

Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ΔΙΑΔΡΑΣΤΙΚΗΣ ΕΚΠΑΙΔΕΥΤΙΚΗΣ
ΠΛΑΤΦΟΡΜΑΣ ΓΙΑ ΤΗΝ ΕΚΜΑΘΗΣΗ ΤΗΣ ΛΟΓΙΚΗΣ**

Στυλιάνα Φέσια

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2026

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάπτυξη Διαδραστικής Εκπαιδευτικής Πλατφόρμας για την Εκμάθηση της Λογικής
Στυλιάνα Φέσια

Επιβλέπων Καθηγητής
Ανδρέας Πιερής

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων
απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου
Κύπρου

Μάιος 2026

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον καθηγητή Ανδρέα Πιερή για όλη την καθοδήγηση, την βοήθεια και την ανατροφοδότηση που μου παρείχε κατά την διάρκεια εκπόνησης της Ατομικής Διπλωματικής Εργασίας μου σαν επιβλέπων καθηγητής.

Επίσης, ευχαριστώ την οικογένεια μου και τους φίλους μου που ήταν δίπλα μου σε όλη αυτή την πορεία. Η υποστήριξη, η κατανόηση και η ενθάρρυνση τους ήταν πολύτιμες και συνέβαλαν σημαντικά στην ολοκλήρωση των σπουδών και της παρούσας εργασίας.

Περίληψη

Η παρούσα διπλωματική εργασία αφορά την ανάπτυξη μιας διαδικτυακής εκπαιδευτικής πλατφόρμας για την υποστήριξη της εκμάθησης της Λογικής, με κύρια έμφαση στην Προτασιακή Λογική. Η ανάγκη ανάπτυξης της πλατφόρμας προκύπτει από τον θεωρητικό χαρακτήρα του αντικειμένου και τις δυσκολίες που συχνά αντιμετωπίζουν οι φοιτητές κατά την κατανόηση βασικών εννοιών και την επίλυση ασκήσεων.

Στόχος της εργασίας ήταν η δημιουργία ενός ολοκληρωμένου περιβάλλοντος μάθησης που συνδυάζει θεωρητικό υλικό, διαδραστικές ασκήσεις και quizzes αξιολόγησης, ώστε να ενισχύεται τόσο η κατανόηση όσο και η πρακτική εξάσκηση των χρηστών. Η πλατφόρμα υποστηρίζει διαφορετικούς ρόλους χρηστών, διαχείριση περιεχομένου, εκπαιδευτικές δραστηριότητες και μηχανισμούς ανατροφοδότησης μέσω επεξηγήσεων και αξιολόγησης απαντήσεων.

Το περιεχόμενο της πλατφόρμας περιλαμβάνει θεωρία Προτασιακής Λογικής, ασκήσεις διαφορετικών τύπων, επεξηγήσεις λύσεων και quizzes, επιτρέποντας στους χρήστες να εφαρμόζουν άμεσα τις γνώσεις που αποκτούν. Παρότι η παρούσα υλοποίηση επικεντρώνεται κυρίως στην Προτασιακή Λογική, η αρχιτεκτονική του συστήματος επιτρέπει μελλοντική επέκταση σε επιπλέον θεματικές ενότητες.

Η αξιολόγηση της πλατφόρμας πραγματοποιήθηκε μέσω ερωτηματολογίου σε φοιτητές Πληροφορικής που είχαν παρακολουθήσει σχετικό μάθημα, με τα αποτελέσματα να δείχνουν θετική αποδοχή ως προς τη χρηστικότητα, τη δομή και τη συμβολή της στη μαθησιακή διαδικασία. Τα αποτελέσματα υποδεικνύουν ότι η ενσωμάτωση διαδραστικών ασκήσεων και ανατροφοδότησης μπορεί να υποστηρίξει αποτελεσματικά την εκμάθηση της Προτασιακής Λογικής.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Κίνητρο και Σκοπός Διπλωματικής Εργασίας	1
	1.2 Αντικείμενο και Πεδίο Εφαρμογής	2
	1.3 Μεθοδολογία Ανάπτυξης	3
	1.4 Δομή Διπλωματικής Εργασίας	3
Κεφάλαιο 2	Θεωρητικό Υπόβαθρο Προτασιακής Λογικής.....	5
	2.1 Εισαγωγή στην Προτασιακή Λογική	5
	2.2 Προτάσεις	6
	2.3 Λογικοί Τελεστές	6
	2.4 Προτεραιότητα Λογικών Τελεστών	7
	2.5 Πίνακες Αληθείας	7
	2.6 Προτασιακές Ισοδυναμίες	7
	2.7 Εφαρμογές της Προτασιακής Λογικής	8
	2.8 Συναρτησιακή Πληρότητα	8
	2.9 Σύνδεση Θεωρίας με την Πλατφόρμα	9
Κεφάλαιο 3	Ανάλυση Απαιτήσεων Συστήματος.....	10
	3.1 Περιγραφή του Συστήματος	10
	3.2 Κατηγορίες Χρηστών και Ρόλοι	11
	3.2.1 Φοιτητής	12
	3.2.2 Διαχειριστής	12
	3.2.3 Έλεγχος Πρόσβασης	13
	3.3 Λειτουργικές Απαιτήσεις	13
	3.3.1 Λειτουργικές Απαιτήσεις Φοιτητή	13
	3.3.2 Λειτουργικές Απαιτήσεις Διαχειριστή	14
	3.3.3 Γενικές Λειτουργικές Απαιτήσεις Συστήματος	15
	3.4 Μη Λειτουργικές Απαιτήσεις	15
	3.4.1 Ευχρηστία	16
	3.4.2 Ασφάλεια	16
	3.4.3 Απόδοση	16
	3.4.4 Επεκτασιμότητα	16
	3.5 Use Cases	17

Κεφάλαιο 4	Σχεδίαση Συστήματος.....	19
4.1	Αρχιτεκτονική Συστήματος MVC	19
4.2	Δομή Βάσης Δεδομένων	21
4.2.1	Πίνακες Βάσης Δεδομένων	21
4.2.2	ER Διάγραμμα	27
4.3	Σχεδίαση Διεπαφής Χρήστη (UX/UI)	28
4.3.1	Σχεδιαστικές Αρχές Διεπαφής	28
4.3.2	Διεπαφή Φοιτητή	29
4.3.2.1	Dashboard Φοιτητή	30
4.3.2.2	Σελίδα Θεωρίας	30
4.3.2.3	Σελίδα Ασκήσεων	33
4.3.2.4	Σελίδα Quizzes	38
4.3.3	Διεπαφή Διαχειριστή	39
4.3.3.1	Dashboard Διαχειριστή	39
4.3.3.2	Σελίδα Διαχείρισης Θεωρίας	40
4.3.3.3	Σελίδα Διαχείρισης Ασκήσεων	43
4.3.3.4	Σελίδα Διαχείρισης Quiz	50
4.3.3.5	Σελίδα Διαχείρισης Χρηστών	53
4.3.3.6	Σελίδα Προβολής ως Χρήστης	55
4.3.4	Πλοήγηση και Εμπειρία Χρήστη	56
4.4	Σχεδίαση Λειτουργιών Ασκήσεων	59
4.5	Σχεδίαση Λειτουργιών Quiz	60
 Κεφάλαιο 5	 Υλοποίηση Συστήματος.....	 63
5.1	Τεχνολογίες Υλοποίησης	63
5.2	Οργάνωση και Δομή Κώδικα	65
5.2.1	Δομή Φακέλων Πλατφόρμας	66
5.2.2	Υλοποίηση Controllers και Repositories	67
5.3	Υλοποίηση Συστήματος Σύνδεσης και Ελέγχου Πρόσβασης	70
5.3.1	Εγγραφή Χρήστη	71
5.3.2	Σύνδεση Χρήστη	72
5.3.3	Έλεγχος Ρόλων και Πρόσβασης	73
5.4	Υλοποίηση Διαχείρισης Θεωρίας	74
5.5	Υλοποίηση Ασκήσεων	75
5.5.1	Υλοποίηση Ασκήσεων Drag & Drop	76
5.5.2	Υλοποίηση Ασκήσεων Επιλογής Κανόνων	76
5.5.3	Υλοποίηση Ασκήσεων Εξέτασης Ισοδυναμίας	77

5.5.4 Υλοποίηση Ασκήσεων Ελέγχου Απόδειξης	78
5.6 Υλοποίηση Quizzes	78
5.6.1 Δημιουργία Quiz	79
5.6.2 Δημιουργία και Διαχείριση Ερωτήσεων	79
5.6.3 Υποβολή Απαντήσεων Quiz	79
5.6.4 Υπολογισμός Βαθμολογίας και Penalty System	80
5.7 Υλοποίηση Διαχείρισης Χρηστών	80
 ΚΕΦΑΛΑΙΟ 6 Συμπεράσματα	 82
6.1 Μεθοδολογία Αξιολόγησης	82
6.2 Ανάλυση Αποτελεσμάτων	83
6.3 Συμπεράσματα	84
6.4 Μελλοντική Εργασία	84
 Βιβλιογραφία	 86
 Παράρτημα Α.....	 88
 Παράρτημα Β.....	 9

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο και Σκοπός Διπλωματικής Εργασίας	1
1.2 Αντικείμενο και Πεδίο Εφαρμογής	2
1.3 Μεθοδολογία Ανάπτυξης	3
1.4 Δομή Διπλωματικής Εργασίας	3

1.1 Κίνητρο και Σκοπός Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία έχει ως κύριο σκοπό την ανάπτυξη μιας διαδραστικής εκπαιδευτικής πλατφόρμας για την υποστήριξη της εκμάθησης της Λογικής και στο παρόν στάδιο, της Προτασιακής Λογικής. Η πλατφόρμα σχεδιάστηκε και υλοποιήθηκε ως διαδικτυακή πλατφόρμα, με στόχο την παροχή ενός ολοκληρωμένου περιβάλλοντος μάθησης που συνδυάζει θεωρητικό υλικό, ασκήσεις και quiz αξιολόγησης.

Η Λογική αποτελεί θεμελιώδες αντικείμενο της Πληροφορικής, καθώς παρέχει τα βασικά εργαλεία για την ανάλυση προτάσεων και την διατύπωση αποδείξεων. Ειδικότερα, η Προτασιακή Λογική είναι ένα από τα πρώτα θεωρητικά πεδία με τα οποία έρχονται σε επαφή οι φοιτητές της Πληροφορικής, αφού εισάγει την τυπική αναπαράσταση προτάσεων, τη χρήση λογικών τελεστών και την εξαγωγή συμπερασμάτων μέσα από κανόνες λογικής.

Η ανάγκη για την ανάπτυξη ενός τέτοιου συστήματος προκύπτει από τον θεωρητικό χαρακτήρα της Λογικής, οι βασικές της έννοιες ενδέχεται να είναι απαιτητικές για τους φοιτητές, ιδιαίτερα κατά τα πρώτα στάδια των σπουδών τους. Στο πλαίσιο μαθημάτων όπως το ΕΠΑ111 – Διακριτές Δομές στην Πληροφορική και Υπολογισμό, το οποίο διδάσκεται στο πρώτο εξάμηνο των σπουδών της Πληροφορικής στο Πανεπιστήμιο Κύπρου, οι φοιτητές καλούνται να κατανοήσουν κάποιες βασικές έννοιες της Προτασιακής Λογικής. Οι έννοιες αυτές μπορούν να γίνουν πιο κατανοητές μέσα από ένα περιβάλλον που προσφέρει οργανωμένο υλικό, πρακτικές ασκήσεις και άμεση ανατροφοδότηση.

Η πλατφόρμα Logic Learning αναπτύχθηκε με σκοπό να λειτουργήσει υποστηρικτικά στη διαδικασία μάθησης. Δεν αντικαθιστά την διδασκαλία ή το διδακτικό υλικό του μαθήματος, αλλά προσφέρει ένα συμπληρωματικό ψηφιακό περιβάλλον όπου ο χρήστης μπορεί να μελετήσει θεωρία, να λύσει ασκήσεις και να ελέγξει την κατανόηση του μέσα από quizzes. Με αυτό τον τρόπο, δίνεται η δυνατότητα στον φοιτητή να εξασκείται με πιο ενεργό τρόπο και να λαμβάνει άμεση ανατροφοδότηση για τις απαντήσεις του.

Επομένως, ο στόχος της διπλωματικής είναι η υλοποίηση μιας διαδικτυακής εφαρμογής που να είναι εύχρηστη, οργανωμένη και επεκτάσιμη, ώστε να μπορεί να αξιοποιηθεί για την εκμάθηση της Προτασιακής Λογικής.

1.2 Αντικείμενο και Πεδίο Εφαρμογής

Το αντικείμενο της παρούσας εργασίας είναι η ανάπτυξη διαδικτυακής εκπαιδευτικής πλατφόρμας για την εκμάθηση της Λογικής, με έμφαση στην Προτασιακή Λογική. Η πλατφόρμα απευθύνεται σε οποιονδήποτε επιθυμεί να μελετήσει και να εξοικειωθεί με βασικές έννοιες Λογικής στην Πληροφορική, ενώ ειδικότερα μπορεί να αξιοποιηθεί από τους φοιτητές που παρακολουθούν το μάθημα ΕΠΛ111.

Στην παρούσα φάση, το εκπαιδευτικό υλικό της πλατφόρμας καλύπτει κεφάλαια της Προτασιακής Λογικής. Τα κεφάλαια που έχουν οργανωθεί περιλαμβάνουν τις προτάσεις, τους λογικούς τελεστές, την προτεραιότητα λογικών τελεστών, τους πίνακες αληθείας, τις εφαρμογές της Προτασιακής Λογικής, τις προτασιακές ισοδυναμίες και την συναρτησιακή πληρότητα. Αυτά τα κεφάλαια καλύπτουν τους βασικούς άξονες κατανόησης της Προτασιακής Λογικής.

Η πλατφόρμα απευθύνεται σε δύο κατηγορίες χρηστών: τον φοιτητή και τον διαχειριστή. Ο φοιτητής μπορεί να μελετά το θεωρητικό υλικό, να λύνει ασκήσεις διαφορετικών μορφών και να κάνει quizzes πολλαπλής επιλογής, ώστε να εξασκείται και να ελέγχει τον βαθμό κατανόησης του. Αντίστοιχα, ο διαχειριστής έχει τη δυνατότητα να οργανώνει και να διαχειρίζεται το περιεχόμενο της πλατφόρμας, δημιουργώντας ή τροποποιώντας θεωρητικές ενότητες και τα κεφάλαιά τους, ασκήσεις και quizzes, καθώς και να βλέπει πληροφορίες που σχετίζονται με τις προσπάθειες των χρηστών στα quizzes.

Στην παρούσα φάση, η πλατφόρμα επικεντρώνεται στην Προτασιακή Λογική, η οποία αποτελεί βασική εισαγωγική ενότητα της Λογικής στην Πληροφορική. Η δομή της πλατφόρμας, ωστόσο, έχει σχεδιαστεί με τρόπο που επιτρέπει μελλοντική επέκταση σε επιπλέον θεματικές ενότητες, όπως η Κατηγορηματική Λογική.

1.3 Μεθοδολογία Ανάπτυξης

Η ανάπτυξη της πλατφόρμας ακολούθησε μια επαναληπτική διαδικασία, με σταδιακή υλοποίηση και βελτίωση των επιμέρους λειτουργιών. Αρχικά υλοποιήθηκε η βασική δομή της διαδικτυακής εφαρμογής, ώστε να δημιουργηθεί ένα λειτουργικό περιβάλλον χρήσης. Στη συνέχεια αναπτύχθηκαν οι βασικοί μηχανισμοί του συστήματος, όπως η διαχείριση χρηστών, η πρόσβαση με βάση ρόλους, η διαχείριση θεωρητικού περιεχομένου, τα quizzes και οι ασκήσεις.

Μετά την ολοκλήρωση της βασικής λειτουργικότητας, συγκεντρώθηκε, οργανώθηκε και ενσωματώθηκε στην πλατφόρμα το εκπαιδευτικό υλικό της Προτασιακής Λογικής. Το υλικό αυτό συνδυάστηκε με ασκήσεις και ερωτήσεις αξιολόγησης, ώστε να υποστηρίζεται τόσο η μελέτη όσο και η πρακτική εξάσκηση του χρήστη.

Στη συνέχεια πραγματοποιήθηκαν δοκιμές για τον έλεγχο της ορθής λειτουργίας της εφαρμογής, τόσο ως προς τις λειτουργίες του φοιτητή όσο και ως προς τις λειτουργίες του διαχειριστή. Ελέγχθηκαν η σύνδεση χρηστών, η πρόσβαση ανά ρόλο, η προβολή θεωρίας, η επίλυση quizzes, η επίλυση ασκήσεων και η διαχείριση περιεχομένου.

Τέλος, η πλατφόρμα αξιολογήθηκε από φοιτητές πληροφορικής μέσω ερωτηματολογίου. Η ανατροφοδότηση που συλλέχθηκε αξιοποιήθηκε για την εξαγωγή συμπερασμάτων σχετικά με τη χρηστικότητα, την κατανόηση του περιεχομένου και τη συνολική εμπειρία χρήσης.

1.4 Δομή Διπλωματικής Εργασίας

Στο Κεφάλαιο 1 παρουσιάζεται η εισαγωγή της διπλωματικής εργασίας. Περιγράφεται το κίνητρο και ο σκοπός της διπλωματικής εργασίας, το αντικείμενο και το πεδίο εφαρμογής της πλατφόρμας, η μεθοδολογία που ακολουθήθηκε και η συνολική δομή της εργασίας.

Στο Κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο της Προτασιακής Λογικής.

Στο Κεφάλαιο 3 γίνεται η ανάλυση των απαιτήσεων του συστήματος. Παρουσιάζονται οι βασικές λειτουργικές και μη λειτουργικές απαιτήσεις της πλατφόρμας, οι κατηγορίες χρηστών, τα δικαιώματά τους και οι βασικές περιπτώσεις χρήσης.

Στο Κεφάλαιο 4 περιγράφεται η σχεδίαση του συστήματος. Αναλύεται η αρχιτεκτονική της πλατφόρμας, η οργάνωση της βάσης δεδομένων και ο σχεδιασμός της διεπαφής χρήστη (φοιτητή και διαχειριστή).

Στο Κεφάλαιο 5 παρουσιάζεται η υλοποίηση της πλατφόρμας. Περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν, η οργάνωση του κώδικα, η υλοποίηση του μηχανισμού σύνδεσης και ελέγχου πρόσβασης, η υλοποίηση της διαχείρισης εκπαιδευτικού περιεχομένου, των ασκήσεων, των quizzes και των χρηστών.

Στο κεφάλαιο 6 παρουσιάζεται η μεθοδολογία αξιολόγησης, η ανάλυση των αποτελεσμάτων, τα συμπεράσματα που προέκυψαν και οι πιθανές μελλοντικές επεκτάσεις.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο Προτασιακής Λογικής

2.1 Εισαγωγή στην Προτασιακή Λογική	5
2.2 Προτάσεις	6
2.3 Λογικοί Τελεστές	6
2.4 Προτεραιότητα Λογικών Τελεστών	7
2.5 Πίνακες Αληθείας	7
2.6 Προτασιακές Ισοδυναμίες	7
2.7 Εφαρμογές της Προτασιακής Λογικής	8
2.8 Συναρτησιακή Πληρότητα	8
2.9 Σύνδεση Θεωρίας με την Πλατφόρμα	9

2.1 Εισαγωγή στην Προτασιακή Λογική [2]

Η Προτασιακή Λογική αποτελεί βασικό μέρος της Λογικής και ασχολείται με προτάσεις που μπορούν να χαρακτηριστούν ως αληθείς ή ψευδείς. Μέσα από τη χρήση συμβόλων και λογικών τελεστών, οι προτάσεις μπορούν να εκφραστούν με πιο τυπικό τρόπο, ώστε να μελετηθεί η σχέση μεταξύ τους και να εξαχθούν συμπεράσματα.

Στην Πληροφορική, η Προτασιακή Λογική είναι ιδιαίτερα σημαντική, καθώς βοηθά στην κατανόηση εννοιών όπως οι συνθήκες, οι κανόνες, οι λογικές εκφράσεις και η ορθότητα συλλογισμών. Για τον λόγο αυτό διδάσκεται συχνά στα πρώτα στάδια σπουδών, καθώς προσφέρει το απαραίτητο υπόβαθρο για πιο σύνθετα θέματα που εμφανίζονται στη συνέχεια των σπουδών.

Το εκπαιδευτικό περιεχόμενο που παρουσιάζεται στην παρούσα εργασία βασίστηκε στο διδακτικό υλικό του μαθήματος ΕΠΛ111 [3], καθώς και στο βιβλίο του Kenneth H. Rosen [2]. Το πλήρες θεωρητικό υλικό που ενσωματώθηκε στα κεφάλαια στην πλατφόρμα παρατίθεται στο Παράρτημα Β.

2.2 Προτάσεις

Μία πρόταση ορίζεται ως μία δήλωση η οποία έχει συγκεκριμένη τιμή αληθείας, δηλαδή μπορεί να είναι είτε αληθής είτε ψευδής, αλλά όχι και τα δύο ταυτόχρονα. Οι προτάσεις συμβολίζονται συνήθως με κεφαλαία γράμματα όπως P, Q, R.

Για παράδειγμα η πρόταση «Το Παρίσι είναι η πρωτεύουσα της Γαλλίας» είναι αληθής, ενώ η πρόταση « $7 > 10$ » είναι ψευδής. Αντίθετα, εκφράσεις όπως ερωτήσεις, εντολές ή προτάσεις που περιέχουν μεταβλητές δεν θεωρούνται προτάσεις, αφού δεν μπορούν να χαρακτηριστούν ως αληθείς ή ψευδείς.

2.3 Λογικοί Τελεστές

Από απλές προτάσεις μπορούν να δημιουργηθούν σύνθετες προτάσεις με τη χρήση λογικών τελεστών. Οι βασικοί τελεστές της Προτασιακής Λογικής είναι οι εξής:

- Άρνηση: $\neg$
Αντιστρέφει την τιμή αληθείας μιας πρότασης.
- Σύζευξη: $\wedge$
Η σύζευξη $P \wedge Q$ είναι αληθής μόνο όταν και οι δύο προτάσεις είναι αληθείς.
- Διάζευξη: $\vee$
Η διάζευξη $P \vee Q$ είναι αληθής όταν τουλάχιστον μία από τις δύο προτάσεις είναι αληθής.
- Αποκλειστική διάζευξη: $\oplus$
Η αποκλειστική διάζευξη $P \oplus Q$ είναι αληθής αν ακριβώς μία από τις δύο προτάσεις είναι αληθής, όχι και οι δύο.
- Συνεπαγωγή: $\rightarrow$
Η συνεπαγωγή εκφράζει τη σχέση «αν... τότε...» και είναι ψευδής μόνο όταν η αρχική πρόταση είναι αληθής και το συμπέρασμα ψευδές.
- Διπλή συνεπαγωγή: $\leftrightarrow$
Η διπλή συνεπαγωγή είναι αληθής όταν οι δύο προτάσεις έχουν την ίδια τιμή αληθείας.

Οι λογικοί τελεστές αποτελούν το βασικό εργαλείο για τη δημιουργία και ανάλυση σύνθετων λογικών εκφράσεων.

2.4 Προτεραιότητα Λογικών Τελεστών

Σε σύνθετες λογικές προτάσεις είναι συχνό να εμφανίζονται περισσότεροι από ένας λογικοί τελεστές. Για να είναι σαφές με ποια σειρά πρέπει να διαβαστεί και να αξιολογηθεί μια τέτοια πρόταση, χρησιμοποιείται η προτεραιότητα λογικών τελεστών.

