2)
    cursor.execute("UPDATE quiz_attempts SET score = %s WHERE id = %s", (score, attempt_id))

    cursor.close()
    conn.commit()

```

```

conn.close()

return redirect(url_for('quiz_result', attempt_id=attempt_id))

@app.route('/quiz/result/<int:attempt_id>')
def quiz_result(attempt_id):
    if 'user_id' not in session:
        return redirect('/login')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    cursor.execute("SELECT qa.*, q.question, q.correct_option FROM quiz_answers qa JOIN questions q ON qa.question_id
= q.id WHERE qa.attempt_id = %s", (attempt_id,))
    answers = cursor.fetchall()

    cursor.execute("SELECT * FROM quiz_attempts WHERE id = %s", (attempt_id,))
    attempt = cursor.fetchone()

    cursor.close()
    conn.close()

    return render_template("quiz_result.html", attempt=attempt, answers=answers)

@app.route('/dashboard')
def dashboard():
    if 'user_id' not in session:
        return redirect('/login')

    user_id = session['user_id']
    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    # Επιλογή θεματικών που είναι διαθέσιμες βάσει βαθμολογίας (>= unlocked_by_score)
    cursor.execute("""
        SELECT * FROM topics
        WHERE level = 1
        OR id IN (
            SELECT t2.id FROM topics t2
            WHERE t2.level <= (
                SELECT MAX(t.level)
                FROM topics t
                JOIN quizzes q ON q.topic_id = t.id
    """

```

```

        JOIN quiz_attempts qa ON qa.quiz_id = q.id
        WHERE qa.user_id = %s AND qa.score >= t.unlocked_by_score
    )
)
ORDER BY level
"""", (user_id,))

topics = cursor.fetchall()
cursor.close()
conn.close()

return render_template('dashboard.html', topics=topics)

@app.route('/admin/edit-page/<int:page_id>', methods=['GET', 'POST'])
def edit_topic_page(page_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    if request.method == 'POST':
        new_text = request.form['explanation']
        cursor.execute("UPDATE topic_pages SET explanation = %s WHERE id = %s", (new_text, page_id))
        conn.commit()
        flash("Η σελίδα ενημερώθηκε.", "success")
        return redirect(f'/admin/edit-page/{page_id}')

    cursor.execute("SELECT * FROM topic_pages WHERE id = %s", (page_id,))
    page = cursor.fetchone()

    cursor.close()
    conn.close()

    return render_template('admin_edit_page.html', page=page)

@app.route('/admin/pages')
def list_topic_pages():
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

```

```

conn = get_db_connection()
cursor = conn.cursor(dictionary=True)

cursor.execute("""
    SELECT tp.id, tp.page_number, tp.topic_id, t.title AS topic_title
    FROM topic_pages tp
    JOIN topics t ON tp.topic_id = t.id
    ORDER BY tp.topic_id, tp.page_number
""")
pages = cursor.fetchall()

cursor.close()
conn.close()

return render_template("admin_pages_list.html", pages=pages)

@app.route('/admin/delete-page/<int:page_id>', methods=['POST'])
def delete_topic_page(page_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor()
    cursor.execute("DELETE FROM topic_pages WHERE id = %s", (page_id,))
    conn.commit()
    cursor.close()
    conn.close()

    flash("Η σελίδα διαγράφηκε επιτυχώς.", "success")
    return redirect('/admin/pages')

@app.route('/admin/topics')
def admin_topics():
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)
    cursor.execute("SELECT * FROM topics ORDER BY level")
    topics = cursor.fetchall()
    cursor.close()

```

```

conn.close()

return render_template("admin_topics_list.html", topics=topics)

@app.route('/admin/new-topic', methods=['GET', 'POST'])
def new_topic():
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    if request.method == 'POST':
        title = request.form['title']
        description = request.form['description']
        image_url = request.form['image_url']
        level = request.form['level']
        unlock = request.form['unlocked_by_score']

        conn = get_db_connection()
        cursor = conn.cursor()
        cursor.execute("""
            INSERT INTO topics (title, description, image_url, level, unlocked_by_score, created_by)
            VALUES (%s, %s, %s, %s, %s, %s)
            """, (title, description, image_url, level, unlock, session['user_id']))
        conn.commit()
        cursor.close()
        conn.close()

        flash("Η θεματική δημιουργήθηκε.", "success")
        return redirect('/admin/topics')

    return render_template("admin_topic_form.html", mode="create")

@app.route('/admin/edit-topic/<int:topic_id>', methods=['GET', 'POST'])
def edit_topic(topic_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    if request.method == 'POST':
        title = request.form['title']

```