Στην Προτασιακή Λογική, η άρνηση έχει την υψηλότερη προτεραιότητα, επειδή εφαρμόζεται άμεσα στην πρόταση που ακολουθεί. Στη συνέχεια αξιολογούνται η σύζευξη, η διάζευξη, η αποκλειστική διάζευξη, η συνεπαγωγή και τέλος η διπλή συνεπαγωγή. Παρόλο που η προτεραιότητα των τελεστών καθορίζει τη σωστή ερμηνεία, η χρήση παρενθέσεων είναι συχνά προτιμότερη, καθώς κάνει τις λογικές εκφράσεις πιο ευανάγνωστες και μειώνει την πιθανότητα λανθασμένης κατανόησης.

Για παράδειγμα, η έκφραση $P \vee Q \wedge \neg R$ διαβάζεται ως $P \vee (Q \wedge (\neg R))$, λόγω της προτεραιότητας των τελεστών.

2.5 Πίνακες Αληθείας

Οι πίνακες αληθείας χρησιμοποιούνται για την μελέτη σύνθετων προτάσεων. Μέσω αυτών παρουσιάζονται όλοι οι δυνατοί συνδυασμοί τιμών αληθείας των προτασιακών μεταβλητών και υπολογίζεται η τελική τιμή της σύνθετης πρότασης για κάθε περίπτωση. Αν μια λογική έκφραση περιέχει n μεταβλητές, τότε ο πίνακας αληθείας θα περιέχει 2^n γραμμές.

Με τη χρήση πινάκων αληθείας μπορεί να ελεγχθεί αν μια πρόταση είναι ταυτολογία, αντίφαση ή ενδεχομενική. Για παράδειγμα, αν η τελική στήλη ενός πίνακα περιέχει μόνο τιμές αληθείας True, τότε η πρόταση είναι ταυτολογία, ενώ αν περιέχει μόνο False, τότε είναι αντίφαση.

2.6 Προτασιακές Ισοδυναμίες

Δύο λογικές προτάσεις θεωρούνται ισοδύναμες όταν έχουν την ίδια τιμή αληθείας για κάθε δυνατό συνδυασμό τιμών των μεταβλητών τους, δηλαδή ανεξάρτητα από τις τιμές των μεταβλητών, οι δύο προτάσεις δίνουν πάντα το ίδιο αποτέλεσμα.

Οι σύνθετες προτάσεις μπορούν να ταξινομηθούν σε τρεις βασικές κατηγορίες. Οι ταυτολογίες είναι προτάσεις που είναι πάντα αληθείς, οι αντιφάσεις είναι προτάσεις που είναι πάντα ψευδείς, ενώ οι ενδεχομενικές προτάσεις μπορεί να είναι άλλοτε αληθείς και άλλοτε ψευδείς, ανάλογα με τις τιμές των μεταβλητών τους.

Η έννοια της ισοδυναμίας είναι ιδιαίτερα χρήσιμη, αφού επιτρέπει την μετατροπή μιας λογικής έκφρασης σε μια πιο απλή και κατανοητή μορφή. Ο έλεγχος της ισοδυναμίας μπορεί να πραγματοποιηθεί είτε με τη χρήση πινάκων αληθείας είτε με την εφαρμογή βασικών λογικών κανόνων.

2.7 Εφαρμογές της Προτασιακής Λογικής

Η Προτασιακή Λογική δεν περιορίζεται μόνο στην θεωρητική ανάλυση των προτάσεων, αλλά έχει σημαντικές εφαρμογές στην αναπαράσταση και επίλυση πραγματικών προβλημάτων.

Η χρήση της επιτρέπει τη μετατροπή προτάσεων της φυσικής γλώσσας σε λογικές εκφράσεις, μειώνοντας την ασάφεια και διευκολύνοντας τον έλεγχο της ορθότητας μιας διατύπωσης. Λέξεις και φράσεις όπως «ή», «μόνο αν», «εκτός αν» και «αν... τότε» μπορεί να έχουν διαφορετικές σημασίες στη φυσική γλώσσα, ενώ στην Προτασιακή Λογική μπορούν να εκφραστούν με συγκεκριμένους λογικούς τελεστές, ακολουθώντας καθορισμένους λογικούς κανόνες.

2.8 Συναρτησιακή Πληρότητα

Η συναρτησιακή πληρότητα αφορά τη δυνατότητα έκφρασης όλων των λογικών προτάσεων με τη χρήση ενός συγκεκριμένου συνόλου τελεστών. Ένα σύνολο τελεστών ονομάζεται συναρτησιακά πλήρες όταν μπορεί να χρησιμοποιηθεί για την κατασκευή οποιασδήποτε σύνθετης πρότασης της Προτασιακής Λογικής.

Για παράδειγμα, το σύνολο $\{\neg, \wedge, \vee\}$ είναι συναρτησιακά πλήρες, καθώς με τους τελεστές αυτούς μπορούν να εκφραστούν και άλλοι λογικοί τελεστές, όπως η συνεπαγωγή, η διπλή συνεπαγωγή και η αποκλειστική διάζευξη.

Η έννοια αυτή είναι σημαντική, καθώς δείχνει ότι πολύπλοκες λογικές εκφράσεις μπορούν να κατασκευαστούν με τη χρήση ενός περιορισμένου αριθμού βασικών τελεστών.

2.9 Σύνδεση Θεωρίας με την Πλατφόρμα

Το θεωρητικό υπόβαθρο που παρουσιάστηκε στο κεφάλαιο αυτό αποτέλεσε τη βάση για την οργάνωση του εκπαιδευτικού περιεχομένου της πλατφόρμας. Το θεωρητικό υλικό οργανώθηκε σε επιμέρους κεφάλαια και συνδυάστηκε με ασκήσεις και quizzes, ώστε να υποστηρίζεται τόσο η κατανόηση όσο και η πρακτική εξάσκηση των χρηστών.

Η πλατφόρμα επιτρέπει στον χρήστη να μελετά τη θεωρία και στη συνέχεια να εφαρμόζει τις γνώσεις του μέσω διαδραστικών δραστηριοτήτων, ενισχύοντας έτσι τη μαθησιακή διαδικασία.

Κεφάλαιο 3

Ανάλυση Απαιτήσεων Συστήματος

3.1 Περιγραφή του Συστήματος	10
3.2 Κατηγορίες Χρηστών και Ρόλοι	11
3.2.1 Φοιτητής	12
3.2.2 Διαχειριστής	12
3.2.3 Έλεγχος Πρόσβασης	13
3.3 Λειτουργικές Απαιτήσεις	13
3.3.1 Λειτουργικές Απαιτήσεις Φοιτητή	13
3.3.2 Λειτουργικές Απαιτήσεις Διαχειριστή	14
3.3.3 Γενικές Λειτουργικές Απαιτήσεις Συστήματος	15
3.4 Μη Λειτουργικές Απαιτήσεις	15
3.4.1 Ευχρηστία	16
3.4.2 Ασφάλεια	16
3.4.3 Απόδοση	16
3.4.4 Επεκτασιμότητα	16
3.5 Use Cases	17

3.1 Περιγραφή του Συστήματος

Η πλατφόρμα Logic Learning αποτελεί μια διαδικτυακή εκπαιδευτική πλατφόρμα, η οποία αναπτύχθηκε με σκοπό την υποστήριξη της εκμάθησης Λογικής, με έμφαση στην Προτασιακή Λογική. Το σύστημα συνδυάζει θεωρητικό υλικό, διαδραστικές ασκήσεις και quizzes, ώστε ο χρήστης να μπορεί να μελετά την ύλη και στη συνέχεια να εφαρμόζει τις γνώσεις του μέσα από την πρακτική εξάσκηση.

Η εφαρμογή απευθύνεται κυρίως σε φοιτητές που έρχονται πρώτη φορά με έννοιες Λογικής στην Πληροφορική, όπως το μάθημα ΕΠΛ111. Παράλληλα, μπορεί να χρησιμοποιηθεί και από οποιονδήποτε χρήστη επιθυμεί να εξασκηθεί σε βασικές έννοιες της Προτασιακής Λογικής.

Η βασική ιδέα του συστήματος είναι η δημιουργία ενός ενιαίου μαθησιακού περιβάλλοντος, μέσα στο οποίο ο χρήστης δεν περιορίζεται μόνο στην ανάγνωση θεωρίας, αλλά έχει τη δυνατότητα να αλληλεπιδρά με το περιεχόμενο. Για τον σκοπό αυτό, η πλατφόρμα περιλαμβάνει ενότητες με κεφάλαια θεωρίας, ερωτήσεις αξιολόγησης μέσω quizzes και ασκήσεις εξάσκησης διαφορετικών τύπων.

Η εφαρμογή είναι προσβάσιμη μέσω web browser και βασίζεται σε αρχιτεκτονική client-server. Το περιεχόμενο οργανώνεται σε ενότητες με κεφάλαια, ώστε να υποστηρίζεται μια δομημένη μαθησιακή πορεία, ενώ οι ασκήσεις και τα quizzes λειτουργούν συμπληρωματικά προς τη θεωρία.

Το σύστημα περιλαμβάνει τις ακόλουθες λειτουργικές ενότητες:

- Θεωρία: Παρέχει οργανωμένο εκπαιδευτικό υλικό, το οποίο περιλαμβάνει ορισμούς, παραδείγματα και βασικές έννοιες της Προτασιακής Λογικής
- Ασκήσεις: Παρέχουν στον χρήστη τη δυνατότητα πρακτικής εφαρμογής της θεωρίας μέσα από διαφορετικού τύπου ασκήσεις. Η απάντηση του αξιολογείται ως σωστή ή λανθασμένη, χωρίς να υπολογίζεται βαθμός, καθώς ο βασικός στόχος των ασκήσεων είναι καθαρά η εξάσκηση και η κατανόηση του περιεχομένου.
- Quizzes: Χρησιμοποιούνται για την αξιολόγηση του χρήστη. Περιλαμβάνουν ερωτήσεις πολλαπλής επιλογής με μία ή περισσότερες σωστές απαντήσεις και υποστηρίζουν αυτόματη βαθμολόγηση. Στις ερωτήσεις με πολλαπλές σωστές απαντήσεις εφαρμόζεται σύστημα ποινής για λανθασμένες επιλογές, χωρίς όμως η τελική βαθμολογία να γίνεται αρνητική.
- Διαχείριση περιεχομένου: Παρέχει δυνατότητες δημιουργίας, επεξεργασίας και διαγραφής θεωρίας, ασκήσεων και quizzes από εξουσιοδοτημένους χρήστες (admin).
- Διαχείριση χρηστών και προόδου: Υποστηρίζει τη διαγραφή χρηστών, καθώς και την παρακολούθηση των αποτελεσμάτων στα quizzes που έκαναν οι χρήστες.

Συνολικά, το σύστημα λειτουργεί ως ένα υποστηρικτικό εργαλείο μάθησης, το οποίο συνδυάζει θεωρία και πράξη, προσφέροντας άμεση ανατροφοδότηση στον χρήστη.

3.2 Κατηγορίες Χρηστών και Ρόλοι

Το σύστημα υποστηρίζει δύο βασικές κατηγορίες χρηστών: τον φοιτητή και τον διαχειριστή. Οι κατηγορίες αυτές διαφοροποιούνται ως προς τα δικαιώματα πρόσβασης και τις λειτουργίες που μπορούν να εκτελέσουν μέσα στην πλατφόρμα. Η διάκριση αυτή είναι απαραίτητη, καθώς επιτρέπει τον διαχωρισμό ανάμεσα στη χρήση του εκπαιδευτικού περιεχομένου και στη διαχείρισή του. Κάθε νέος χρήστης που εγγράφεται στο σύστημα λαμβάνει αυτόματα τον ρόλο του φοιτητή, ενώ ο ρόλος του διαχειριστή αποδίδεται ξεχωριστά.

3.2.1 Φοιτητής

Ο φοιτητής αποτελεί τον κύριο χρήστη της πλατφόρμας και έχει πρόσβαση στις λειτουργίες μάθησης. Συγκεκριμένα, μπορεί να:

- Εγγραφεί και να συνδέεται στο σύστημα
- Προβάλλει το θεωρητικό περιεχόμενο
- Επιλύει ασκήσεις
- Συμμετέχει σε quizzes
- Προβάλλει τα αποτελέσματα των quizzes

Ο φοιτητής δεν έχει πρόσβαση σε οποιαδήποτε λειτουργία διαχείρισης. Έτσι ο περιορισμός αυτός διασφαλίζει ότι το περιεχόμενο της πλατφόρμας δεν τροποποιείται από μη εξουσιοδοτημένους χρήστες.

3.2.2 Διαχειριστής

Ο διαχειριστής είναι υπεύθυνος για τη διαχείριση της πλατφόρμας και του εκπαιδευτικού περιεχομένου. Έχει πρόσβαση σε ειδικό περιβάλλον διαχείρισης, μέσω του οποίου μπορεί να:

- Δημιουργεί, να επεξεργάζεται και να διαγράφει ενότητες και κεφάλαια θεωρίας
- Δημιουργεί και να επεξεργάζεται ασκήσεις
- Δημιουργεί και να επεξεργάζεται quizzes με ερωτήσεις
- Βλέπει και να διαχειρίζεται τους χρήστες
- Παρακολουθεί τα αποτελέσματα των χρηστών και όπου επιθυμεί να επαναφέρει τα αποτελέσματα τους
- Προβάλλει την πλατφόρμα ως χρήστης: Με αυτόν τον τρόπο μπορεί να ελέγχει πώς εμφανίζεται η πλατφόρμα από την πλευρά του τελικού χρήστη και να δοκιμάζει λειτουργίες όπως η προβολή θεωρίας, η επίλυση ασκήσεων και η συμμετοχή σε quizzes.

Ο ρόλος του διαχειριστή καθιστά την πλατφόρμα πιο ευέλικτη, καθώς το εκπαιδευτικό περιεχόμενο μπορεί να ενημερώνεται και να οργανώνεται μέσα από το ίδιο το σύστημα. Με αυτόν τον τρόπο, η πλατφόρμα μπορεί να προσαρμόζεται πιο εύκολα σε νέες ανάγκες ή αλλαγές του μαθήματος.

3.2.3 Έλεγχος Πρόσβασης

Η πρόσβαση στις λειτουργίες του συστήματος καθορίζεται από τον ρόλο του χρήστη. Μετά τη σύνδεση, το σύστημα αναγνωρίζει τον ρόλο και επιτρέπει την πρόσβαση μόνο στις αντίστοιχες λειτουργίες.

Οι φοιτητές έχουν πρόσβαση αποκλειστικά στις λειτουργίες μάθησης, ενώ οι διαχειριστές έχουν επιπλέον δικαιώματα διαχείρισης. Με τον τρόπο αυτό διασφαλίζεται η ασφάλεια του συστήματος και η ορθή διαχείριση του περιεχομένου.

3.3 Λειτουργικές Απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τις βασικές λειτουργίες που πρέπει να υποστηρίζει το σύστημα, καθώς και τον τρόπο με τον οποίο οι χρήστες αλληλεπιδρούν με αυτό. Το σύστημα υποστηρίζει δύο διακριτούς ρόλους χρηστών, τον φοιτητή και τον διαχειριστή, οι οποίοι διαθέτουν διαφορετικά δικαιώματα και δυνατότητες.

Οι απαιτήσεις που παρουσιάζονται στη συνέχεια διαχωρίζονται ανάλογα με τον ρόλο του χρήστη, καθώς και σε γενικές λειτουργικές απαιτήσεις που αφορούν τη συνολική λειτουργία του συστήματος

3.3.1 Λειτουργικές απαιτήσεις Φοιτητή

Το σύστημα πρέπει να υποστηρίζει τις ακόλουθες λειτουργίες για τον φοιτητή:

1. Εγγραφή χρήστη: Ο χρήστης πρέπει να μπορεί να δημιουργήσει νέο λογαριασμό, εισάγοντας βασικά στοιχεία δηλαδή ονοματεπώνυμο, email και κωδικό.
2. Σύνδεση χρήστη: Ο χρήστης πρέπει να μπορεί να συνδεθεί στο σύστημα χρησιμοποιώντας τα στοιχεία του.

3. Αποσύνδεση χρήστη: Ο χρήστης πρέπει να μπορεί να αποσυνδεθεί από το σύστημα.
4. Προβολή θεωρίας: Ο χρήστης πρέπει να έχει πρόσβαση στο θεωρητικό περιεχόμενο της πλατφόρμας, το οποίο είναι οργανωμένο σε ενότητες και κεφάλαια.
5. Πλοήγηση στο περιεχόμενο: Ο χρήστης πρέπει να μπορεί να μεταβαίνει εύκολα μεταξύ διαφορετικών εννοιών, κεφαλαίων, ασκήσεων και quizzes.
6. Επίλυση ασκήσεων: Ο χρήστης πρέπει να μπορεί να επιλέγει και να επιλύει ασκήσεις. Το σύστημα ελέγχει την απάντηση και ενημερώνει αν είναι σωστή ή λανθασμένη. Οι ασκήσεις χρησιμοποιούνται για εξάσκηση και δεν συνοδεύονται από βαθμολογία, αλλά από ένδειξη ορθότητας.
7. Συμμετοχή σε quizzes: Ο χρήστης πρέπει να μπορεί να συμμετέχει σε quizzes και να απαντά σε ερωτήσεις πολλαπλής επιλογής με μια ή περισσότερες ορθές απαντήσεις.
8. Αυτόματη αξιολόγηση quizzes: Το σύστημα πρέπει να υπολογίζει αυτόματα τη βαθμολογία του χρήστη. Υποστηρίζονται ερωτήσεις με μία σωστή απάντηση, καθώς και ερωτήσεις με περισσότερες από μία σωστές απαντήσεις, όπου εφαρμόζεται σύστημα ποινής για λανθασμένες επιλογές.
9. Προβολή εξήγησης τρόπου βαθμολόγησης: Ο χρήστης να μπορεί με το πάτημα του κουμπιού να βλέπει τον τρόπο με τον οποίο υπολογίζεται ο βαθμός της κάθε ερώτησης και του συνολικού quiz στο τέλος.
10. Προβολή αποτελεσμάτων: Μετά την ολοκλήρωση ενός quiz, ο χρήστης πρέπει να μπορεί να δει τη βαθμολογία του καθώς και τις σωστές απαντήσεις. Σε περίπτωση που επιστρέψει στο ίδιο quiz, δεν θα έχει τη δυνατότητα να το λύσει ξανά, αλλά μόνο να δει τα αποτελέσματα της ήδη ολοκληρωμένης προσπάθειας του.

3.3.2 Λειτουργικές Απαιτήσεις Διαχειριστή

Το σύστημα πρέπει να υποστηρίζει τις ακόλουθες λειτουργίες για τον διαχειριστή:

1. Σύνδεση χρήστη: Ο χρήστης πρέπει να μπορεί να συνδεθεί στο σύστημα χρησιμοποιώντας τα στοιχεία του.
2. Διαχείριση θεωρίας: Ο διαχειριστής πρέπει να μπορεί να δημιουργεί, να επεξεργάζεται και να διαγράφει ενότητες και κεφάλαια θεωρητικού περιεχομένου.
3. Διαχείριση ασκήσεων: Ο διαχειριστής πρέπει να μπορεί να δημιουργεί, να επεξεργάζεται και να διαγράφει ασκήσεις. Για κάθε άσκηση πρέπει να μπορεί να ορίζει τον τύπο της, τις διαθέσιμες επιλογές απάντησης, τη σωστή απάντηση και την εξήγηση.
4. Διαχείριση quizzes: Ο διαχειριστής πρέπει να μπορεί να δημιουργεί, να επεξεργάζεται και να διαγράφει quizzes, καθώς και να προσθέτει ή να επεξεργάζεται τις ερωτήσεις

τους. Για κάθε ερώτηση πρέπει να μπορεί να ορίζει τον τύπο της, τις διαθέσιμες επιλογές απάντησης, τη σωστή απάντηση και την εξήγηση.

5. Διαχείριση χρηστών: Ο διαχειριστής πρέπει να μπορεί να παρακολουθεί ποιοι είναι οι χρήστες του συστήματος και βασικές πληροφορίες που σχετίζονται με αυτούς.
6. Παρακολούθηση αποτελεσμάτων: Ο διαχειριστής πρέπει να έχει τη δυνατότητα να βλέπει την επίδοση των χρηστών στα quizzes.
7. Επαναφορά προόδου: Το σύστημα πρέπει να παρέχει στον διαχειριστή τη δυνατότητα επαναφοράς της προόδου ενός χρήστη σε συγκεκριμένο quiz. Έτσι διαγράφονται τα προηγούμενα αποτελέσματα του χρήστη για το συγκεκριμένο quiz, ώστε να μπορεί να το λύσει ξανά από την αρχή.
8. Προβολή ως χρήστης: Ο διαχειριστής πρέπει να μπορεί να μεταβαίνει στο περιβάλλον του φοιτητή, ώστε να ελέγχει την εμπειρία χρήσης και τη σωστή εμφάνιση του περιεχομένου.

3.3.3 Γενικές Λειτουργικές Απαιτήσεις Συστήματος

Το σύστημα πρέπει να υποστηρίζει τις ακόλουθες λειτουργίες γενικά για το σύστημα:

1. Έλεγχος πρόσβασης: Το σύστημα πρέπει να διαχωρίζει τις λειτουργίες ανάλογα με τον ρόλο του χρήστη και να του επιτρέπει πρόσβαση μόνο σε εξουσιοδοτημένες ενέργειες.
2. Αποθήκευση δεδομένων: Τα δεδομένα χρηστών, θεωρητικού περιεχομένου, ασκήσεων, quizzes και αποτελεσμάτων πρέπει να αποθηκεύονται σε βάση δεδομένων.
3. Ακεραιότητα δεδομένων: Το σύστημα πρέπει να διασφαλίζει ότι οι ενέργειες των χρηστών εκτελούνται με συνέπεια και χωρίς σφάλματα.
4. Αλληλεπίδραση χρήστη-συστήματος: Το σύστημα πρέπει να παρέχει άμεση και κατανοητή ανατροφοδότηση στον χρήστη σχετικά με τις ενέργειες του. Η ανατροφοδότηση μπορεί να αφορά το αποτέλεσμα μιας άσκησης ή ενός quiz, την εμφάνιση μηνυμάτων σφάλματος σε περίπτωση μη έγκυρης ενέργειας, καθώς και την εμφάνιση προειδοποιητικών ή ενημερωτικών μηνυμάτων όταν απαιτείται η προσοχή του χρήστη.

3.4 Μη Λειτουργικές Απαιτήσεις

Οι μη λειτουργικές απαιτήσεις περιγράφουν τα ποιοτικά χαρακτηριστικά του συστήματος και καθορίζουν τον τρόπο με τον οποίο αυτό πρέπει να λειτουργεί, χωρίς να σχετίζονται άμεσα με

συγκεκριμένες λειτουργίες. Οι απαιτήσεις αυτές αφορούν ζητήματα όπως η ευχρηστία, η ασφάλεια, η απόδοση και η επεκτασιμότητα.

3.4.1 Ευχρηστία

Το σύστημα πρέπει να είναι εύχρηστο και κατανοητό από τους χρήστες, ιδιαίτερα από φοιτητές που βρίσκονται στα πρώτα στάδια των σπουδών τους.

- Η διεπαφή του συστήματος πρέπει να είναι απλή και σαφής.
- Η πλοήγηση μεταξύ των διαφόρων σελίδων πρέπει να είναι άμεση και εύκολη.
- Οι βασικές λειτουργίες πρέπει να είναι προσβάσιμες μέσω ξεκάθαρα μενού πλοήγησης.
- Το σύστημα πρέπει να παρέχει άμεση ανατροφοδότηση για τις ενέργειες του χρήστη.