```

description = request.form['description']
image_url = request.form['image_url']
level = request.form['level']
unlock = request.form['unlocked_by_score']

cursor.execute("""
    UPDATE topics SET title=%s, description=%s, image_url=%s, level=%s, unlocked_by_score=%s
    WHERE id=%s
    """, (title, description, image_url, level, unlock, topic_id))
conn.commit()
cursor.close()
conn.close()

flash("Η θεματική ενημερώθηκε.", "success")
return redirect('/admin/topics')

cursor.execute("SELECT * FROM topics WHERE id = %s", (topic_id,))
topic = cursor.fetchone()
cursor.close()
conn.close()

return render_template("admin_topic_form.html", mode="edit", topic=topic)

@app.route('/admin/quizzes')
def admin_quizzes():
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)
    cursor.execute("""
        SELECT q.id, q.topic_id, q.total_time_minutes, q.active, t.title AS topic_title
        FROM quizzes q
        JOIN topics t ON q.topic_id = t.id
        ORDER BY t.level
    """)
    quizzes = cursor.fetchall()
    cursor.close()
    conn.close()

    return render_template("admin_quizzes_list.html", quizzes=quizzes)

```

```

@app.route('/admin/new-quiz', methods=['GET', 'POST'])
def new_quiz():
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    if request.method == 'POST':
        topic_id = request.form['topic_id']
        total_time = request.form['total_time_minutes']
        active = 1 if request.form.get('active') == 'on' else 0

        cursor.execute("""
            INSERT INTO quizzes (topic_id, total_time_minutes, active)
            VALUES (%s, %s, %s)
            """, (topic_id, total_time, active))

        conn.commit()
        cursor.close()
        conn.close()

        flash("Το quiz δημιουργήθηκε.", "success")
        return redirect('/admin/quizzes')

    cursor.execute("SELECT id, title FROM topics ORDER BY level")
    topics = cursor.fetchall()
    cursor.close()
    conn.close()

    return render_template("admin_quiz_form.html", mode="create", topics=topics)

@app.route('/admin/edit-quiz/<int:quiz_id>', methods=['GET', 'POST'])
def edit_quiz(quiz_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    if request.method == 'POST':

```

```

topic_id = request.form['topic_id']
total_time = request.form['total_time_minutes']
active = 1 if request.form.get('active') == 'on' else 0

cursor.execute("""
    UPDATE quizzes SET topic_id = %s, total_time_minutes = %s, active = %s
    WHERE id = %s
""", (topic_id, total_time, active, quiz_id))

conn.commit()
cursor.close()
conn.close()

flash("To quiz ενημερώθηκε.", "success")
return redirect('/admin/quizzes')

cursor.execute("SELECT * FROM quizzes WHERE id = %s", (quiz_id,))
quiz = cursor.fetchone()

cursor.execute("SELECT id, title FROM topics ORDER BY level")
topics = cursor.fetchall()

cursor.close()
conn.close()

return render_template("admin_quiz_form.html", mode="edit", quiz=quiz, topics=topics)

@app.route('/admin/quiz-questions/<int:quiz_id>')
def quiz_questions(quiz_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    cursor.execute("""
        SELECT q.*, t.title AS topic_title FROM quizzes q
        JOIN topics t ON t.id = q.topic_id
        WHERE q.id = %s
    """, (quiz_id,))
    quiz = cursor.fetchone()

```

```

cursor.execute("""
    SELECT qs.*, tp.page_number
    FROM questions qs
    JOIN topic_pages tp ON qs.page_id = tp.id
    WHERE qs.quiz_id = %s
    ORDER BY qs.id
""", (quiz_id,))
questions = cursor.fetchall()

cursor.close()
conn.close()

return render_template("admin_quiz_questions.html", quiz=quiz, questions=questions)

@app.route('/admin/add-question/<int:quiz_id>', methods=['GET', 'POST'])
def add_question(quiz_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    # Quiz & Topic
    cursor.execute("""
        SELECT q.id, t.id AS topic_id, t.title
        FROM quizzes q JOIN topics t ON q.topic_id = t.id
        WHERE q.id = %s
    """, (quiz_id,))
    quiz = cursor.fetchone()

    # Σελίδες του topic για dropdown
    cursor.execute("SELECT id, page_number FROM topic_pages WHERE topic_id = %s ORDER BY page_number",
    (quiz['topic_id'],))
    pages = cursor.fetchall()

    if request.method == 'POST':
        page_id = request.form['page_id']
        question = request.form['question']
        options = [request.form['option1'], request.form['option2'], request.form['option3'], request.form['option4']]
        correct = int(request.form['correct_option'])

        cursor.execute("""

```

```

INSERT INTO questions (quiz_id, page_id, question, option1, option2, option3, option4, correct_option)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
"""', (quiz_id, page_id, question, *options, correct))

conn.commit()
cursor.close()
conn.close()

flash("Η ερώτηση προστέθηκε.", "success")
return redirect(f'/admin/quiz-questions/{quiz_id}')

cursor.close()
conn.close()

return render_template("admin_question_form.html", quiz=quiz, pages=pages, mode='create')