3.4.2 Ασφάλεια

Η προστασία δεδομένων των χρηστών είναι πολύ σημαντική για ένα σύστημα.

- Οι κωδικοί πρόσβασης των χρηστών πρέπει να αποθηκεύονται με hashing, ώστε να μην μπορεί να τους δει ο διαχειριστής της βάσης δεδομένων.
- Η διαδικασία σύνδεσης πρέπει να ελέγχει την εγκυρότητα των στοιχείων του χρήστη.
- Το σύστημα πρέπει να χρησιμοποιεί μηχανισμό διαχείρισης sessions για την πιστοποίηση των χρηστών.
- Πρέπει να εφαρμόζεται έλεγχος πρόσβασης ώστε κάθε χρήστης να έχει πρόσβαση μόνο στις λειτουργίες που αντιστοιχούν στον ρόλο του.
- Η επικοινωνία με τη βάση δεδομένων πρέπει να γίνεται με ασφαλή τρόπο για να αποφευχθούν επιθέσεις όπως SQL injection.

3.4.3 Απόδοση

Η πλατφόρμα πρέπει να ανταποκρίνεται άμεσα στις ενέργειες των χρηστών.

- Οι σελίδες πρέπει να φορτώνονται σε σύντομο χρονικό διάστημα.
- Η ανάκτηση δεδομένων από τη βάση πρέπει να γίνεται γρήγορα.

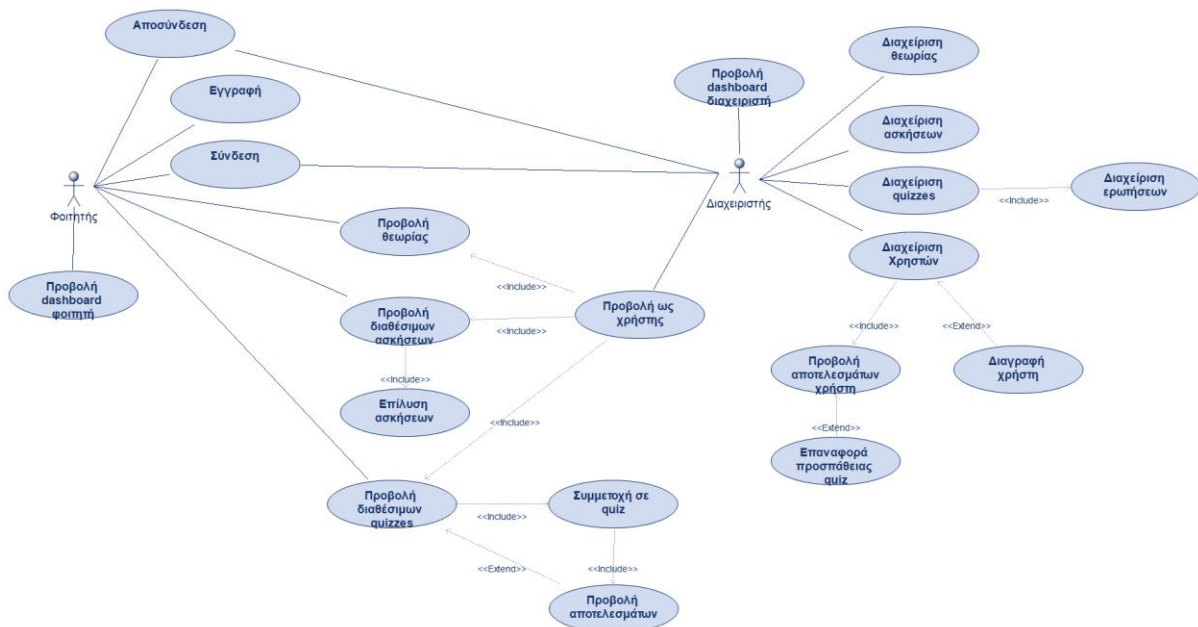
3.4.4 Επεκτασιμότητα

Το σύστημα πρέπει να είναι σχεδιασμένο με τρόπο που να μπορεί να επεκταθεί και μελλοντικά.

- Η αρχιτεκτονική του πρέπει να επιτρέπει την προσθήκη νέων λειτουργιών χωρίς μεγάλες αλλαγές στον κώδικα.
- Ο διαχωρισμός της λογικής του συστήματος (controllers, repositories) διευκολύνει τη συντήρηση και επέκταση του συστήματος.

3.5 Use Cases

Οι περιπτώσεις χρήσης περιγράφουν τις βασικές λειτουργίες που μπορούν να κάνουν οι χρήστες στο σύστημα. Εδώ έχουμε δύο είδη χρηστών: τον φοιτητή και τον διαχειριστή. Κάθε είδος χρήστη έχει πρόσβαση σε διαφορετικές λειτουργίες, ανάλογα με τον ρόλο του.



Εικόνα 1 Use Case Diagram για τον φοιτητή και τον διαχειριστή [8]

Στην Εικόνα 1 φαίνεται το διάγραμμα περιπτώσεων χρήσης για τον φοιτητή και τον διαχειριστή.

Ο φοιτητής μπορεί να εγγραφεί, να συνδεθεί, να μεταβεί στο προσωπικό του dashboard, να προβάλλει τη θεωρία, να δει τις διαθέσιμες ασκήσεις, να επιλύσει ασκήσεις, να δει τα διαθέσιμα quizzes, να συμμετάσχει σε quiz και να βλέπει τα αποτελέσματά του. Μετά την ολοκλήρωση ενός quiz, ο φοιτητής μπορεί να δει τα αποτελέσματα και την επίδοσή του.

Ο διαχειριστής μπορεί να συνδεθεί στο σύστημα, να μεταβεί στο dashboard διαχειριστή, να διαχειριστεί τη θεωρία, τις ασκήσεις, τα quizzes, τις ερωτήσεις και τους χρήστες. Στο πλαίσιο της διαχείρισης quizzes περιλαμβάνεται και η διαχείριση ερωτήσεων, καθώς οι ερωτήσεις αποτελούν μέρος ενός quiz. Επιπλέον, μέσα από τη διαχείριση χρηστών, ο διαχειριστής μπορεί να διαγράψει ένα χρήστη και να δει τα αποτελέσματα των χρηστών και όπου απαιτείται να επαναφέρει την προσπάθεια ενός quiz.

Κεφάλαιο 4

Σχεδίαση Συστήματος

4.1 Αρχιτεκτονική Συστήματος MVC	19
4.2 Δομή Βάσης Δεδομένων	21
4.2.1 Πίνακες Βάσης Δεδομένων	21
4.2.2 ER Διάγραμμα	27
4.3 Σχεδίαση Διεπαφής Χρήστη (UX/UI)	28
4.3.1 Σχεδιαστικές Αρχές Διεπαφής	28
4.3.2 Διεπαφή Φοιτητή	29
4.3.2.1 Dashboard Φοιτητή	30
4.3.2.2 Σελίδα Θεωρίας	30
4.3.2.3 Σελίδα Ασκήσεων	33
4.3.2.4 Σελίδα Quizzes	38
4.3.3 Διεπαφή Διαχειριστή	39
4.3.3.1 Dashboard Διαχειριστή	39
4.3.3.2 Σελίδα Διαχείρισης Θεωρίας	40
4.3.3.3 Σελίδα Διαχείρισης Ασκήσεων	43
4.3.3.4 Σελίδα Διαχείρισης Quiz	50
4.3.3.5 Σελίδα Διαχείρισης Χρηστών	53
4.3.3.6 Σελίδα Προβολής ως Χρήστης	55
4.3.4 Πλοήγηση και Εμπειρία Χρήστη	56
4.4 Σχεδίαση Λειτουργιών Ασκήσεων	59
4.5 Σχεδίαση Λειτουργιών Quiz	60

4.1 Αρχιτεκτονική Συστήματος MVC

Η πλατφόρμα αναπτύχθηκε ως διαδικτυακή εφαρμογή τύπου client-server στην οποία ο χρήστης client αλληλεπιδρά με το frontend μέσω browser, ενώ η επεξεργασία των δεδομένων και η επικοινωνία με τη βάση πραγματοποιούνται από την πλευρά του server στο backend. [4]

Η αρχιτεκτονική της εφαρμογής βασίζεται στη λογική του προτύπου MVC (Model-View-Controller), ώστε να διαχωρίζονται οι επιμέρους ευθύνες του συστήματος δηλαδή η διεπαφή χρήστη, η λογική της εφαρμογής και η πρόσβαση στη βάση δεδομένων. Ο διαχωρισμός αυτός βοηθά στην καλύτερη οργάνωση του κώδικα, στη συντηρησιμότητα της εφαρμογής και για να είναι δυνατή η μελλοντική του επέκταση.

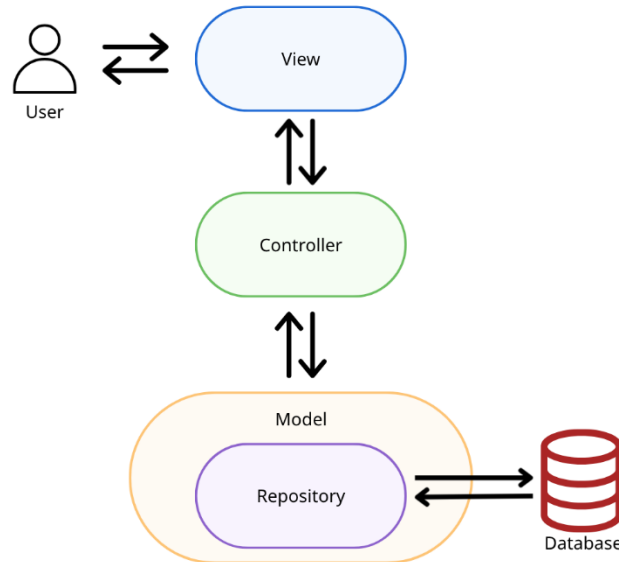
Το επίπεδο View περιλαμβάνει τις σελίδες που εμφανίζονται στον χρήστη, καθώς και τα αρχεία CSS και JavaScript που διαμορφώνουν την εμφάνιση και τη διαδραστικότητα της πλατφόρμας. Σε αυτό το επίπεδο ανήκουν οι σελίδες σύνδεσης και εγγραφής, οι σελίδες φοιτητή για προβολή θεωρίας, επίλυση ασκήσεων και συμμετοχή σε quizzes, καθώς και οι σελίδες διαχειριστή για τη διαχείριση θεωρίας, ασκήσεων, quizzes και χρηστών.

Το επίπεδο Controller περιλαμβάνει τις κλάσεις και τα αρχεία που διαχειρίζονται τη ροή της εφαρμογής. Οι Controllers λαμβάνουν τα δεδομένα που υποβάλλονται από τον χρήστη, εφαρμόζουν τη βασική λογική κάθε λειτουργίας και καλούν το κατάλληλο repository για την ανάκτηση ή αποθήκευση δεδομένων. Στην πλατφόρμα χρησιμοποιούνται controllers όπως οι TheoryController, QuizController και ExerciseController, οι οποίοι αναλαμβάνουν τη λογική που σχετίζεται με τη θεωρία, τα quizzes και τις ασκήσεις αντίστοιχα.

Το επίπεδο Model υλοποιείται κυρίως μέσω των repositories και της βάσης δεδομένων. Η πρόσβαση και η ανάκτηση των δεδομένων γίνεται μέσα από repository classes, όπως UserRepository, TheoryRepository, QuizRepository, ExerciseRepository, UserQuizRepository, UserExerciseRepository και UserTheoryRepository. Τα repositories συγκεντρώνουν τα ερωτήματα προς τη βάση δεδομένων και επιστρέφουν τα απαραίτητα δεδομένα στους controllers ή στις σελίδες που τα χρειάζονται.

Η χρήση repositories έχει σημαντικά πλεονεκτήματα. Αρχικά, αποτρέπει την άμεση ενσωμάτωση SQL ερωτημάτων στις σελίδες παρουσίασης ή στους controllers. Επιπλέον, συγκεντρώνει τη λογική πρόσβασης στα δεδομένα σε συγκεκριμένα αρχεία, διευκολύνοντας τη συντήρηση και την επέκταση του συστήματος. Για παράδειγμα, αν χρειαστεί να αλλάξει ο τρόπος ανάκτησης των quizzes ή των ασκήσεων, η αλλαγή μπορεί να γίνει στο αντίστοιχο repository χωρίς να επηρεαστούν οι σελίδες της διεπαφής χρήστη.

Η γενική ροή λειτουργίας της αρχιτεκτονικής είναι η εξής: ο χρήστης αλληλεπιδρά με τη διεπαφή της εφαρμογής, η ενέργεια μεταφέρεται στον αντίστοιχο controller, ο controller καλεί το κατάλληλο repository, το repository επικοινωνεί με τη βάση δεδομένων και επιστρέφει τα δεδομένα, τα οποία στη συνέχεια εμφανίζονται ξανά στον χρήστη μέσω της αντίστοιχης σελίδας.



Εικόνα 2 Αρχιτεκτονική MVC με Repository Pattern

4.2 Δομή Βάσης Δεδομένων

Η βάση δεδομένων οργανώθηκε σε διακριτούς πίνακες, ώστε κάθε κατηγορία πληροφορίας να αποθηκεύεται ξεχωριστά. Με αυτόν τον τρόπο περιορίζεται η επανάληψη δεδομένων, διευκολύνεται η διαχείριση τους και διατηρούνται πιο καθαρές οι σχέσεις μεταξύ των βασικών οντοτήτων της πλατφόρμας, όπως οι χρήστες, το θεωρητικό περιεχόμενο, οι ασκήσεις, τα quizzes, οι ερωτήσεις, οι επιλογές απαντήσεων και οι προσπάθειες των χρηστών.

4.2.1 Πίνακες Βάσης Δεδομένων

Πιο κάτω παρουσιάζεται η επεξήγηση των βασικών πινάκων της βάσης δεδομένων που προέκυψαν από το διάγραμμα οντοτήτων-συσχετίσεων.

Πίνακας users

Ο πίνακας users αποθηκεύει τα στοιχεία των χρηστών της πλατφόρμας. Το πεδίο id αποτελεί το primary key του πίνακα και χρησιμοποιείται για τη μοναδική αναγνώριση κάθε χρήστη. Τα πεδία fullname, email και password αποθηκεύουν το ονοματεπώνυμο, το email και τον κωδικό

πρόσβασης του χρήστη. Το πεδίο `role` καθορίζει τον ρόλο του χρήστη στο σύστημα, δηλαδή αν είναι απλός χρήστης/φοιτητής ή διαχειριστής. Τα πεδία `is_deleted` και `deleted_at` χρησιμοποιούνται για το soft delete του χρήστη, ενώ τα `created_at` και `last_login` καταγράφουν την ημερομηνία δημιουργίας του λογαριασμού και την τελευταία σύνδεση.

#	Name	Type
1	id 🔑	int(11)
2	fullname	varchar(255)
3	email 🔑	varchar(255)
4	password	varchar(255)
5	role	enum('user', 'admin')
6	is_deleted 🔑	tinyint(1)
7	deleted_at 🔑	datetime
8	created_at 🔑	datetime
9	last_login	datetime

Πίνακας `theory_units`

Ο πίνακας `theory_units` αποθηκεύει τις βασικές ενότητες θεωρίας της πλατφόρμας. Το πεδίο `id` αποτελεί το primary key, ενώ το `title` αποθηκεύει τον τίτλο της ενότητας. Το πεδίο `description` χρησιμοποιείται για σύντομη περιγραφή της ενότητας. Τα πεδία `created_at` και `updated_at` καταγράφουν την ημερομηνία δημιουργίας και τελευταίας ενημέρωσης.

#	Name	Type
1	id 🔑	int(11)
2	unit_id 🔑	int(11)
3	title	varchar(255)
4	content	longtext
5	order_index	int(11)
6	created_at	timestamp
7	updated_at	timestamp
8	is_active	tinyint(1)

Πίνακας `theory_chapters`

Ο πίνακας `theory_chapters` αποθηκεύει τα κεφάλαια θεωρίας που ανήκουν σε κάθε ενότητα. Το πεδίο `id` είναι το primary key, ενώ το `unit_id` συνδέει το κεφάλαιο με την αντίστοιχη

ενότητα θεωρίας. Το πεδίο title αποθηκεύει τον τίτλο του κεφαλαίου και το content περιέχει το πλήρες θεωρητικό περιεχόμενο. Το order_index χρησιμοποιείται για τον καθορισμό της σειράς εμφάνισης των κεφαλαίων. Τα πεδία created_at, updated_at και is_active χρησιμοποιούνται για τη χρονική παρακολούθηση και την κατάσταση εμφάνισης του κεφαλαίου.

#	Name	Type
1	id 	int(11)
2	unit_id 	int(11)
3	title	varchar(255)
4	content	longtext
5	order_index	int(11)
6	created_at	timestamp
7	updated_at	timestamp
8	is_active	tinyint(1)

Πίνακας quizzes

Ο πίνακας quizzes αποθηκεύει τα quizzes της πλατφόρμας. Το πεδίο id αποτελεί το primary key και το chapter_id μπορεί να συνδέει το quiz με συγκεκριμένο κεφάλαιο θεωρίας. Το title και το description αποθηκεύουν τον τίτλο και την περιγραφή του quiz. Το πεδίο difficulty_level δηλώνει το επίπεδο δυσκολίας, ενώ το is_active καθορίζει αν το quiz είναι διαθέσιμο στους χρήστες. Τα πεδία created_at και updated_at καταγράφουν τη δημιουργία και την τελευταία ενημέρωση του quiz.

#	Name	Type
1	id 	int(11)
2	chapter_id 	int(11)
3	title	varchar(255)
4	created_at	datetime
5	updated_at	datetime
6	description	text
7	difficulty_level	varchar(20)
8	is_active	tinyint(1)

Πίνακας questions

Ο πίνακας questions αποθηκεύει τις ερωτήσεις που ανήκουν σε κάθε quiz. Το πεδίο id αποτελεί το primary key και το quiz_id συνδέει την ερώτηση με το αντίστοιχο quiz. Το πεδίο text περιέχει το κείμενο της ερώτησης, ενώ το explanation χρησιμοποιείται για την επεξήγηση της απάντησης. Το πεδίο question_type καθορίζει τον τύπο της ερώτησης, όπως ερώτηση με μία σωστή απάντηση ή ερώτηση με πολλαπλές σωστές απαντήσεις. Το order_index καθορίζει τη σειρά εμφάνισης της ερώτησης μέσα στο quiz.

#	Name	Type
1	id 	int(11)
2	quiz_id 	int(11)
3	text	text
4	explanation	text
5	type	varchar(20)
6	question_type	varchar(20)
7	order_index	int(11)
8	created_at	datetime
9	updated_at	datetime

Πίνακας choices

Ο πίνακας choices αποθηκεύει τις επιλογές απάντησης για κάθε ερώτηση. Το πεδίο id αποτελεί το primary key και το question_id συνδέει την επιλογή με την αντίστοιχη ερώτηση. Το πεδίο text περιέχει το κείμενο της επιλογής, ενώ το is_correct δηλώνει αν η επιλογή είναι σωστή. Το πεδίο explanation μπορεί να χρησιμοποιείται για επιπλέον επεξήγηση σχετικά με την επιλογή.

#	Name	Type
1	id 	int(11)
2	question_id 	int(11)
3	text	varchar(1024)
4	is_correct	tinyint(1)
5	explanation	text

Πίνακας quiz_attempts

Ο πίνακας quiz_attempts αποθηκεύει τις προσπάθειες των χρηστών στα quizzes. Το πεδίο id αποτελεί το primary key, ενώ τα user_id και quiz_id συνδέουν την προσπάθεια με τον χρήστη και το quiz αντίστοιχα. Το πεδίο score αποθηκεύει τη βαθμολογία του χρήστη και το

max_score τη μέγιστη δυνατή βαθμολογία (συνήθως 100). Τα πεδία attempted_at, started_at και finished_at χρησιμοποιούνται για την καταγραφή του χρόνου προσπάθειας του quiz.

#	Name	Type
1	id 	int(11)
2	user_id 	int(11)
3	quiz_id 	int(11)
4	score	decimal(5,2)
5	max_score	int(11)
6	attempted_at	datetime
7	started_at	datetime
8	finished_at	datetime

Πίνακας answers

Ο πίνακας answers αποθηκεύει τις απαντήσεις που δόθηκαν σε κάθε προσπάθεια quiz. Το πεδίο id είναι το primary key. Το attempt_id συνδέει την απάντηση με συγκεκριμένη προσπάθεια quiz, το question_id με την αντίστοιχη ερώτηση και το choice_id με την επιλογή που επέλεξε ο χρήστης. Το πεδίο created_at καταγράφει τον χρόνο καταχώρησης της απάντησης.

#	Name	Type
1	id 	int(11)
2	attempt_id 	int(11)
3	question_id 	int(11)
4	choice_id 	int(11)
5	created_at	datetime

Πίνακας exercises

Ο πίνακας exercises αποθηκεύει τις διαδραστικές ασκήσεις της πλατφόρμας. Το πεδίο id αποτελεί το primary key και το title αποθηκεύει τον τίτλο της άσκησης. Το πεδίο type καθορίζει τον τύπο της άσκησης, όπως drag_drop, logical_equivalence, logic_rules ή proof_check. Το difficulty δηλώνει το επίπεδο δυσκολίας. Τα πεδία content και correct_answer αποθηκεύουν το περιεχόμενο και τη σωστή απάντηση της άσκησης σε μορφή JSON/longtext, ώστε να υποστηρίζονται διαφορετικές δομές ασκήσεων. Τα πεδία explanation και instructions χρησιμοποιούνται για επεξήγηση και οδηγίες προς τον χρήστη.

#	Name	Type
1	id 🔑	int(11)
2	title	varchar(255)
3	type 🔑	enum('drag_drop', 'logical_equivalence', 'logic_ru...')
4	difficulty 🔑	enum('easy', 'medium', 'hard')
5	content	longtext
6	correct_answer	longtext
7	explanation	text
8	instructions	text
9	created_at	datetime
10	updated_at	datetime

Πίνακας exercise_attempts

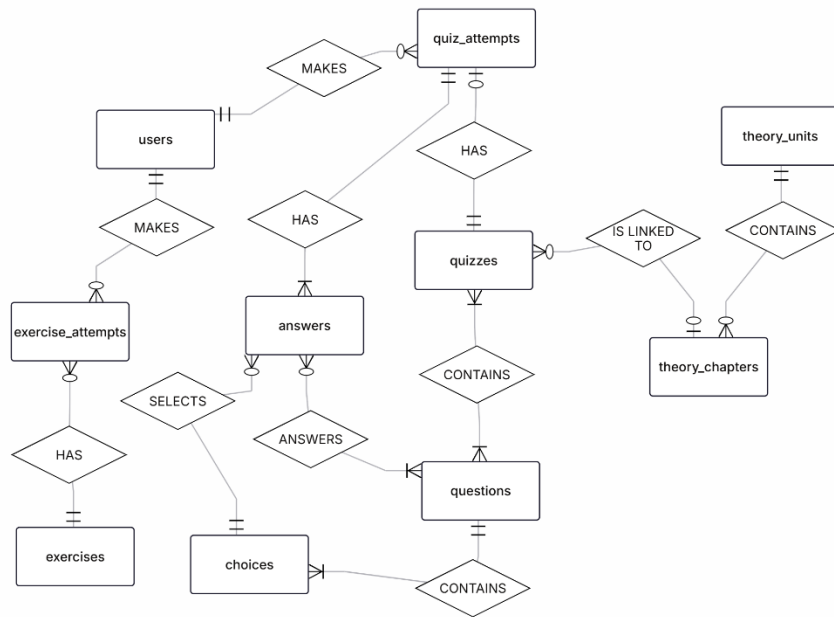
Ο πίνακας exercise_attempts αποθηκεύει τις προσπάθειες των χρηστών στις ασκήσεις. Το πεδίο id αποτελεί το primary key, ενώ τα user_id και exercise_id συνδέουν την προσπάθεια με τον χρήστη και την άσκηση αντίστοιχα. Το πεδίο answer αποθηκεύει την απάντηση που έδωσε ο χρήστης. Το is_correct δηλώνει αν η απάντηση ήταν σωστή ή λανθασμένη, ενώ το feedback αποθηκεύει την ανατροφοδότηση που εμφανίζεται στον χρήστη. Το πεδίο attempted_at καταγράφει την ημερομηνία και ώρα της προσπάθειας.

#	Name	Type
1	id 🔑	int(11)
2	user_id 🔑	int(11)
3	exercise_id 🔑	int(11)
4	answer	longtext
5	is_correct	tinyint(1)
6	feedback	text
7	attempted_at	datetime

4.2.2 ER Διάγραμμα

Στην Εικόνα 3 παρουσιάζεται το διάγραμμα οντοτήτων–συσχετίσεων της βάσης δεδομένων της πλατφόρμας. Το διάγραμμα αποτυπώνει τις βασικές οντότητες του συστήματος και τις

μεταξύ τους σχέσεις, με σκοπό να παρουσιαστεί συνοπτικά ο τρόπος με τον οποίο οργανώνονται τα δεδομένα της εφαρμογής. [5]



Εικόνα 3: Διάγραμμα οντοτήτων–συσχετίσεων της βάσης δεδομένων [9]

Στο διάγραμμα παρουσιάζονται οι οντότητες της βάσης, όπως οι χρήστες, οι ενότητες και τα κεφάλαια θεωρίας, τα quizzes, οι ερωτήσεις, οι επιλογές απαντήσεων, οι ασκήσεις και οι προσπάθειες των χρηστών. Η αναλυτική περιγραφή των πεδίων κάθε πίνακα έχει προηγηθεί στο υποκεφάλαιο 4.2.1, επομένως στο ER διάγραμμα δίνεται έμφαση κυρίως στις σχέσεις μεταξύ των οντοτήτων.

Όπως φαίνεται στο διάγραμμα, ένας χρήστης μπορεί να πραγματοποιήσει πολλαπλές προσπάθειες τόσο σε ασκήσεις όσο και σε quizzes. Κάθε προσπάθεια quiz συνδέεται με έναν συγκεκριμένο χρήστη και ένα συγκεκριμένο quiz, ενώ περιλαμβάνει τις απαντήσεις που καταχωρεί ο χρήστης κατά την επίλυσή του. Κάθε απάντηση συνδέεται με μία ερώτηση και με την επιλογή που επέλεξε ο χρήστης, επιτρέποντας στο σύστημα να αποθηκεύει και να αξιολογεί τις απαντήσεις κάθε προσπάθειας.

Η δομή των quizzes ακολουθεί ιεραρχική οργάνωση. Κάθε quiz περιέχει πολλές ερωτήσεις, ενώ κάθε ερώτηση μπορεί να περιέχει πολλές διαθέσιμες επιλογές απάντησης. Με αυτόν τον τρόπο υποστηρίζεται η δημιουργία quizzes πολλαπλής επιλογής, καθώς και η αποθήκευση των απαντήσεων των χρηστών.

Αντίστοιχα, το θεωρητικό περιεχόμενο οργανώνεται σε ενότητες και κεφάλαια. Μία ενότητα θεωρίας μπορεί να περιλαμβάνει πολλά κεφάλαια, ενώ ένα quiz μπορεί να συνδέεται με κάποιο κεφάλαιο θεωρίας. Η σύνδεση αυτή επιτρέπει τη συσχέτιση των ερωτήσεων αξιολόγησης με το αντίστοιχο θεωρητικό υλικό.

Τέλος, οι ασκήσεις συνδέονται με τις προσπάθειες των χρηστών μέσω του πίνακα `exercise_attempts`. Κάθε προσπάθεια άσκησης αντιστοιχεί σε έναν χρήστη και σε μία συγκεκριμένη άσκηση.

4.3 Σχεδίαση Διεπαφής Χρήστη (UX/UI)

4.3.1 Σχεδιαστικές Αρχές Διεπαφής

Κατά τη σχεδίαση της διεπαφής χρήστη της πλατφόρμας δόθηκε ιδιαίτερη έμφαση στην απλότητα, την ευχρηστία και την οργάνωση του περιεχομένου. Βασικός στόχος ήταν η δημιουργία ενός περιβάλλοντος το οποίο να επιτρέπει στον χρήστη να αλληλεπιδρά εύκολα με το σύστημα, χωρίς πολύπλοκες διαδικασίες και περιττές λειτουργίες.

Η σχεδίαση της εφαρμογής βασίστηκε στις αρχές ευχρηστίας που παρουσιάστηκαν στο πλαίσιο του μαθήματος ΕΠΛ435 και ειδικότερα στα χαρακτηριστικά ευχρηστίας του Nielsen [1]. Πιο συγκεκριμένα, η ανάπτυξη της διεπαφής επικεντρώθηκε στις ακόλουθες αρχές:

- Ευκολία εκμάθησης (learnability)
- Υψηλή αποδοτικότητα χρήσης (efficiency)
- Χαμηλή συχνότητα σφαλμάτων (low error frequency)
- Ευκολία συγκράτησης γνώσης και επαναχρησιμοποίησης (memorability)
- Υποκειμενική ικανοποίηση του χρήστη (user satisfaction)

Οι αρχές αυτές επηρέασαν σημαντικά τον τρόπο με τον οποίο οργανώθηκαν οι οθόνες, τα μενού πλοήγησης, οι φόρμες εισόδου και οι βασικές λειτουργίες της εφαρμογής.

Η δομή της εφαρμογής βασίζεται σε καθαρό και οργανωμένο layout, ώστε οι βασικές λειτουργίες να είναι άμεσα προσβάσιμες. Οι σελίδες της εφαρμογής ακολουθούν κοινή σχεδιαστική λογική, με συνεπή χρήση χρωμάτων, κουμπιών, φορμών και στοιχείων

πλοήγησης. Με τον τρόπο αυτό επιτυγχάνεται οπτική συνέπεια και βελτιώνεται η συνολική εμπειρία χρήστη.

Η πλοήγηση σχεδιάστηκε με απλό και άμεσο τρόπο, επιτρέποντας στον χρήστη να έχει γρήγορη πρόσβαση στις βασικές λειτουργίες της πλατφόρμας, όπως η προβολή θεωρίας, η επίλυση ασκήσεων, η συμμετοχή σε quizzes και η προβολή αποτελεσμάτων. Παράλληλα, πραγματοποιήθηκε σαφής διαχωρισμός μεταξύ του περιβάλλοντος φοιτητή και του περιβάλλοντος διαχειριστή, ώστε κάθε κατηγορία χρήστη να έχει πρόσβαση μόνο στις λειτουργίες που σχετίζονται με τον ρόλο της.

Επιπλέον, το σύστημα παρέχει άμεση ανατροφοδότηση για τις ενέργειες του χρήστη. Για παράδειγμα, μετά την ολοκλήρωση μιας άσκησης ή ενός quiz εμφανίζονται αποτελέσματα, βαθμολογία και επεξηγήσεις, επιτρέποντας στον χρήστη να βλέπει την επίδοσή του και να κατανοεί τα λάθη του.

Μετά τη σύνδεση στο σύστημα, ο χρήστης μεταφέρεται στο βασικό περιβάλλον της πλατφόρμας, από όπου αποκτά πρόσβαση στις κύριες λειτουργίες, όπως η θεωρία, οι ασκήσεις, τα quizzes και στην περίπτωση διαχειριστή, οι λειτουργίες διαχείρισης περιεχομένου και χρηστών.

Η πλοήγηση στην εφαρμογή πραγματοποιείται μέσω ενός σταθερού navigation bar στο πάνω μέρος της σελίδας, το οποίο εμφανίζεται σε όλες τις βασικές οθόνες. Με αυτόν τον τρόπο, διευκολύνεται η μετάβαση μεταξύ των διαθέσιμων ενοτήτων και παρέχεται άμεση δυνατότητα αποσύνδεσης από το σύστημα.

Η σχεδίαση της διεπαφής βασίζεται σε light mode περιβάλλον με χρωματική παλέτα μπλε και άσπρου χρώματος, ώστε να έχει καθαρή εμφάνιση και ευκολότερη ανάγνωση του περιεχομένου. Παράλληλα, χρησιμοποιούνται ευδιάκριτα κουμπιά, σαφείς τίτλοι και κάρτες περιεχομένου, με στόχο τη βελτίωση της εμπειρίας χρήσης και την ευκολότερη αλληλεπίδραση με τις λειτουργίες της πλατφόρμας.

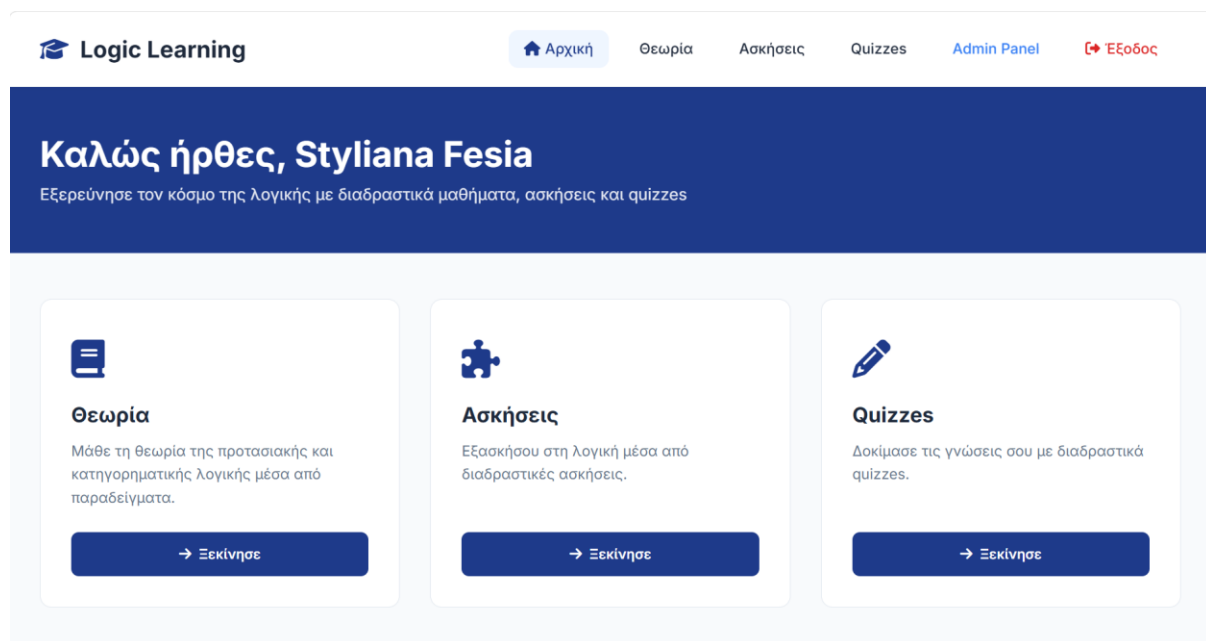
4.3.2 Διεπαφή Φοιτητή

Η διεπαφή του φοιτητή διαμορφώθηκε έτσι ώστε να προσφέρει ένα καθαρό και εύκολα κατανοητό περιβάλλον μάθησης.

4.3.2.1 Dashboard φοιτητή

Μετά τη σύνδεση στο σύστημα, ο φοιτητής μεταφέρεται στο κεντρικό dashboard της πλατφόρμας. Η σελίδα αυτή παρουσιάζει συνοπτικά τις βασικές λειτουργίες της πλατφόρμας και λειτουργεί ως αρχικό σημείο αλληλεπίδρασης του χρήστη με το σύστημα.

Στο πάνω μέρος εμφανίζεται μήνυμα καλωσορίσματος και από κάτω ακολουθούν τρεις βασικές κάρτες επιλογών για πρόσβαση στη θεωρία, επίλυση ασκήσεων και συμμετοχή σε quizzes. Κάθε κάρτα περιλαμβάνει εικονίδιο, σύντομη περιγραφή και κουμπί μετάβασης στη συγκεκριμένη λειτουργία.

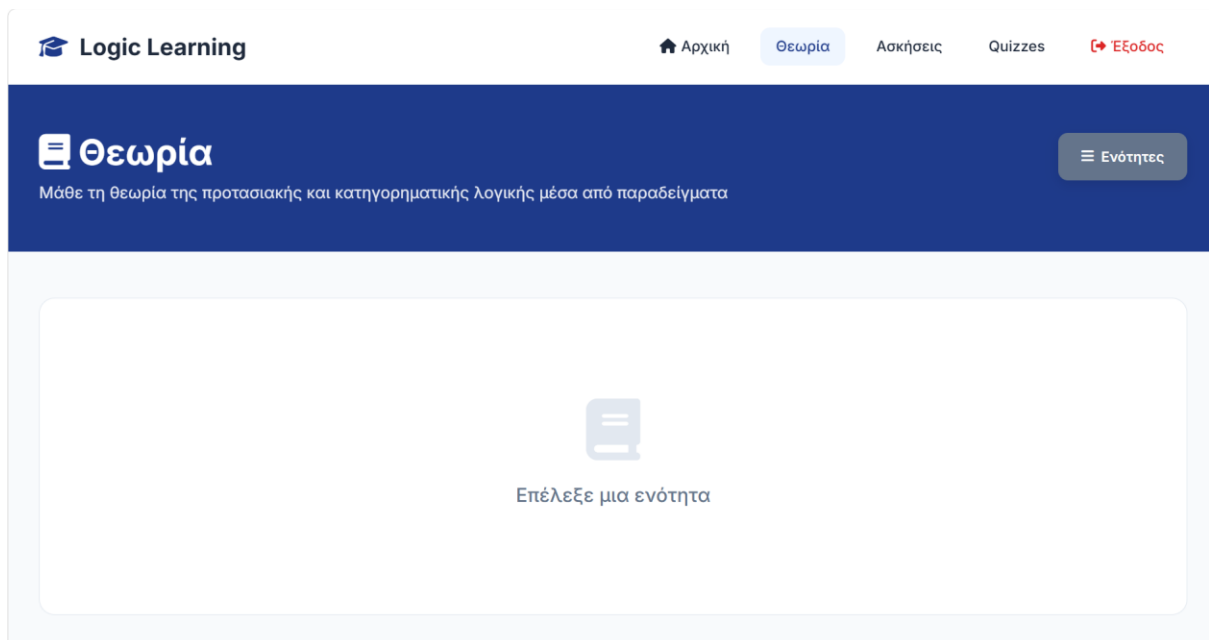


Εικόνα 4: Κεντρικό dashboard φοιτητή

4.3.2.2 Σελίδα θεωρίας

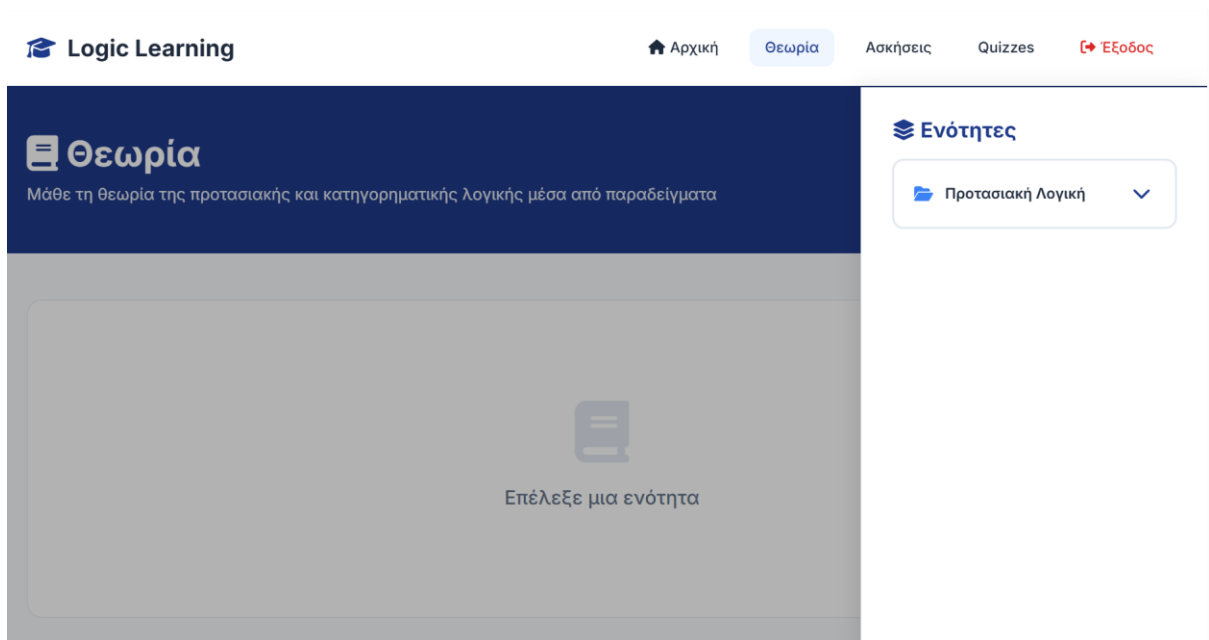
Η σελίδα θεωρίας σχεδιάστηκε ώστε να παρέχει οργανωμένη παρουσίαση του εκπαιδευτικού υλικού. Ο χρήστης μπορεί να πλοηγηθεί στις διαθέσιμες ενότητες και κεφάλαια μέσω ειδικού side panel πλοήγησης.

Αρχικά εμφανίζεται η βασική σελίδα θεωρίας, όπου ο φοιτητής πρέπει να επιλέξει μία ενότητα.



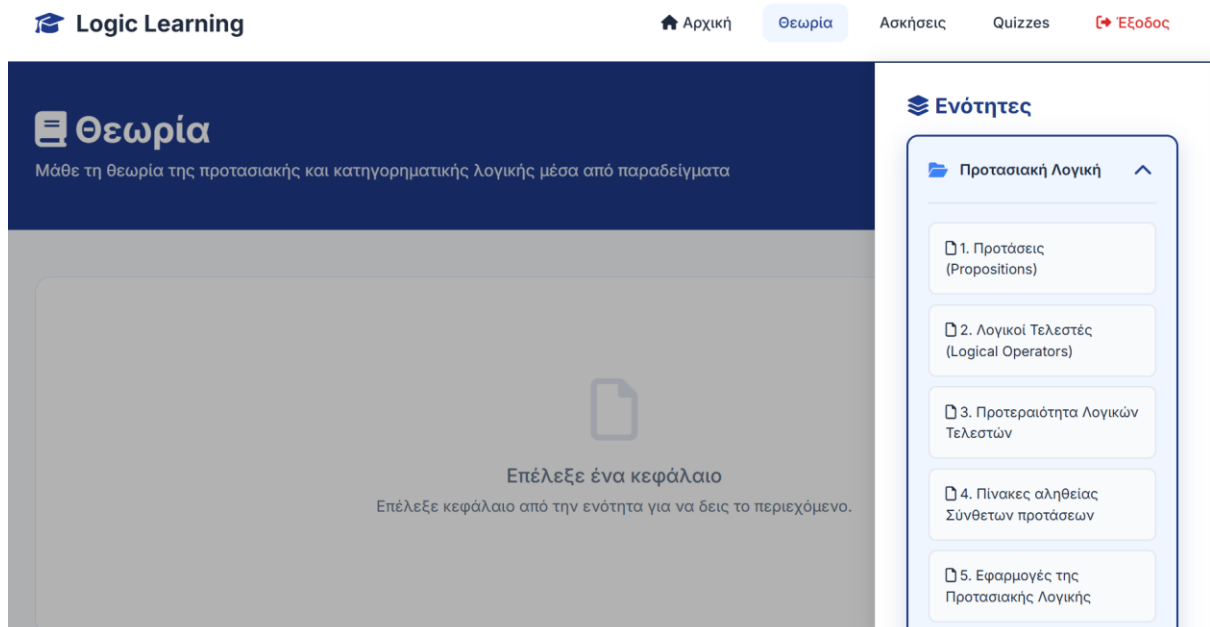
Εικόνα 5: Αρχική σελίδα θεωρίας

Όταν ο χρήστης πατήσει το κουμπί «Ενότητες» εμφανίζεται πλαϊνό panel πλοήγησης το οποίο περιέχει τις διαθέσιμες ενότητες της θεωρίας.



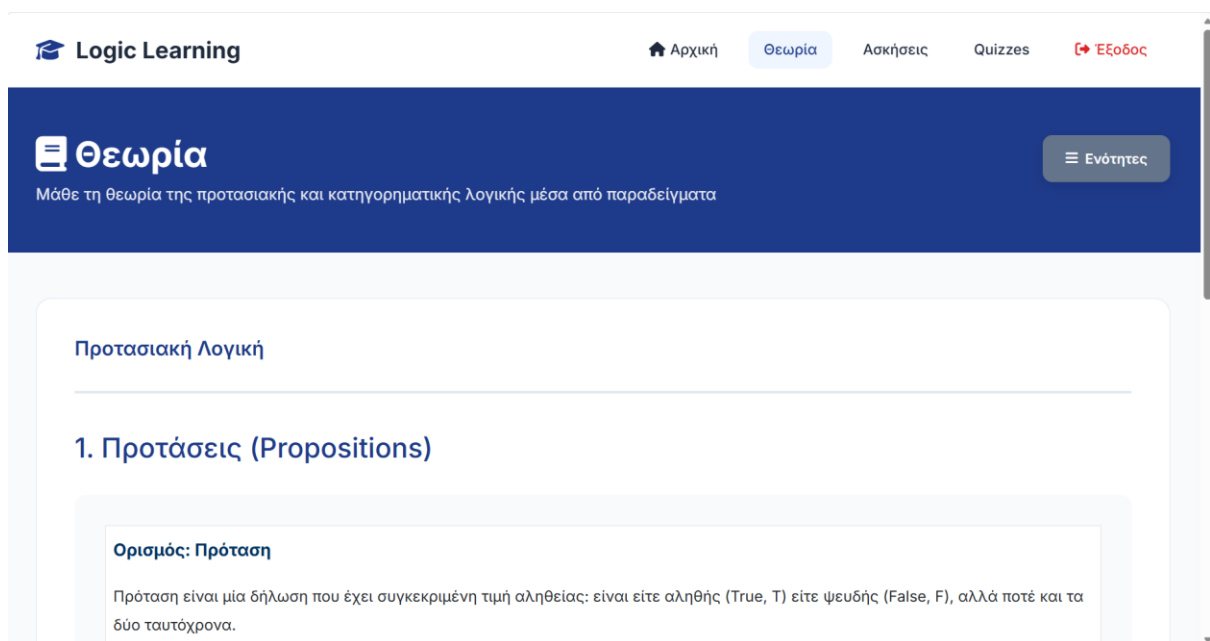
Εικόνα 6: Panel ενότητων θεωρίας

Στη συνέχεια, ο χρήστης μπορεί να επιλέξει μία ενότητα και να εμφανίσει τα διαθέσιμα κεφάλαια θεωρίας.