@app.route('/admin/edit-question/<int:question_id>', methods=['GET', 'POST'])
def edit_question(question_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    # Φέρνουμε την ερώτηση + σελίδες του topic
    cursor.execute("""
        SELECT q.*, tp.topic_id, t.title AS topic_title
        FROM questions q
        JOIN topic_pages tp ON q.page_id = tp.id
        JOIN topics t ON tp.topic_id = t.id
        WHERE q.id = %s
    """, (question_id,))
    question = cursor.fetchone()

    cursor.execute("SELECT id, page_number FROM topic_pages WHERE topic_id = %s ORDER BY page_number",
(question['topic_id'],))
    pages = cursor.fetchall()

    if request.method == 'POST':
        page_id = request.form['page_id']
        qtext = request.form['question']
        options = [request.form['option1'], request.form['option2'], request.form['option3'], request.form['option4']]

```

```

correct = int(request.form['correct_option'])

cursor.execute("""
    UPDATE questions
    SET page_id=%s, question=%s, option1=%s, option2=%s, option3=%s, option4=%s, correct_option=%s
    WHERE id=%s
    """, (page_id, qtext, *options, correct, question_id))

conn.commit()
cursor.close()
conn.close()

flash("Η ερώτηση ενημερώθηκε.", "success")
return redirect(f'/admin/quiz-questions/{question["quiz_id"]}')
```

```

cursor.close()
conn.close()

return render_template("admin_question_form.html", mode='edit', question=question, pages=pages, quiz=question)

@app.route('/admin/delete-question/<int:question_id>', methods=['POST'])
def delete_question(question_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    cursor.execute("SELECT quiz_id FROM questions WHERE id = %s", (question_id,))
    q = cursor.fetchone()

    cursor.execute("DELETE FROM questions WHERE id = %s", (question_id,))
    conn.commit()
    cursor.close()
    conn.close()

    flash("Η ερώτηση διαγράφηκε.", "success")
    return redirect(f'/admin/quiz-questions/{q["quiz_id"]}')
```

```

@app.route('/admin/export-results/<int:quiz_id>')
def export_results(quiz_id):
    if 'user_id' not in session or session.get('role') != 'admin':

```

```

    flash("Μόνο για διαχειριστές", "danger")
    return redirect('/dashboard')

conn = get_db_connection()
cursor = conn.cursor(dictionary=True)

cursor.execute("""
    SELECT u.fullname, u.email, qa.score, qa.submitted_at
    FROM quiz_attempts qa
    JOIN users u ON qa.user_id = u.id
    WHERE qa.quiz_id = %s
    ORDER BY qa.submitted_at DESC
""", (quiz_id,))
results = cursor.fetchall()
cursor.close()
conn.close()

output = StringIO()
writer = csv.writer(output)
writer.writerow(['Όνοματεπώνυμο', 'Email', 'Βαθμολογία (%)', 'Ημερομηνία Υποβολής'])

for r in results:
    writer.writerow([r['fullname'], r['email'], r['score'], r['submitted_at']])

output.seek(0)
return Response(output.getvalue(), mimetype='text/csv',
                 headers={"Content-Disposition": f"attachment; filename=quiz_{quiz_id}_results.csv"})

@app.route('/admin/quiz-stats/<int:quiz_id>')
def quiz_stats(quiz_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για admin", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    cursor.execute("SELECT * FROM quizzes WHERE id = %s", (quiz_id,))
    quiz = cursor.fetchone()

    cursor.execute("""
        SELECT score FROM quiz_attempts
        WHERE quiz_id = %s

```

```

        """ , (quiz_id,))
    scores = [row['score'] for row in cursor.fetchall()]
    total = len(scores)
    avg = round(sum(scores) / total, 2) if total > 0 else 0
    passed = len([s for s in scores if s >= 70])

    cursor.close()
    conn.close()

    return render_template("admin_quiz_stats.html", quiz=quiz, scores=scores, average=avg, passed=passed, total=total)