Εικόνα 7: Προβολή κεφαλαίων θεωρίας

Με την επιλογή ενός κεφαλαίου εμφανίζεται το αντίστοιχο θεωρητικό περιεχόμενο στο κέντρο της σελίδας. Το περιεχόμενο παρουσιάζεται σε μορφή οργανωμένων sections με τίτλους, παραδείγματα και μορφοποιημένο κείμενο, ώστε να διευκολύνεται η μελέτη από τον χρήστη.



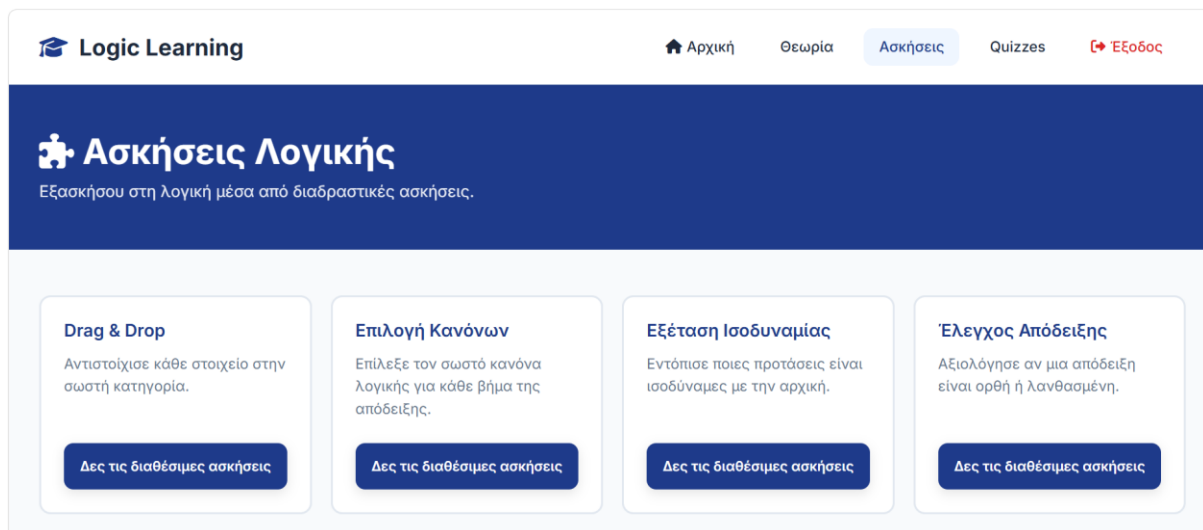
Εικόνα 8: Προβολή θεωρητικού περιεχομένου

Η συγκεκριμένη σχεδίαση επιτρέπει στον φοιτητή να μεταβαίνει γρήγορα μεταξύ ενοτήτων και κεφαλαίων χωρίς να αλλάζει σελίδα, βελτιώνοντας έτσι τη συνολική εμπειρία πλοήγησης και μελέτης.

4.3.2.3 Σελίδα ασκήσεων

Η σελίδα ασκήσεων σχεδιάστηκε έτσι ώστε να προσφέρει στον φοιτητή τη δυνατότητα να εφαρμόσει στην πράξη τις έννοιες της Προτασιακής Λογικής μέσα από διαφορετικές μορφές δραστηριοτήτων. Η σελίδα παρουσιάζει οργανωμένα τους διαθέσιμους τύπους ασκήσεων, επιτρέποντας στον χρήστη να επιλέξει το είδος δραστηριότητας στο οποίο επιθυμεί να επιλύσει ασκήσεις.

Η πλατφόρμα υποστηρίζει τέσσερις βασικούς τύπους ασκήσεων: Drag & Drop, Επιλογή Κανόνων, Εξέταση Ισοδυναμίας και Έλεγχος Απόδειξης. Κάθε κατηγορία εμφανίζεται σε ξεχωριστό card με τίτλο, σύντομη περιγραφή και κουμπί μετάβασης στις διαθέσιμες ασκήσεις της συγκεκριμένης κατηγορίας.



Εικόνα 9: Σελίδα επιλογής τύπων ασκήσεων

Μετά την επιλογή ενός τύπου άσκησης, ο χρήστης μεταφέρεται στη σελίδα με τις διαθέσιμες ασκήσεις της συγκεκριμένης κατηγορίας. Εκεί εμφανίζεται η λίστα των ασκήσεων, μαζί με το επίπεδο δυσκολίας κάθε άσκησης και το αντίστοιχο κουμπί έναρξης προσπάθειας. Επιπλέον, υπάρχει και κουμπί «Πίσω στους τύπους ασκήσεων», ώστε ο φοιτητής να μπορεί εύκολα να επιστρέψει στην προηγούμενη οθόνη και να επιλέξει διαφορετικό τύπο άσκησης.

Άσκηση Drag & Drop

The screenshot shows the 'Logic Learning' interface. At the top, there are navigation links: 'Αρχική', 'Θεωρία', 'Ασκήσεις' (highlighted), 'Quizzes', and 'Εξοδος'. Below the navigation bar, there is a button '← Πίσω στους τύπους ασκήσεων'. The main heading is 'Drag and Drop' with the subtitle 'Διαθέσιμες ασκήσεις'. There are three exercise cards:

- Άσκηση 1 - Drag&Drop**: Includes a 'Drag & Drop' icon and a 'Εύκολο' (Easy) difficulty tag. A '▶ Προσπάθησε' button is on the right.
- Άσκηση 2 - Drag&Drop**: Includes a 'Drag & Drop' icon and a 'Μεσαίο' (Medium) difficulty tag. A '▶ Προσπάθησε' button is on the right.
- Άσκηση 3 - Drag&Drop**: Includes a 'Drag & Drop' icon and a 'Δύσκολο' (Difficult) difficulty tag. A '▶ Προσπάθησε' button is on the right.

Εικόνα 10: Διαθέσιμες ασκήσεις Drag & Drop

Όταν ο φοιτητής επιλέξει μία άσκηση τύπου Drag & Drop, εμφανίζονται τα διαθέσιμα στοιχεία προς αντιστοίχιση και οι αντίστοιχες κατηγορίες. Ο χρήστης καλείται να μεταφέρει κάθε στοιχείο στη σωστή κατηγορία και στη συνέχεια να υποβάλει την απάντησή του.

The screenshot shows the 'Άσκηση 3 - Drag&Drop' interface. At the top, there is a title bar with a close button. Below the title bar, there is a instruction: 'Σύρετε κάθε λεκτική διατύπωση στη σωστή λογική μορφή.' (Drag each verbal statement to the correct logical form). There are three statements:

- P: Ο φοιτητής παίρνει πτυχίο
- Q: Ο φοιτητής έχει περάσει όλα τα μαθήματα
- R: Ο φοιτητής έχει ολοκληρώσει τη διπλωματική εργασία

Below the statements, there is a section titled 'Στοιχεία προς Σύριμο' (Items to be dragged). It contains four blue boxes with text:

- Δεν ισχύει ότι ο φοιτητής παίρνει πτυχίο ενώ δεν έχει περάσει όλα τα μαθήματα.
- Δεν ισχύει ότι ο φοιτητής παίρνει πτυχίο ενώ δεν έχει περάσει όλα τα μαθήματα ή δεν έχει ολοκληρώσει τη διπλωματική εργασία
- Ο φοιτητής παίρνει πτυχίο μόνο αν έχει περάσει όλα τα μαθήματα και έχει ολοκληρώσει τη διπλωματική εργασία.
- Αν ο φοιτητής δεν έχει περάσει όλα τα μαθήματα ή δεν έχει ολοκληρώσει τη διπλωματική εργασία, τότε δεν παίρνει

On the right, there are two logical form boxes:

- $P \rightarrow (Q \wedge R)$ with a placeholder text: 'Σύρετε το στοιχείο που ταιριάζει εδώ' (Drag the element that fits here).
- $\neg(P \wedge (\neg Q \vee \neg R))$ with a placeholder text: 'Σύρετε το στοιχείο που ταιριάζει εδώ' (Drag the element that fits here).

Εικόνα 11: Άσκηση Drag & Drop

Άσκηση Επιλογής Κανόνων

The screenshot shows the 'Logic Learning' interface. At the top, there are navigation links: 'Αρχική', 'Θεωρία', 'Ασκήσεις' (highlighted), 'Quizzes', and 'Έξοδος'. Below the navigation bar, there is a button '← Πίσω στους τύπους ασκήσεων'. The main heading is 'Επιλογή Κανόνων' with the subtitle 'Διαθέσιμες ασκήσεις'. There are three exercise cards:

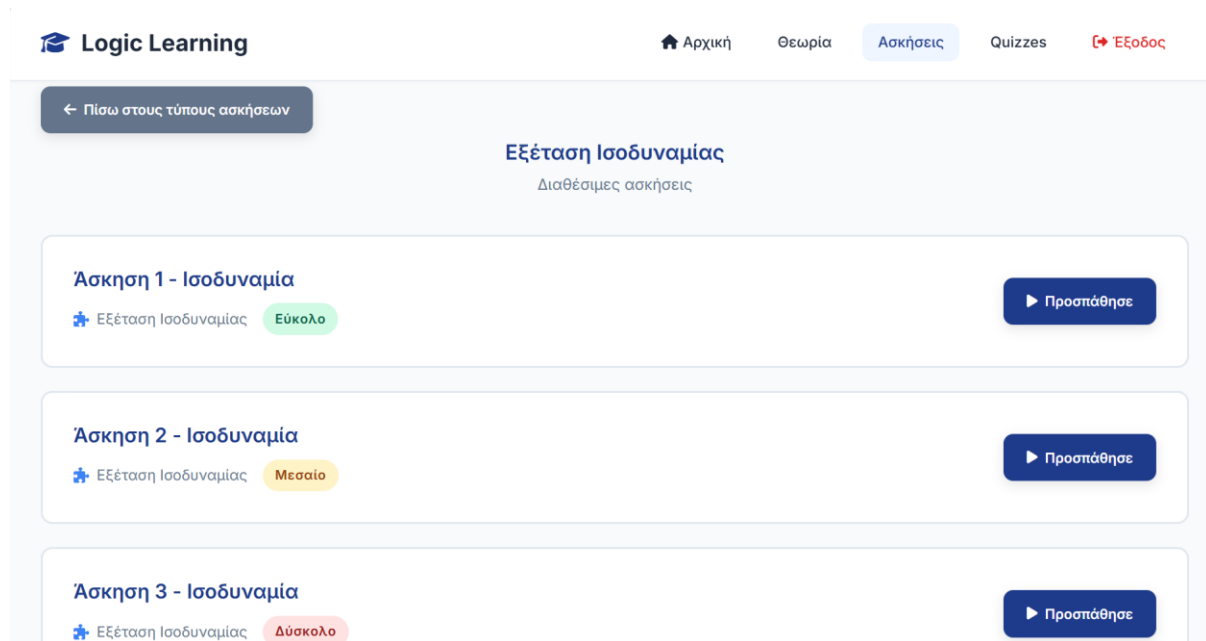
- Άσκηση 1 - Λογικοί Κανόνες**: Includes a puzzle icon, 'Επιλογή Κανόνων', 'Εύκολο' (green), and a '▶ Προσπάθησε' button.
- Άσκηση 2 - Λογικοί Κανόνες**: Includes a puzzle icon, 'Επιλογή Κανόνων', 'Μεσαίο' (yellow), and a '▶ Προσπάθησε' button.
- Άσκηση 3 - Λογικοί Κανόνες**: Includes a puzzle icon, 'Επιλογή Κανόνων', 'Δύσκολο' (red), and a '▶ Προσπάθησε' button.

Εικόνα 12: Διαθέσιμες ασκήσεις επιλογής κανόνων

Όταν ο φοιτητής επιλέξει μία άσκηση επιλογής κανόνων συλλογισμού, εμφανίζεται μία ακολουθία βημάτων λογικών ισοδυναμιών ή απόδειξης. Για κάθε βήμα, ο χρήστης καλείται να επιλέξει από το dropdown menu τον λογικό κανόνα που εφαρμόστηκε και στη συνέχεια να υποβάλει την απάντησή του.

The screenshot shows the exercise interface for 'Άσκηση 1 - Λογικοί Κανόνες'. It includes a close button (X) in the top right corner. The text says: 'Ποιοι λογικοί κανόνες έχουν εφαρμοστεί στην πιο κάτω ακολουθία λογικών ισοδυναμιών;'. Below this, it says: 'Επιλέξτε τον σωστό κανόνα για κάθε βήμα:'. The first step is labeled 'Βήμα 1:' and shows the logical expression:
$$\neg((P \leftrightarrow Q) \vee R) \equiv \neg(((P \rightarrow Q) \wedge (Q \rightarrow P)) \vee R)$$
. Below the expression, it says 'Επιλέξτε κανόνα:'. There is a dropdown menu with the placeholder '-- Επιλέξτε κανόνα --'. The dropdown is open, showing a list of logical laws: Identity law, Domination law, Idepotent law, Νόμος διπλής άρνησης, Νόμος αντιμεταθετικότητας, Νόμος προσαιρεριστικότητας, Νόμος επιμεριστικότητας, Ορισμός συνεπαγωγής, Ορισμός διπλής συνεπαγωγής, Νόμος De Morgan, Absorption law, and Negation law.

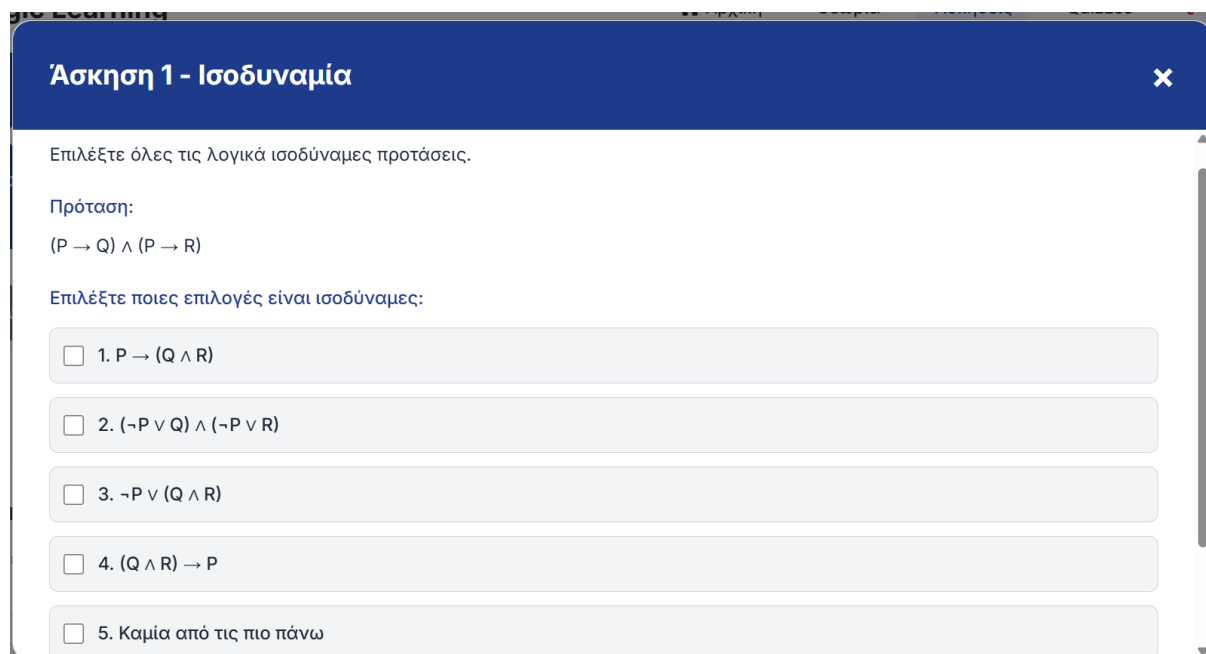
Εικόνα 13: Άσκηση επιλογής κανόνων



Εικόνα 14: Διαθέσιμες ασκήσεις εξέτασης ισοδυναμίας

Στις ασκήσεις εξέτασης ισοδυναμίας, εμφανίζεται μία αρχική λογική πρόταση και ένα σύνολο πιθανών επιλογών. Ο φοιτητής καλείται να επιλέξει όλες τις προτάσεις που είναι λογικά ισοδύναμες με την αρχική.

Οι απαντήσεις δίνονται μέσω επιλογών πολλαπλής επιλογής και μπορεί να υπάρχουν περισσότερες από μία σωστές απαντήσεις και έπειτα ο χρήστης μπορεί να υποβάλει την απάντησή του.



Εικόνα 15: Άσκηση εξέτασης ισοδυναμίας

Άσκηση Ελέγχου Απόδειξης

The screenshot shows the 'Logic Learning' interface. At the top, there are navigation links: 'Αρχική', 'Θεωρία', 'Ασκήσεις' (highlighted), 'Quizzes', and 'Έξοδος'. Below the navigation bar, there is a button 'Πίσω στους τύπους ασκήσεων'. The main heading is 'Έλεγχος Απόδειξης' with the subtitle 'Διαθέσιμες ασκήσεις'. There are three exercise cards:

- Άσκηση 1 - Απόδειξη**: 'Έλεγχος Απόδειξης' (blue icon), 'Εύκολο' (green tag), 'Προσπάθησε' (blue button).
- Άσκηση 2 - Απόδειξη**: 'Έλεγχος Απόδειξης' (blue icon), 'Μεσαίο' (yellow tag), 'Προσπάθησε' (blue button).
- Άσκηση 3 - Απόδειξη**: 'Έλεγχος Απόδειξης' (blue icon), 'Δύσκολο' (red tag), 'Προσπάθησε' (blue button).

Εικόνα 16: Διαθέσιμες ασκήσεις ελέγχου απόδειξης

Στις ασκήσεις ελέγχου απόδειξης, παρουσιάζεται μία ολοκληρωμένη και ο χρήστης καλείται να εξετάσει αν η απόδειξη είναι ορθή και να επιλέξει ανάμεσα στις απαντήσεις «Ναι» ή «Όχι» και στη συνέχεια να υποβάλει την απάντηση του.

The screenshot shows the details of 'Άσκηση 1 - Απόδειξη'. The question is 'Είναι η πιο κάτω απόδειξη ορθή;'. The proof is as follows:

Απόδειξη:

Θέλουμε να αποδείξουμε ότι $\neg(P \rightarrow Q) \equiv \neg P \wedge \neg Q$

- $\neg(P \rightarrow Q) \equiv \neg(\neg P \vee Q)$ Ορισμός συνεπαγωγής
- $\equiv \neg(\neg P) \wedge \neg Q$ Νόμος De Morgan
- $\equiv \neg P \wedge \neg Q$ Νόμος διπλής άρνησης

Είναι σωστή η απόδειξη;

☐ Ναι

☐ Όχι

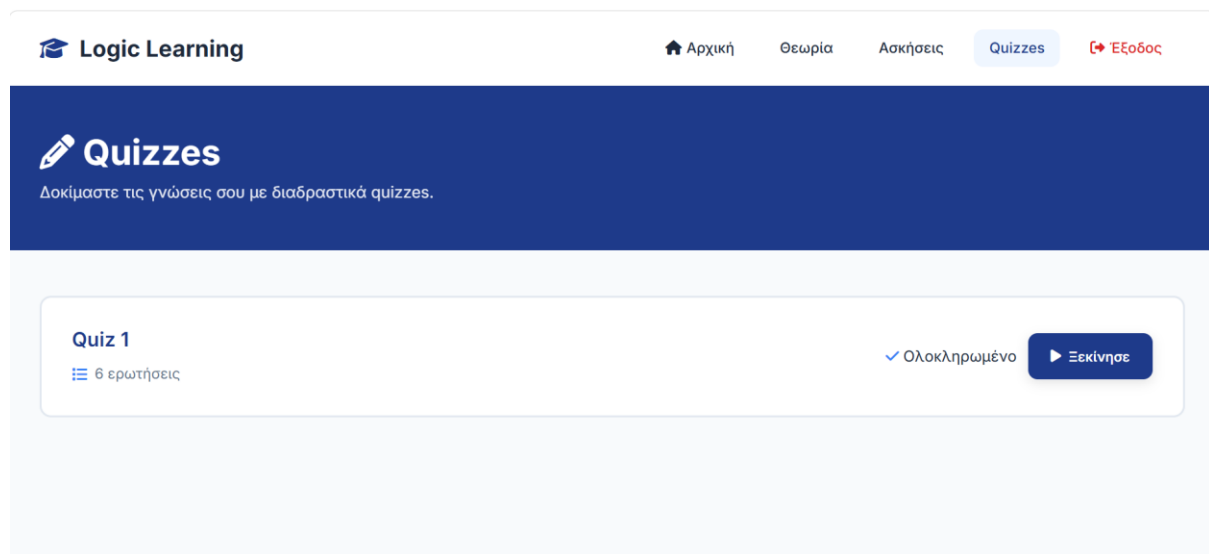
Υποβολή

Εικόνα 17: Άσκηση ελέγχου απόδειξης

4.3.2.4 Σελίδα quizzes

Η ενότητα quizzes σχεδιάστηκε με στόχο την αξιολόγηση των γνώσεων του χρήστη μέσω διαδραστικών ερωτήσεων πολλαπλής επιλογής. Η σελίδα παρουσιάζει τα διαθέσιμα quizzes της πλατφόρμας σε μορφή λίστας, παρέχοντας βασικές πληροφορίες για κάθε quiz, όπως ο αριθμός ερωτήσεων και η κατάσταση ολοκλήρωσης.

Κάθε quiz εμφανίζεται μέσα σε ξεχωριστό card, το οποίο περιλαμβάνει τίτλο, τον αριθμό ερωτήσεων που περιέχει και κουμπί έναρξης της προσπάθειας. Επιπλέον, εμφανίζεται ένδειξη ολοκλήρωσης όταν ο χρήστης έχει ολοκληρώσει ήδη το συγκεκριμένο quiz.



Εικόνα 18: Σελίδα διαθέσιμων quizzes

Αν ο χρήστης δεν έχει ολοκληρώσει προηγουμένως ένα quiz, επιλέγοντας την έναρξη μεταφέρεται στη σελίδα εκτέλεσης του quiz, όπου παρουσιάζονται οι ερωτήσεις μαζί με τις διαθέσιμες επιλογές απαντήσεων. Ο χρήστης μπορεί να επιλέξει μία ή περισσότερες απαντήσεις, ανάλογα με τον τύπο της ερώτησης, και στη συνέχεια να υποβάλει το quiz για αξιολόγηση.

Quiz 1

Εξήγηση Βαθμολόγησης

1

Είναι οι πιο κάτω προτάσεις λογικά ισοδύναμες;
 $(\neg P \vee Q) \rightarrow Q \wedge (Q \rightarrow R) \wedge \neg R$ και $(Q \wedge \neg R) \wedge (\neg R \rightarrow \neg Q) \vee (P \wedge \neg Q)$

☐ Ναι
 ☐ Όχι

2

Επίλεξε όλες τις προτάσεις που είναι λογικά ισοδύναμες με την πρόταση: $((P \rightarrow Q) \wedge P) \vee ((P \rightarrow R) \wedge P)$

☐ 1. $P \wedge (Q \vee R)$
☐ 2. $(P \wedge Q) \vee (P \wedge R)$

Εικόνα 19: Σελίδα με ερωτήσεις ενός quiz

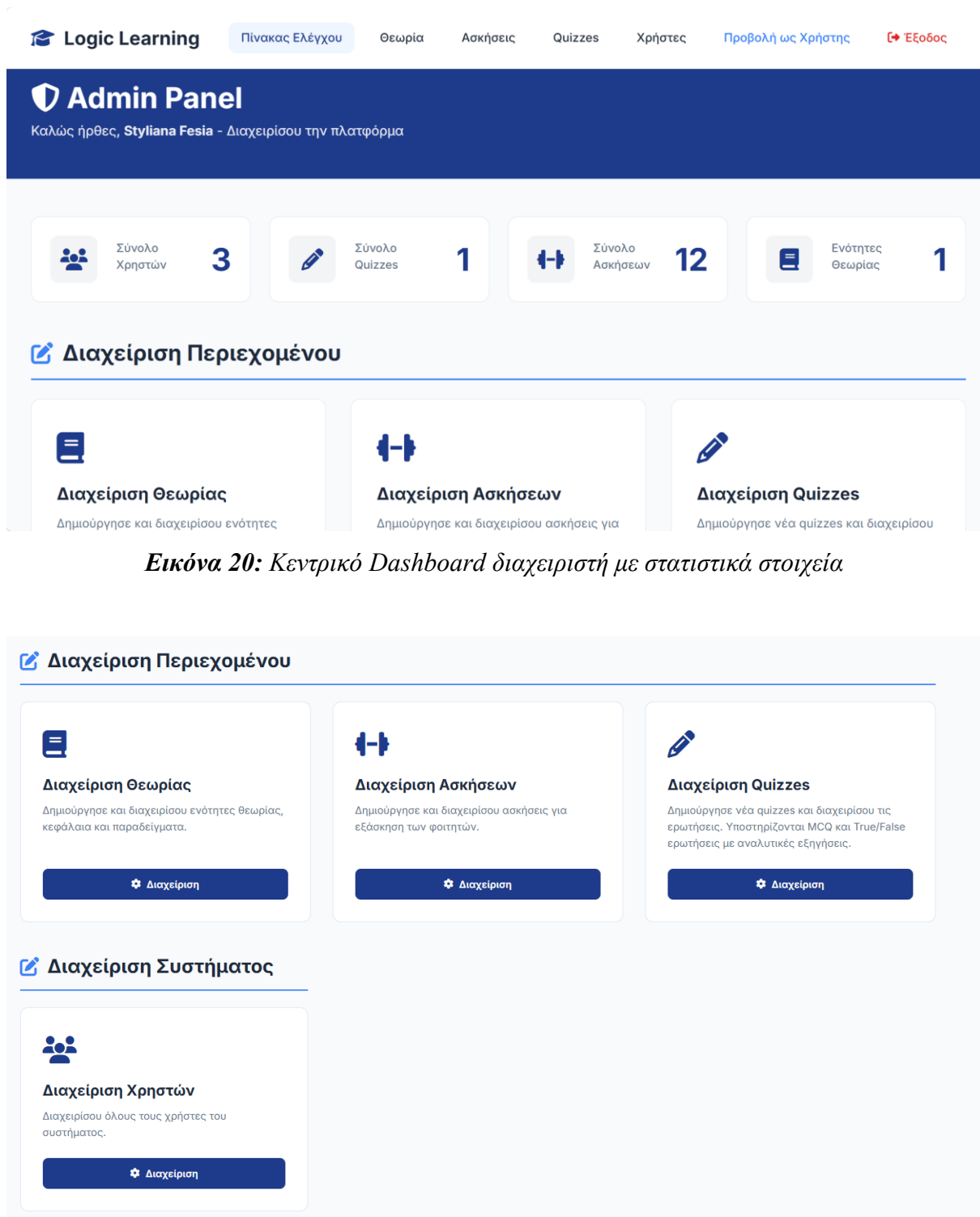
4.3.3 Διεπαφή Διαχειριστή

Ο διαχειριστής διαθέτει επιπλέον λειτουργίες σε σχέση με έναν απλό χρήστη, όπως δημιουργία, επεξεργασία και διαγραφή θεωρητικού υλικού, ασκήσεων, quizzes και χρηστών.

4.3.3.1 Dashboard διαχειριστή

Μετά τη σύνδεση του διαχειριστή στο σύστημα, μεταφέρεται στο κεντρικό dashboard της σελίδας. Όπως και στη σελίδα του φοιτητή, έτσι και εδώ υπάρχει μήνυμα καλωσορίσματος για τον διαχειριστή, με τη διαφορά ότι ακολουθούν διάφορα στατιστικά στοιχεία όπως συνολικός αριθμός χρηστών, αριθμός διαθέσιμων quizzes, αριθμός ασκήσεων και αριθμός ενοτήτων θεωρίας. Με αυτό τον τρόπο παρέχεται μια συνοπτική εικόνα της κατάστασης της πλατφόρμας. Επιπλέον, παρέχεται γρήγορη πρόσβαση στις βασικές λειτουργίες διαχείρισης, μέσω καρτών επιλογής.

39



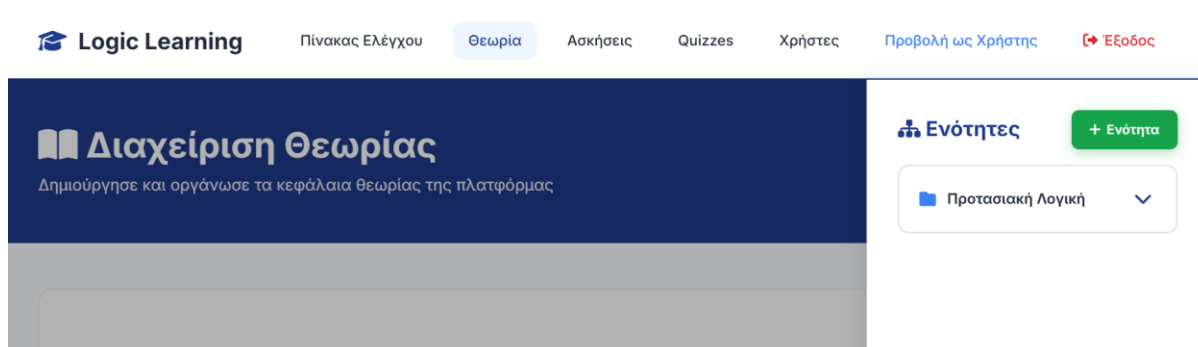
Εικόνα 20: Κεντρικό Dashboard διαχειριστή με στατιστικά στοιχεία

Εικόνα 21: Dashboard διαχειριστή με κάρτες για τις λειτουργίες διαχείρισης

4.3.3.2 Σελίδα διαχείρισης θεωρίας

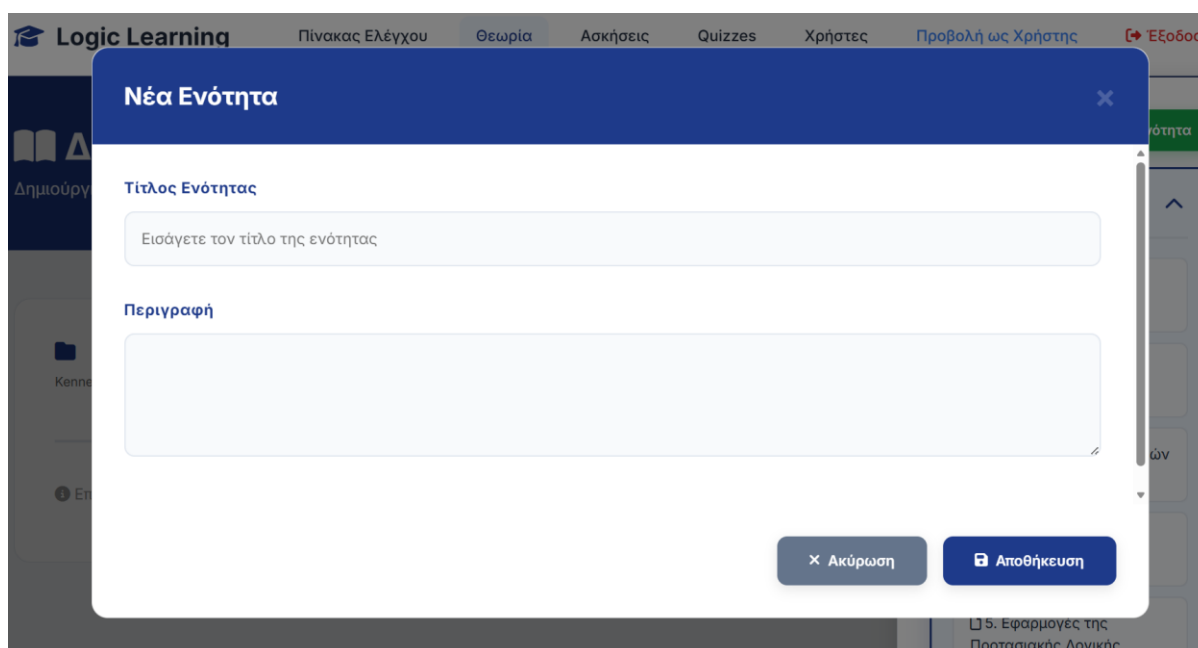
Η σελίδα διαχείρισης θεωρίας επιτρέπει στον διαχειριστή να οργανώνει το θεωρητικό υλικό της πλατφόρμας σε ενότητες και κεφάλαια και να το επεξεργάζεται.

Μόλις μεταφερθεί στη σελίδα διαχείρισης θεωρίας, ο διαχειριστής έχει τη δυνατότητα πατώντας το κουμπί «Ενότητες» και έπειτα από το side panel το κουμπί «+Ενότητα», να δημιουργήσει νέες ενότητες ή να επιλέξει υπάρχουσες ενότητες και κεφάλαια για επεξεργασία ή διαγραφή.



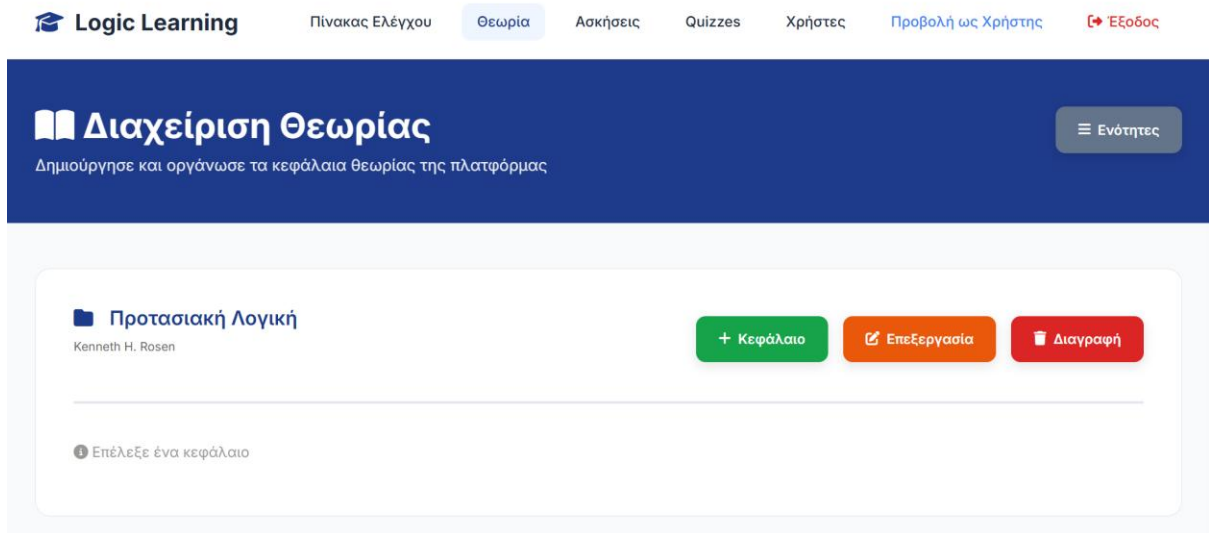
Εικόνα 22: Side panel για προσθήκη ή επιλογή ενότητας για επεξεργασία

Ο διαχειριστής μπορεί να δημιουργήσει μια ενότητα πατώντας το κουμπί «+Ενότητα».



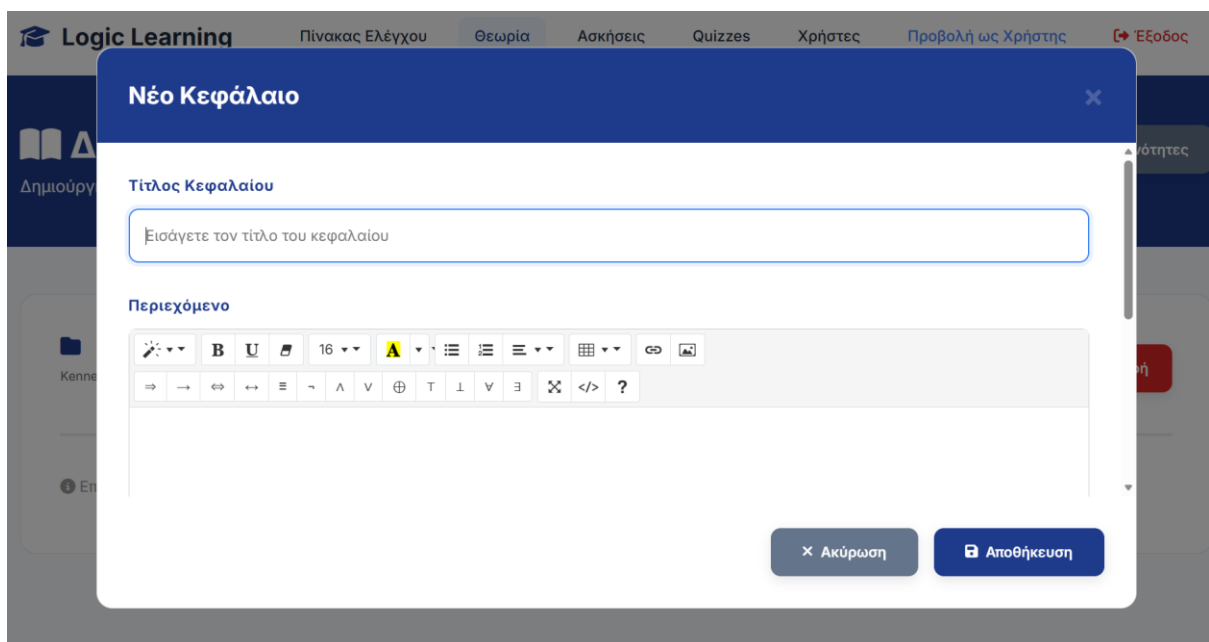
Εικόνα 23: Modal δημιουργίας ενότητας

Με την επιλογή μιας ενότητα, εμφανίζονται οι επιλογές κουμπιών «+Κεφάλαιο», «Επεξεργασία» και «Διαγραφή» όπου μπορεί να προσθέσει νέο κεφάλαιο στην επιλεγμένη ενότητα, να την επεξεργαστεί ή να την διαγράψει.



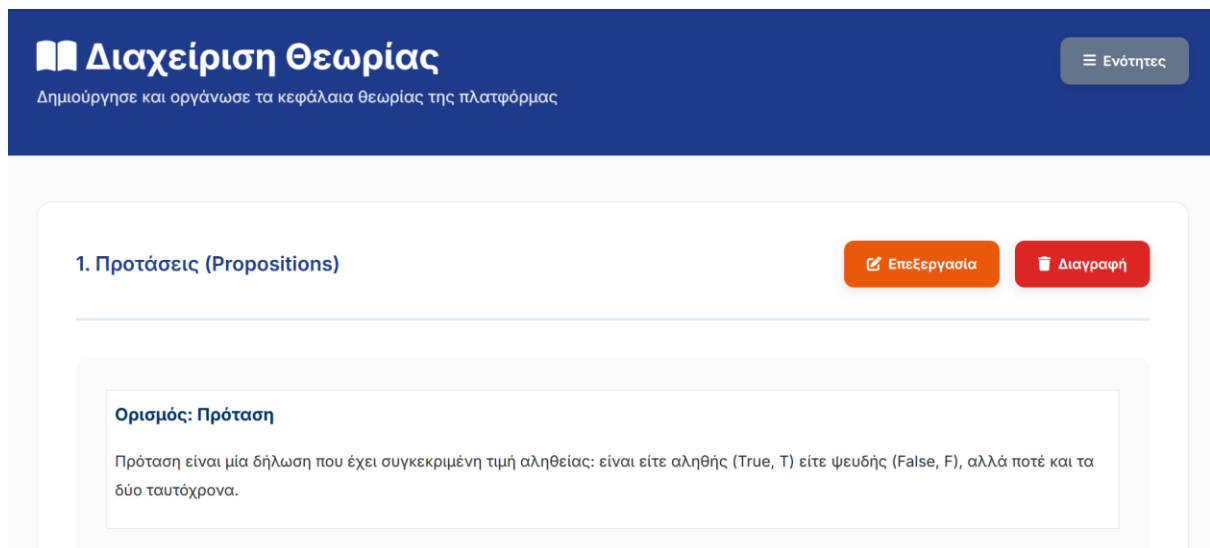
Εικόνα 24: Επιλογές προσθήκης κεφαλαίου στην ενότητα, επεξεργασίας και διαγραφής ενότητας

Με την επιλογή του κουμπιού «+Κεφάλαιο» ο διαχειριστής μπορεί να προσθέσει νέο κεφάλαιο συμπληρώνοντας τίτλο και περιεχόμενο.



Εικόνα 25: Modal προσθήκης κεφαλαίου

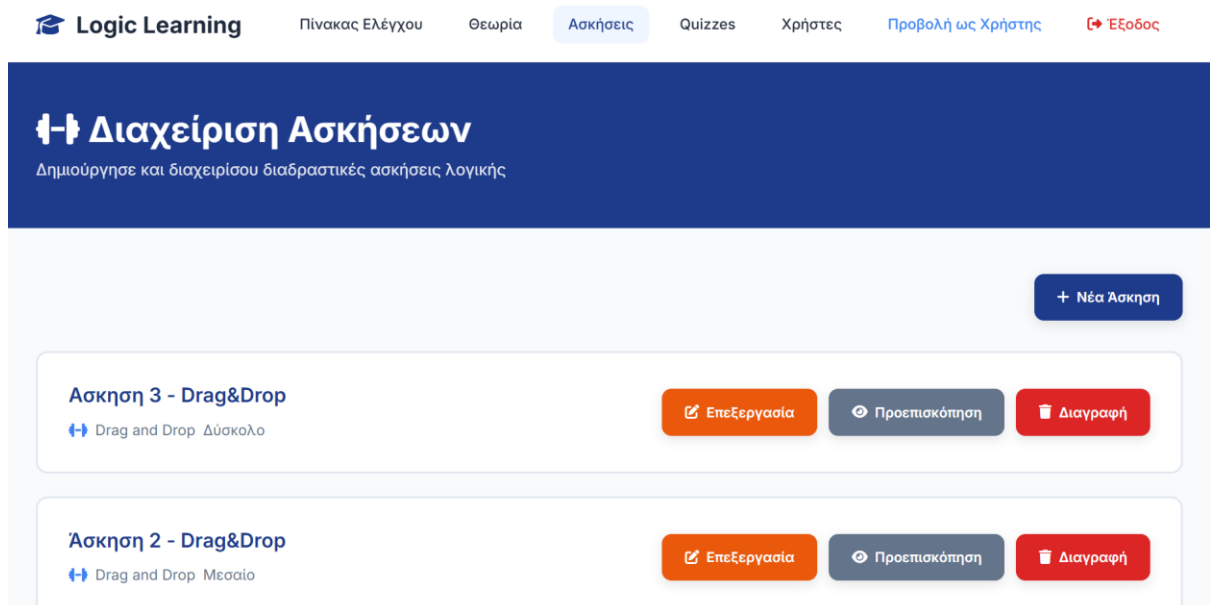
Όταν ο χρήστης επιλέξει ένα κεφάλαιο από μία ενότητα εμφανίζεται το περιεχόμενο του κεφαλαίου μαζί με τις επιλογές «Επεξεργασία» και «Διαγραφή» για να επεξεργαστεί ή να διαγράψει το συγκεκριμένο κεφάλαιο.



Εικόνα 26: Επιλογές επεξεργασίας και διαγραφής κεφαλαίου

4.3.3.3 Σελίδα διαχείρισης ασκήσεων

Η ενότητα διαχείρισης ασκήσεων επιτρέπει στον διαχειριστή να δημιουργεί και να επεξεργάζεται ασκήσεις. Η σελίδα παρουσιάζει όλες τις διαθέσιμες ασκήσεις με πληροφορίες όπως ο τίτλος, ο τύπος άσκησης και το επίπεδο δυσκολίας με επιλογές για επεξεργασία, προεπισκόπηση όπου μπορεί να δει την τελική μορφή της άσκησης για σκοπούς επαλήθευσης και διαγραφή.



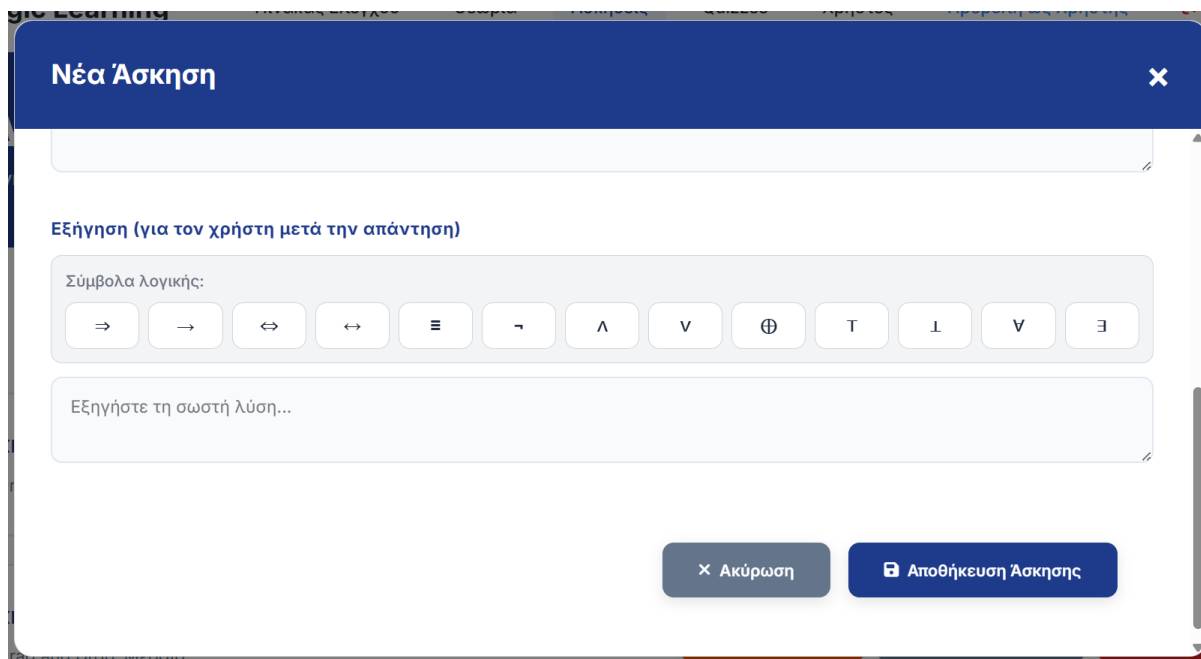
Εικόνα 27: Σελίδα διαχείρισης ασκήσεων

Μέσω της επιλογής «+Νέα Άσκηση» ο διαχειριστής μπορεί να δημιουργήσει νέα άσκηση καθορίζοντας τίτλο, τύπο άσκησης, επίπεδο δυσκολίας, οδηγίες προς τον χρήστη και

επεξήγηση που εμφανίζεται μετά την ολοκλήρωση της προσπάθειας. Παρέχονται επίσης έτοιμα σύμβολα λογικής για ευκολότερη εισαγωγή λογικών εκφράσεων.