@app.route('/admin/quiz-question-stats/<int:quiz_id>')
def quiz_question_stats(quiz_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για διαχειριστές", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    # Φέρνουμε τις ερωτήσεις και τον αριθμό σωστών / συνολικών
    cursor.execute("""
        SELECT q.id, q.question,
            COUNT(a.id) AS total,
            SUM(CASE WHEN a.correct = 1 THEN 1 ELSE 0 END) AS correct
        FROM questions q
        LEFT JOIN quiz_answers a ON a.question_id = q.id
        WHERE q.quiz_id = %s
        GROUP BY q.id
        ORDER BY q.id
    """, (quiz_id,))
    stats = cursor.fetchall()

    cursor.execute("SELECT * FROM quizzes WHERE id = %s", (quiz_id,))
    quiz = cursor.fetchone()

    cursor.close()
    conn.close()

    return render_template("admin_quiz_question_stats.html", quiz=quiz, stats=stats)

@app.route('/my-quizzes')
def my_quizzes():

```

```

if 'user_id' not in session or session.get('role') != 'student':
    flash("Μόνο για φοιτητές", "danger")
    return redirect('/dashboard')

user_id = session['user_id']
conn = get_db_connection()
cursor = conn.cursor(dictionary=True)

# Quiz που έχει συμμετάσχει
cursor.execute("""
    SELECT q.id, t.title AS topic_title, qa.score, qa.submitted_at
    FROM quiz_attempts qa
    JOIN quizzes q ON qa.quiz_id = q.id
    JOIN topics t ON q.topic_id = t.id
    WHERE qa.user_id = %s
    ORDER BY qa.submitted_at DESC
""", (user_id,))
attempts = cursor.fetchall()

cursor.close()
conn.close()
return render_template("student_my_quizzes.html", attempts=attempts)

@app.route('/admin/users')
def admin_users():
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για admin", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)
    cursor.execute("SELECT id, fullname, email, am, role, created_at FROM users ORDER BY role, fullname")
    users = cursor.fetchall()
    cursor.close()
    conn.close()

    return render_template("admin_users.html", users=users)

@app.route('/admin/new-user', methods=['GET', 'POST'])
def new_user():
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για admin", "danger")
        return redirect('/dashboard')

```

```

if request.method == 'POST':
    fullname = request.form['fullname']
    email = request.form['email']
    am = request.form['am']
    role = request.form['role']
    password = request.form['password']

    conn = get_db_connection()
    cursor = conn.cursor()
    cursor.execute("""
        INSERT INTO users (fullname, email, am, role, password)
        VALUES (%s, %s, %s, %s, SHA2(%s, 256))
        """, (fullname, email, am, role, password))
    conn.commit()
    cursor.close()
    conn.close()

    flash("Ο χρήστης δημιουργήθηκε.", "success")
    return redirect('/admin/users')

return render_template("admin_user_form.html", mode="create")

@app.route('/admin/edit-user/<int:user_id>', methods=['GET', 'POST'])
def edit_user(user_id):
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για admin", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    if request.method == 'POST':
        fullname = request.form['fullname']
        email = request.form['email']
        am = request.form['am']
        role = request.form['role']
        password = request.form.get('password')

    if password:
        cursor.execute("""
            UPDATE users SET fullname=%s, email=%s, am=%s, role=%s, password=SHA2(%s, 256)
            WHERE id=%s
        """)

```

```

        """ , (fullname, email, am, role, password, user_id))
else:
    cursor.execute("""
        UPDATE users SET fullname=%s, email=%s, am=%s, role=%s
        WHERE id=%s
        """ , (fullname, email, am, role, user_id))

    conn.commit()
    cursor.close()
    conn.close()

    flash("Ο χρήστης ενημερώθηκε.", "success")
    return redirect('/admin/users')

cursor.execute("SELECT * FROM users WHERE id = %s", (user_id,))
user = cursor.fetchone()
cursor.close()
conn.close()

return render_template("admin_user_form.html", mode="edit", user=user)

@app.route('/admin/new-page', methods=['GET', 'POST'])
def new_topic_page():
    if 'user_id' not in session or session.get('role') != 'admin':
        flash("Μόνο για admin", "danger")
        return redirect('/dashboard')

    conn = get_db_connection()
    cursor = conn.cursor(dictionary=True)

    if request.method == 'POST':
        topic_id = request.form['topic_id']
        page_number = request.form['page_number']
        image_url = request.form['image_url']
        explanation = request.form['explanation']

        cursor.execute("""
            INSERT INTO topic_pages (topic_id, page_number, image_url, explanation)
            VALUES (%s, %s, %s, %s)
            """ , (topic_id, page_number, image_url, explanation))

        conn.commit()
        cursor.close()

```

```
conn.close()

flash("Η σελίδα δημιουργήθηκε.", "success")
return redirect('/admin/pages')

# Fetch topics for dropdown
cursor.execute("SELECT id, title FROM topics ORDER BY level")
topics = cursor.fetchall()
cursor.close()
conn.close()

return render_template("admin_new_page.html", topics=topics)

if __name__ == '__main__':
    app.run(debug=True)
```