Εικόνα 28: Γενική μορφή modal δημιουργίας νέας άσκησης



Εικόνα 29: Γενική μορφή modal δημιουργίας νέας άσκησης

Η πλατφόρμα υποστηρίζει 4 τύπους ασκήσεων, όπου κάθε τύπος διαθέτει διαφορετική δομή περιεχομένου.

Ασκήσεις Drag & Drop

Στις ασκήσεις Drag & Drop, ο διαχειριστής δημιουργεί νέα άσκηση ορίζοντας τα στοιχεία προς σύρσιμο, τις κατηγορίες αντιστοίχισης και τα σωστά ζευγάρια στοιχείων-κατηγοριών.

The interface is titled "Νέα Άσκηση" (New Exercise) and includes a close button (X). It contains the following sections:

- Περιεχόμενο άσκησης** (Exercise Content): A text area with instructions: "Δώστε τα στοιχεία και τις κατηγορίες. Τα στοιχεία μπορούν να είναι ίσα ή περισσότερα από τις κατηγορίες (δηλαδή κάποια στοιχεία μπορεί να μην αντιστοιχούν σε καμία κατηγορία)." (Provide the elements and categories. Elements can be equal to or more than the categories (i.e., some elements may not correspond to any category)).
- Σύμβολα λογικής:** A row of logical symbols: $\Rightarrow$, $\rightarrow$, $\leftrightarrow$, $\Leftarrow$, $\equiv$, $\neg$, $\wedge$, $\vee$, $\oplus$, $\top$, $\perp$, $\forall$, $\exists$.
- Στοιχεία προς Σύρσιμο** (Elements to Drag): A text input field labeled "Στοιχείο προς σύρσιμο" (Element to drag) with a delete icon (trash can) and a "+ Προσθήκη Στοιχείου" (Add Element) button.
- Κατηγορίες** (Categories): A text input field labeled "Κατηγορία" (Category) with a delete icon (trash can) and a "+ Προσθήκη Κατηγορίας" (Add Category) button.

Εικόνα 30: Περιεχόμενο άσκησης Drag & Drop

The interface is titled "Νέα Άσκηση" (New Exercise) and includes a close button (X). It contains the following sections:

- Σωστό Ταίριασμα** (Correct Matching): A section with a text area labeled "Ορίστε ποια κατηγορία αντιστοιχεί σε κάθε στοιχείο:" (Specify which category corresponds to each element:). Below this is a table with two columns: "Κατηγορία 1" (Category 1) and "Στοιχείο 1" (Element 1). A dropdown menu is open, showing options: "-- Κανένα στοιχείο --" (No element), "-- Κανένα στοιχείο --" (No element), and "Στοιχείο 1" (Element 1). The second option is highlighted in blue.
- Εξήγηση (για τον χρήστη μετά την απάντηση)** (Explanation (for the user after the answer)): A section with a text area labeled "Ορίστε ποια κατηγορία αντιστοιχεί σε κάθε στοιχείο:" (Specify which category corresponds to each element:).
- Σύμβολα λογικής:** A row of logical symbols: $\Rightarrow$, $\rightarrow$, $\leftrightarrow$, $\Leftarrow$, $\equiv$, $\neg$, $\wedge$, $\vee$, $\oplus$, $\top$, $\perp$, $\forall$, $\exists$.

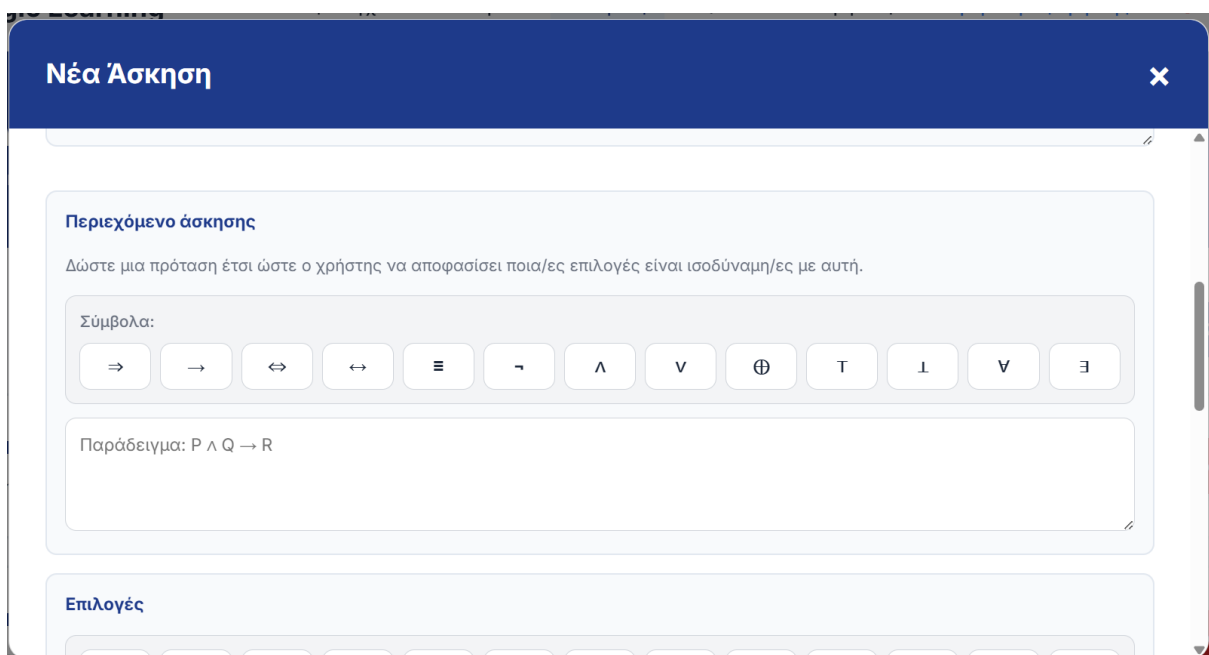
Εικόνα 31: Σωστές απαντήσεις άσκησης Drag & Drop



Εικόνα 32: Προεπισκόπηση άσκησης Drag & Drop

Ασκήσεις Εξέτασης Ισοδυναμίας

Στις ασκήσεις εξέτασης ισοδυναμίας, ο διαχειριστής γράφει τη βασική πρόταση και ορίζει πιθανές επιλογές απαντήσεων, καθώς και τις σωστές επιλογές που αντιστοιχούν στις λογικά ισοδύναμες εκφράσεις της αρχικής. Μπορεί να υπάρχουν πολλές σωστές απαντήσεις.



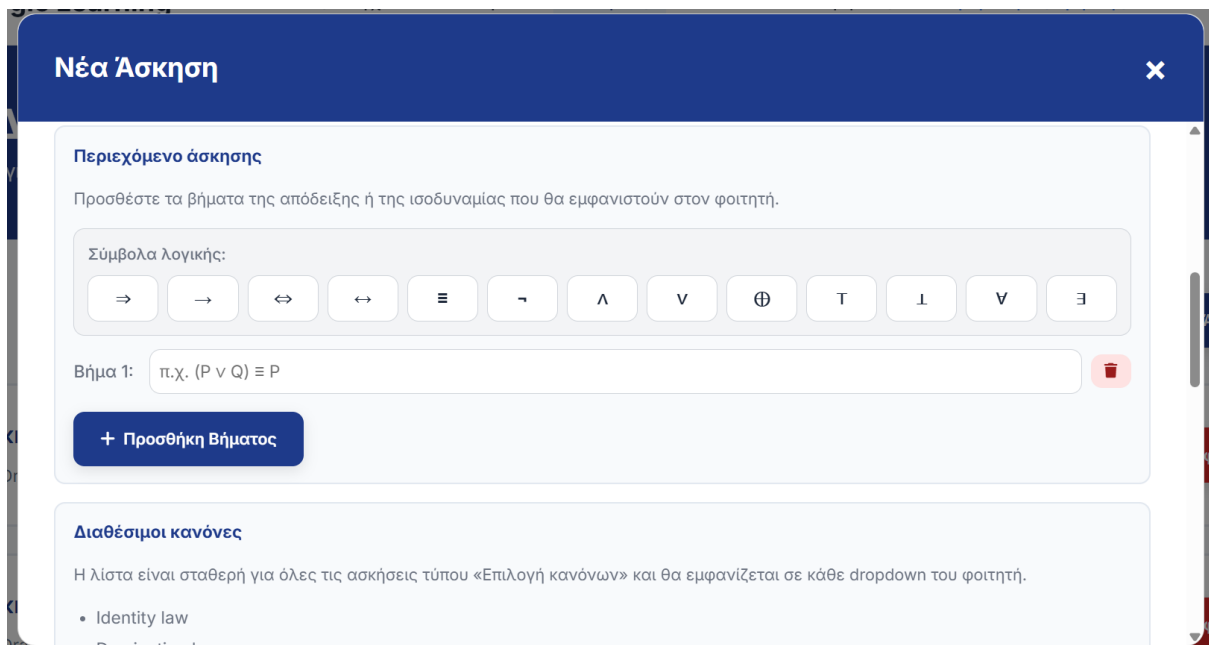
Εικόνα 33: Περιεχόμενο άσκησης εξέτασης ισοδυναμίας

Εικόνα 34: Σωστές απαντήσεις άσκησης εξέτασης ισοδυναμίας

Εικόνα 35: Προεπισκόπηση άσκησης εξέτασης ισοδυναμίας

Ασκήσεις Επιλογής Κανόνων

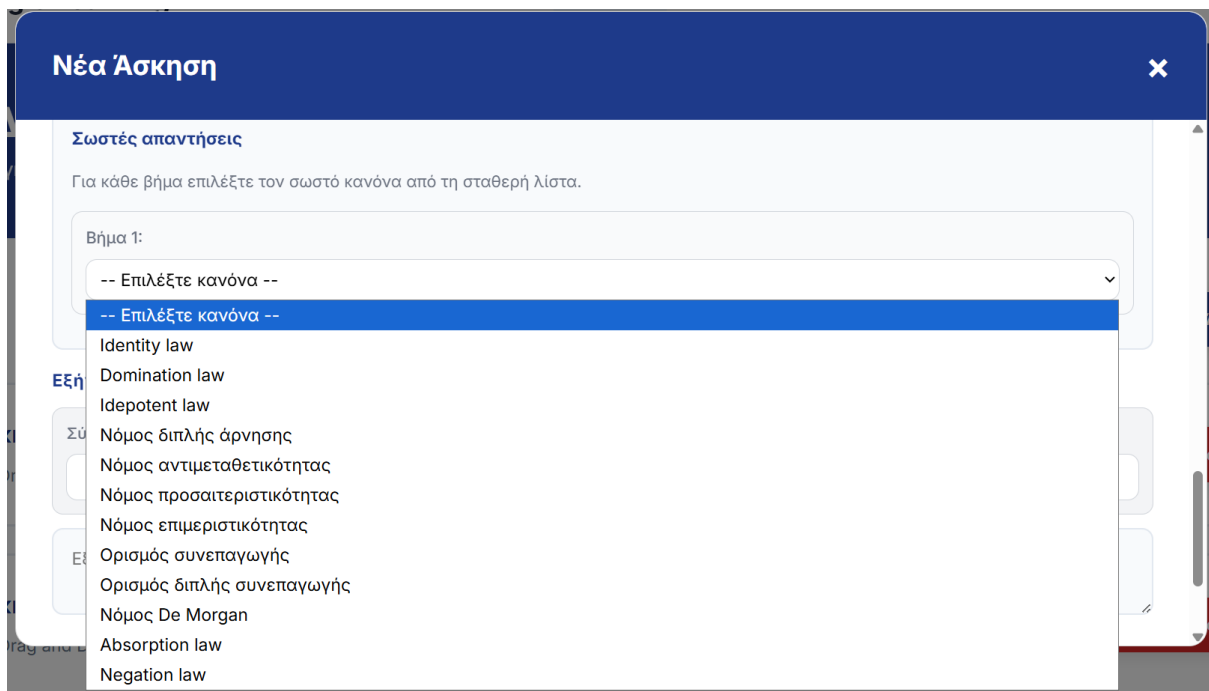
Στις ασκήσεις επιλογής κανόνων, ο διαχειριστής γράφει τα βήματα μιας απόδειξης ή ισοδυναμίας και επιλέγει τον σωστό λογικό κανόνα για κάθε βήμα από την λίστα διαθέσιμων κανόνων.



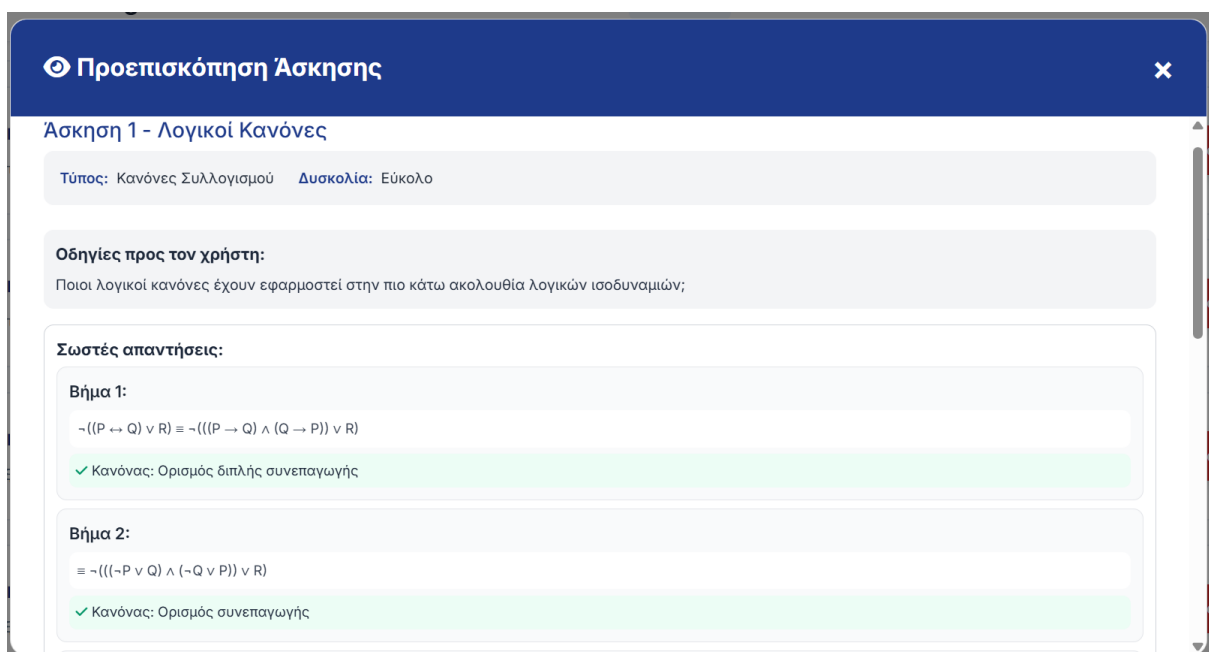
Εικόνα 36: Περιεχόμενο άσκησης επιλογής κανόνων



Εικόνα 37: Λίστα διαθέσιμων κανόνων άσκησης επιλογής κανόνων



Εικόνα 38: Σωστές απαντήσεις άσκησης επιλογής κανόνων



Εικόνα 39: Προεπισκόπηση άσκησης επιλογής κανόνων

Ασκήσεις Ελέγχου Απόδειξης

Στις ασκήσεις ελέγχου απόδειξης, ο διαχειριστής γράφει μία ολοκληρωμένη απόδειξη και καθορίζει την σωστή απάντηση: ναι αν η απόδειξη είναι σωστή ή όχι αν η απόδειξη είναι λανθασμένη.

Νέα Άσκηση

Περιεχόμενο άσκησης

Απόδειξη

Σύμβολα:

⇒ → ↔ ⇔ ≡ ¬ ∧ ∨ ⊕ ⊔ ⊥ ∇ ∃

Γράψτε την απόδειξη με τα βήματά της...

Σωστή απάντηση

☐ Ναι

☐ Όχι

Εικόνα 40: Περιεχόμενο και σωστές απαντήσεις άσκησης ελέγχου απόδειξης

Προεπισκόπηση Άσκησης

Άσκηση 1 - Απόδειξη

Τύπος: Έλεγχος Απόδειξης Δυσκολία: Εύκολο

Οδηγίες προς τον χρήστη:

Είναι η πιο κάτω απόδειξη ορθή;

Απόδειξη:

Θέλουμε να αποδείξουμε ότι $\neg(P \rightarrow Q) \equiv \neg P \wedge \neg Q$

1. $\neg(P \rightarrow Q) \equiv \neg(\neg P \vee Q)$ Ορισμός συνεπαγωγής
 2. $\equiv \neg(\neg P) \wedge \neg Q$ Νόμος De Morgan
 3. $\equiv \neg P \wedge \neg Q$ Νόμος διπλής άρνησης

Σωστή απάντηση:

☒ Ναι

☒ Όχι

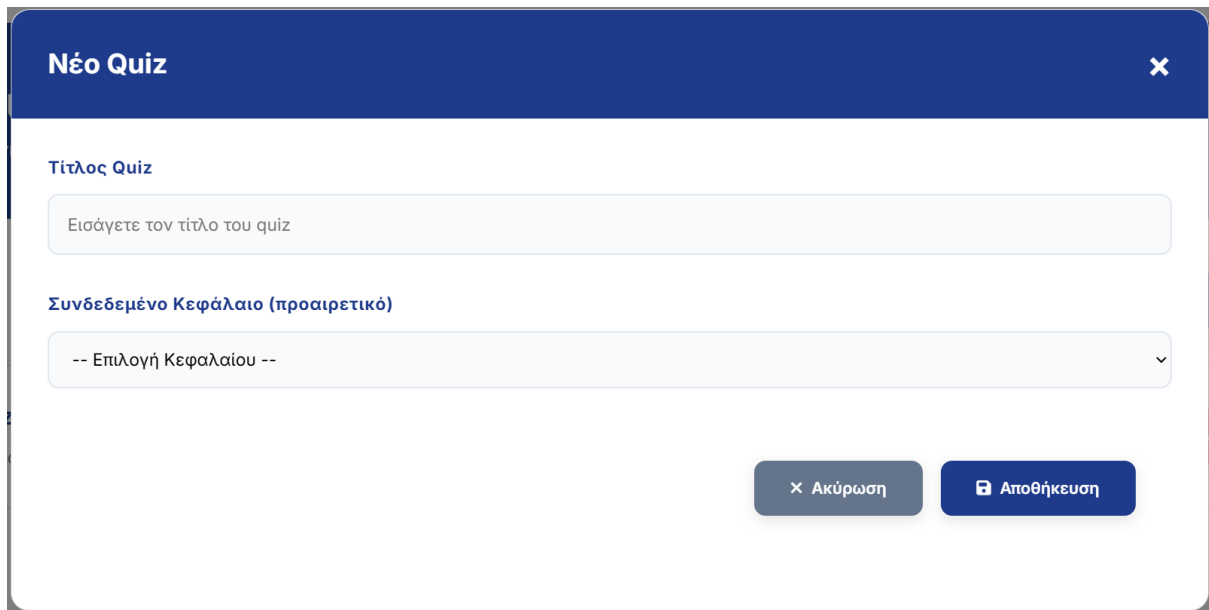
Εικόνα 41: Προεπισκόπηση άσκησης ελέγχου απόδειξης

4.3.3.4 Σελίδα διαχείρισης quiz

Η σελίδα διαχείρισης quiz επιτρέπει στον διαχειριστή να δημιουργεί, να επεξεργάζεται και να οργανώνει quiz αξιολόγησης γνώσεων. Παρέχονται λειτουργίες δημιουργίας νέου quiz, διαχείρισης ερωτήσεων, επιλογής τύπων απαντήσεων και προεπισκόπησης της τελικής του μορφής.

Δημιουργία νέου quiz

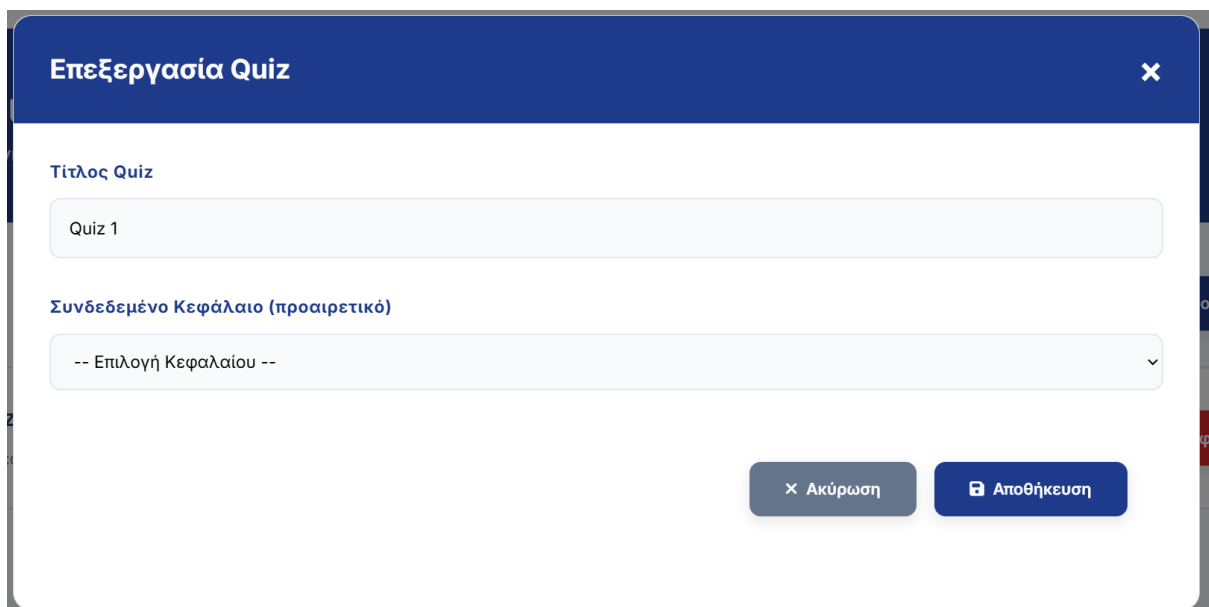
Όταν ο διαχειριστής επιλέξει το κουμπί «+Δημιουργία νέου quiz», εμφανίζεται φόρμα εισαγωγής τίτλου του quiz και συνδεδεμένου κεφαλαίου θεωρίας το οποίο είναι προαιρετικό και επιτρέπει την σύνδεση του quiz με ένα συγκεκριμένο κεφάλαιο.



Εικόνα 42: Δημιουργία νέου quiz

Επεξεργασία quiz

Ο διαχειριστής μπορεί να επεξεργαστεί υπάρχον quiz αλλάζοντας τον τίτλο ή το συνδεδεμένο του κεφάλαιο όπως φαίνεται στην εικόνα.



Εικόνα 43: Επεξεργασία quiz

Διαχείριση ερωτήσεων quiz

Με την επιλογή του κουμπιού «Ερωτήσεις», εμφανίζεται λίστα με τις ήδη αποθηκευμένες ερωτήσεις του συγκεκριμένου quiz. Ο διαχειριστής μπορεί να προσθέσει νέα ερώτηση, να επεξεργαστεί υπάρχουσες ερωτήσεις και να διαγράψει ερωτήσεις.

Διαχείριση Ερωτήσεων

Ερωτήσεις (6) + Προσθήκη Ερώτησης

1 Είναί οι πιο κάτω προτάσεις λογικά ισοδύναμες;
 $(\neg P \vee Q) \rightarrow Q \wedge (Q \rightarrow R) \wedge \neg R$ και $(Q \wedge \neg R) \wedge (\neg R \rightarrow \neg Q) \vee (P \wedge \neg Q)$

Πολλαπλής Επιλογής - 1 σωστή απάντηση

☒ Ναι

☐ Όχι

Εξήγηση:
Πρώτα εξετάζουμε την υποπρόταση: $Q \wedge (Q \rightarrow R)$
Έχουμε:
 $Q \wedge (Q \rightarrow R) \equiv Q \wedge (\neg Q \vee R)$ Ορισμός συνεπαγωγής
 $\equiv (Q \wedge \neg Q) \vee (Q \wedge R)$ Νόμος επιμεριστικότητας
 $\equiv F \vee (Q \wedge R)$ Negation Laws

Εικόνα 44: Modal διαχείρισης ερωτήσεων του quiz

Προσθήκη νέας ερώτησης

Κατά τη δημιουργία νέας ερώτησης, ο διαχειριστής πρέπει να εισάγει το κείμενο της ερώτησης, τον τύπο της ερώτησης, τις επιλογές απαντήσεων, τη σωστή απάντηση και την εξήγηση που εμφανίζεται μετά την ολοκλήρωση.

Το σύστημα υποστηρίζει δύο διαφορετικούς τύπους ερωτήσεων: πολλαπλής επιλογής με μία σωστή απάντηση και πολλαπλής επιλογής με πολλές σωστές απαντήσεις. Η επιλογή τύπου καθορίζει τη μορφή αξιολόγησης της ερώτησης.

Διαχείριση Ερωτήσεων

Κείμενο Ερώτησης

Εισάγετε το κείμενο της ερώτησης...

Τύπος Ερώτησης

-- Επιλέξτε τύπο --

Για True/False, επιλέξτε "Πολλαπλής Επιλογής - 1 σωστή απάντηση" και προσθέστε "Σωστό" και "Λάθος"

Επιλογές Απάντησης

Εισάγετε την επιλογή...

Εικόνα 45: Δημιουργία νέας ερώτησης quiz

Προεπισκόπηση quiz

Παρέχεται λειτουργία προεπισκόπησης, μέσω της οποίας ο διαχειριστής βλέπει την τελική μορφή του quiz, δηλαδή θα βλέπει τις ερωτήσεις και τις απαντήσεις όπως θα τα δει ο χρήστης.

Προεπισκόπηση Quiz

Quiz 1

Κεφάλαιο: -

1. Είναι οι πιο κάτω προτάσεις λογικά ισοδύναμες;
 $(\neg P \vee Q) \rightarrow Q \wedge (Q \rightarrow R) \wedge \neg R$ και $(Q \wedge \neg R) \wedge (\neg R \rightarrow \neg Q) \vee (P \wedge \neg Q)$

Πολλαπλής Επιλογής - 1 σωστή απάντηση

✓ Ναι

Όχι

Εξήγηση

Πρώτα εξετάζουμε την υποπρόταση: $Q \wedge (Q \rightarrow R)$

Έχουμε:

$$Q \wedge (Q \rightarrow R) \equiv Q \wedge (\neg Q \vee R) \text{ Ορισμός συνεπαγωγής}$$

$$\equiv (Q \wedge \neg Q) \vee (Q \wedge R) \text{ Νόμος επιμεριστικότητας}$$

Εικόνα 46: Προεπισκόπηση quiz

4.3.3.5 Σελίδα διαχείρισης χρηστών

Η σελίδα διαχείρισης χρηστών επιτρέπει στον διαχειριστή να παρακολουθεί και να διαχειρίζεται τους εγγεγραμμένους χρήστες της πλατφόρμας. Παρουσιάζεται λίστα όλων των χρηστών μαζί με βασικές πληροφορίες, όπως ονοματεπώνυμο, email, ρόλος χρήστη και ημερομηνία εγγραφής. Επιπλέον, παρέχονται δυνατότητες αναζήτησης και φιλτραρίσματος χρηστών, ώστε να διευκολύνεται η εύρεση συγκεκριμένων χρηστών. Για κάθε χρήστη εμφανίζονται διαθέσιμες ενέργειες, όπως προβολή αποτελεσμάτων και διαγραφή λογαριασμού. Στην περίπτωση του διαχειριστή, ο ίδιος δεν μπορεί να διαγράψει τον εαυτό του.

Ονοματεπώνυμο	Email	Ρόλος	Εγγραφή	Ενέργειες
Natalia Christoforou	nchris23@ucy.ac.cy	Φοιτητής	24/04/2026	Προβολή Αποτελεσμάτων Διαγραφή
Styliana Fesia	sfesia01@ucy.ac.cy	Admin	09/02/2026	Προβολή Αποτελεσμάτων
Administrator	admin@ucy.ac.cy	Admin	09/02/2026	Προβολή Αποτελεσμάτων Διαγραφή

Σύνολο χρηστών: 3

Εικόνα 47: Σελίδα διαχείρισης χρηστών

Με την επιλογή «Προβολή Αποτελεσμάτων», ο διαχειριστής μπορεί να δει τα αποτελέσματα του συγκεκριμένου χρήστη στα quizzes που ολοκλήρωσε. Για κάθε quiz εμφανίζεται η επίδοση του χρήστη σε ποσοστό, καθώς και επιλογές προβολής περισσότερων λεπτομερειών ή επαναφοράς (reset) της προόδου του χρήστη στο συγκεκριμένο quiz.

Η λειτουργία επαναφοράς επιτρέπει στον διαχειριστή να διαγράψει τα αποθηκευμένα αποτελέσματα ενός quiz, δίνοντας τη δυνατότητα στον χρήστη να το επαναλάβει από την αρχή.

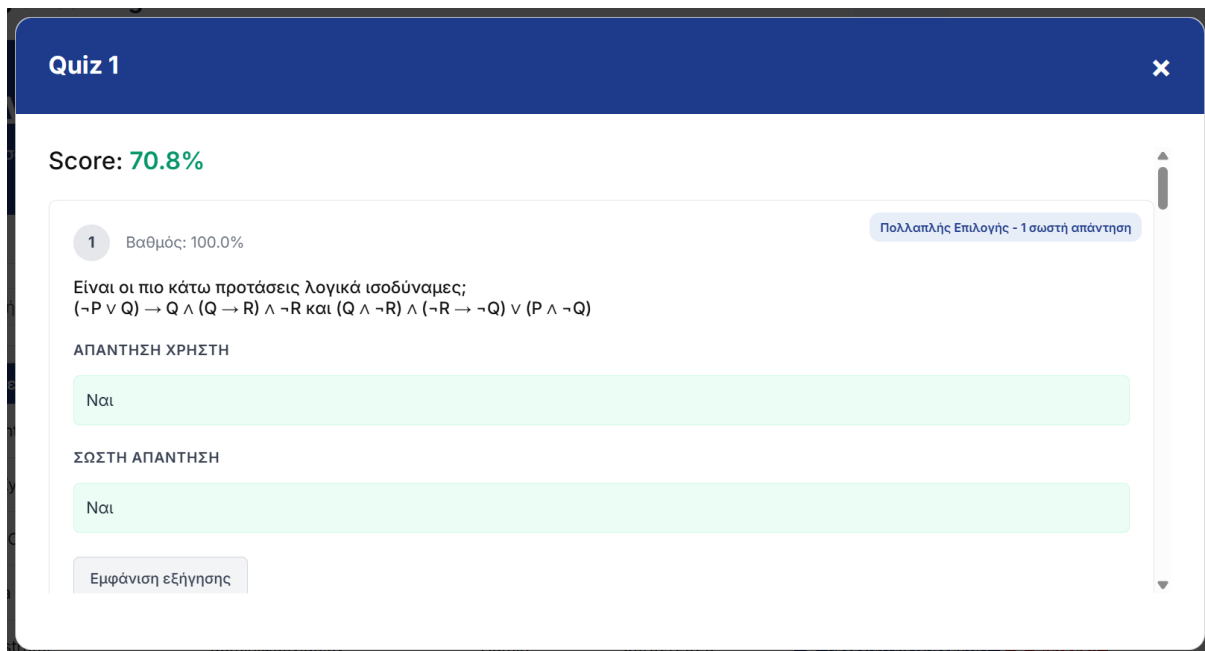
Styliana Fesia

Quizzes

Quiz 1 Score: 70.8%	Λεπτομέρειες Reset
------------------------	--

Εικόνα 48: Προβολή αποτελεσμάτων χρήστη σε quizzes

Με την επιλογή «Λεπτομέρειες», εμφανίζονται αναλυτικότερες πληροφορίες σχετικά με την επίδοση του χρήστη στο επιλεγμένο quiz, όπως οι απαντήσεις που έδωσε, η ορθότητα των επιλογών και η συνολική αξιολόγηση της προσπάθειας. Με αυτόν τον τρόπο, ο διαχειριστής μπορεί να εξετάσει αναλυτικά την πρόοδο και την κατανόηση του χρήστη στις αντίστοιχες ερωτήσεις.



Εικόνα 49: Αναλυτική προβολή αποτελεσμάτων quiz χρήστη

4.3.3.6 Σελίδα Προβολής ως χρήστης

Η λειτουργία «Προβολή ως χρήστης» επιτρέπει στον διαχειριστή να πηγαίνει στο περιβάλλον του φοιτητή και να αλληλεπιδρά με την πλατφόρμα όπως ένας απλός χρήστης. Μέσω της λειτουργίας αυτής, ο διαχειριστής μπορεί να ελέγχει την εμφάνιση της θεωρίας, των ασκήσεων, των quizzes και γενικότερα τη συνολική εμπειρία χρήσης χωρίς να απαιτείται σύνδεση με ξεχωριστό λογαριασμό φοιτητή.

Πιο συγκεκριμένα, παρέχεται η δυνατότητα προβολής των διαθέσιμων ενοτήτων θεωρίας, ασκήσεων και quizzes, καθώς και η επιβεβαίωση της σωστής λειτουργίας στοιχείων όπως η πλοήγηση, η παρουσίαση περιεχομένου, οι οδηγίες, οι εξηγήσεις και τα αποτελέσματα μετά την ολοκλήρωση δραστηριοτήτων.

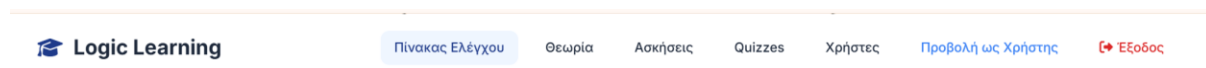
Η λειτουργία αυτή χρησιμοποιείται κυρίως για σκοπούς ελέγχου ποιότητας, καθώς επιτρέπει στον διαχειριστή να επαληθεύει ότι το περιεχόμενο που έχει δημιουργηθεί εμφανίζεται σωστά και ότι η συνολική εμπειρία χρήσης ανταποκρίνεται στις απαιτήσεις της πλατφόρμας.

4.3.4 Πλοήγηση και Εμπειρία Χρήστη

Η πλοήγηση στην πλατφόρμα σχεδιάστηκε με στόχο την απλότητα και τη γρήγορη πρόσβαση στις βασικές λειτουργίες του συστήματος. Οι επιλογές πλοήγησης παραμένουν σταθερές σε όλες τις κύριες σελίδες μέσω ενός κοινού navigation bar, επιτρέποντας στον χρήστη να μεταβαίνει εύκολα μεταξύ του πίνακα ελέγχου, της θεωρίας, των ασκήσεων, των quizzes και των υπόλοιπων λειτουργιών που σχετίζονται με τον ρόλο του.



Εικόνα 50: Navigation bar φοιτητή



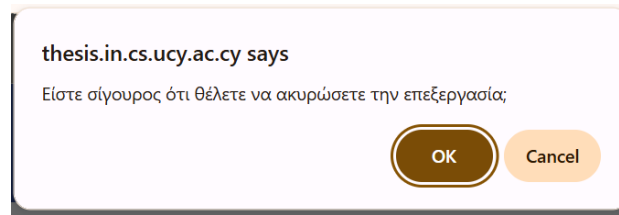
Εικόνα 51: Navigation bar διαχειριστή

Η δομή της εφαρμογής οργανώθηκε με τρόπο ώστε οι πληροφορίες να παρουσιάζονται σταδιακά και με σαφή ιεραρχία. Για παράδειγμα, ο χρήστης επιλέγει αρχικά ενότητα, στη συνέχεια κεφάλαιο και τέλος αποκτά πρόσβαση στο αντίστοιχο περιεχόμενο. Παρόμοια προσέγγιση ακολουθείται και στις ασκήσεις, όπου αρχικά επιλέγεται τύπος άσκησης και έπειτα η διαθέσιμη δραστηριότητα.

Επίσης παρέχεται άμεση ανατροφοδότηση μετά από ενέργειες του χρήστη. Το σύστημα εμφανίζει τα κατάλληλα μηνύματα επιτυχίας ή αποτυχίας σε λειτουργίες όπως αποθήκευση περιεχομένου, δημιουργία ή επεξεργασία ασκήσεων και quizzes, διαγραφή στοιχείων καθώς και ολοκλήρωση δραστηριοτήτων.

Για παράδειγμα, όταν ο διαχειριστής προσπαθεί να κλείσει παράθυρο επεξεργασίας άσκησης, quiz ή θεωρητικού περιεχομένου χωρίς να έχει ολοκληρώσει την αποθήκευση, εμφανίζεται μήνυμα επιβεβαίωσης της ενέργειας. Με αυτόν τον τρόπο, ο χρήστης ενημερώνεται πριν αποχωρήσει από τη φόρμα επεξεργασίας και έχει τη δυνατότητα να ακυρώσει την ενέργεια.

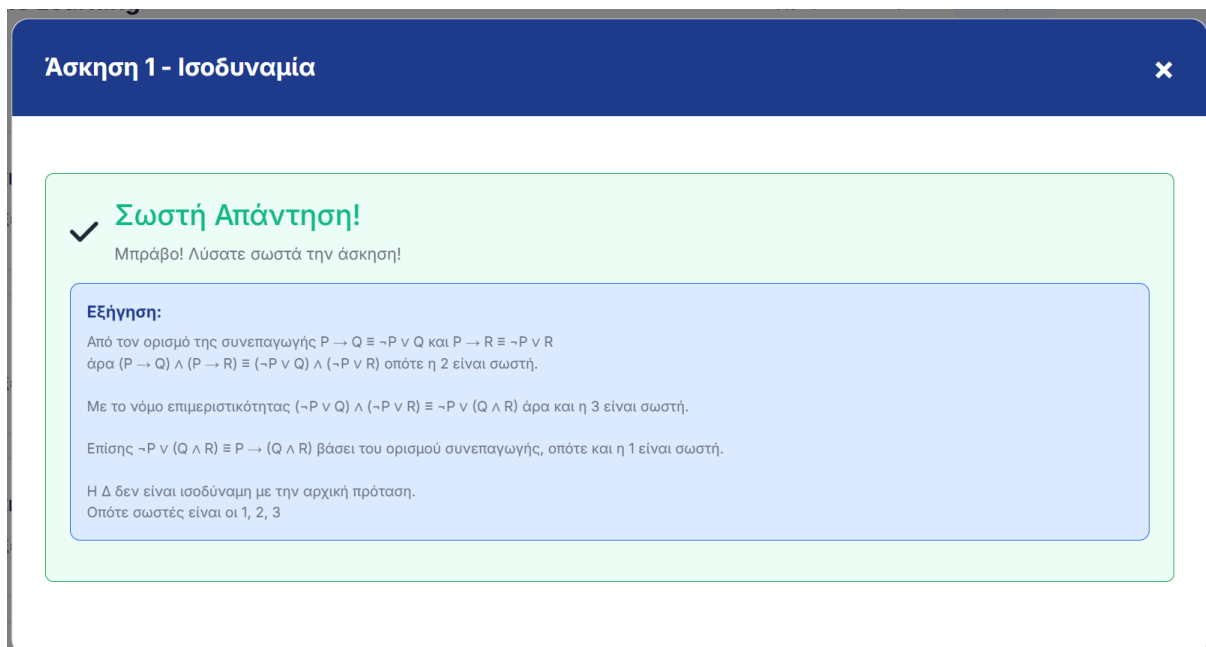
Έτσι, περιορίζεται ο κίνδυνος απώλειας μη αποθηκευμένων αλλαγών και ενισχύεται η ευχρηστία της πλατφόρμας.



Εικόνα 52: Μήνυμα προειδοποίησης

Επιπλέον, μετά την ολοκλήρωση ασκήσεων ή quizzes παρέχονται αποτελέσματα, βαθμολογία, σωστές απαντήσεις και επεξηγήσεις, δίνοντας τη δυνατότητα στον χρήστη να αναγνωρίσει τα λάθη του και να ενισχύσει τη μαθησιακή διαδικασία μέσω ανατροφοδότησης.

Για τις ασκήσεις, η θετική ανατροφοδότηση εμφανίζεται μόνο όταν η άσκηση έχει απαντηθεί πλήρως σωστά. Αν υπάρχει έστω και μία λανθασμένη επιλογή ή αντιστοίχιση, η άσκηση αξιολογείται ως λανθασμένη και εμφανίζεται αντίστοιχο μήνυμα μαζί με επεξήγηση της σωστής λύσης. Η προσέγγιση αυτή ενθαρρύνει την προσεκτική επίλυση των ασκήσεων και βοηθά τον χρήστη να κατανοήσει τα λάθη του.



Εικόνα 53: Παράδειγμα εμφάνισης αποτελεσμάτων και επεξήγησης μετά την ολοκλήρωση άσκησης



Εικόνα 54: Παράδειγμα εμφάνισης ανατροφοδότησης σε λανθασμένη επίλυση άσκησης

Μετά την ολοκλήρωση ενός quiz, εμφανίζεται αναλυτική ανατροφοδότηση σχετικά με την επίδοση του χρήστη. Οι σωστές απαντήσεις επισημαίνονται με πράσινο χρώμα, ενώ οι λανθασμένες εμφανίζονται με κόκκινο. Παράλληλα, σε ερωτήσεις με πολλές σωστές απαντήσεις παρουσιάζονται οι σωστές απαντήσεις που δεν επιλέχθηκαν από τον χρήστη με κίτρινο χρώμα, ώστε να είναι δυνατός ο εντοπισμός παραλείψεων των σωστών απαντήσεων.

Η διάκριση αυτή επιτρέπει στον χρήστη να αναγνωρίζει εύκολα ποια σημεία απάντησε σωστά και σε ποια έκανε λάθος, διευκολύνοντας τη διαδικασία αυτοαξιολόγησης και κατανόησης των σφαλμάτων. Επιπλέον, παρέχεται δυνατότητα προβολής αναλυτικής επεξήγησης, ενισχύοντας την εκπαιδευτική αξία της ανατροφοδότησης. Στο υποκεφάλαιο 4.5 παρουσιάζεται και ο τρόπος βαθμολόγησης αυτού του τύπου ερωτήσεων.

2. Επίλεξε όλες τις προτάσεις που είναι λογικά ισοδύναμες με την πρόταση: $((P \rightarrow Q) \wedge P) \vee ((P \rightarrow R) \wedge P)$
Βαθμός: 0%

Η ΑΠΑΝΤΗΣΗ ΣΟΥ

2. $(P \wedge Q) \vee (P \wedge R)$

3. $P \rightarrow (Q \vee R)$

ΣΩΣΤΕΣ ΑΠΑΝΤΗΣΕΙΣ

1. $P \wedge (Q \vee R)$

2. $(P \wedge Q) \vee (P \wedge R)$

5. $P \wedge ((P \rightarrow Q) \vee (P \rightarrow R))$

ΑΝΑΛΥΣΗ ΑΠΑΝΤΗΣΗΣ

✓ Σωστές απαντήσεις που επέλεξες

2. $(P \wedge Q) \vee (P \wedge R)$

✗ Λάθος απαντήσεις που επέλεξες

3. $P \rightarrow (Q \vee R)$

? Σωστές απαντήσεις που δεν επέλεξες

1. $P \wedge (Q \vee R)$

5. $P \wedge ((P \rightarrow Q) \vee (P \rightarrow R))$

Εμφάνιση εξήγησης

Εικόνα 55: Παράδειγμα αναλυτικής ανατροφοδότησης μετά την ολοκλήρωση quiz

Τέλος, η εμπειρία χρήστη ενισχύεται επίσης μέσω της συνεπούς σχεδίασης όλων των οθονών της εφαρμογής. Η κοινή μορφοποίηση στοιχείων, η χρήση σταθερής χρωματικής παλέτας και η ομοιόμορφη διάταξη των λειτουργιών συμβάλλουν στην ευκολότερη εκμάθηση του συστήματος και στη βελτίωση της συνολικής ευχρηστίας.

4.4 Σχεδίαση Λειτουργιών Ασκήσεων

Η ενότητα των ασκήσεων σχεδιάστηκε με στόχο την πρακτική εξάσκηση του χρήστη στις βασικές έννοιες της Προτασιακής Λογικής. Σε αντίθεση με τα quiz, τα οποία έχουν περισσότερο αξιολογικό χαρακτήρα, οι ασκήσεις χρησιμοποιούνται κυρίως για εξάσκηση και εμπέδωση της ύλης.

Η πλατφόρμα υποστηρίζει τέσσερις τύπους ασκήσεων: Drag & Drop, Επιλογή Κανόνων, Εξέταση Ισοδυναμίας και Έλεγχο Απόδειξης. Κάθε τύπος άσκησης έχει διαφορετική δομή και

διαφορετικό τρόπο αλληλεπίδρασης με τον χρήστη. Για τον λόγο αυτό, το περιεχόμενο κάθε άσκησης και η σωστή απάντηση αποθηκεύονται σε μορφή JSON, ώστε το σύστημα να μπορεί να υποστηρίζει διαφορετικές μορφές δεδομένων με ευέλικτο τρόπο.

Στις ασκήσεις Drag & Drop, ο χρήστης καλείται να αντιστοιχίσει στοιχεία με τις σωστές κατηγορίες. Στις ασκήσεις επιλογής κανόνων, πρέπει να επιλέξει τον κατάλληλο λογικό κανόνα για κάθε βήμα μιας απόδειξης ή ισοδυναμίας. Στις ασκήσεις εξέτασης ισοδυναμίας, ο χρήστης επιλέγει ποιες από τις διαθέσιμες προτάσεις είναι λογικά ισοδύναμες με την αρχική. Τέλος, στις ασκήσεις ελέγχου απόδειξης, καλείται να αξιολογήσει αν μια δοσμένη απόδειξη είναι σωστή ή λανθασμένη.

Ο έλεγχος της απάντησης γίνεται αυτόματα από το σύστημα, με βάση τη σωστή απάντηση που έχει οριστεί από τον διαχειριστή κατά τη δημιουργία της άσκησης. Η αξιολόγηση των ασκήσεων είναι δυαδική, δηλαδή η απάντηση χαρακτηρίζεται είτε ως σωστή είτε ως λανθασμένη. Δεν εφαρμόζεται σύστημα βαθμολόγησης ή ποινής, καθώς οι ασκήσεις έχουν σκοπό την εξάσκηση και όχι την αριθμητική αξιολόγηση του χρήστη.

Μετά την υποβολή της απάντησης, το σύστημα εμφανίζει άμεση ανατροφοδότηση στον χρήστη. Αν η απάντηση είναι σωστή, εμφανίζεται σχετικό μήνυμα επιτυχίας. Αν είναι λανθασμένη, ο χρήστης ενημερώνεται για το αποτέλεσμα και μπορεί να δει την επεξήγηση της σωστής λύσης, εφόσον αυτή έχει καταχωρηθεί από τον διαχειριστή.

4.5 Σχεδίαση Λειτουργιών Quiz

Η λειτουργία των quiz σχεδιάστηκε με στόχο την αξιολόγηση της κατανόησης του θεωρητικού υλικού από τον χρήστη. Κάθε quiz αποτελείται από ένα σύνολο ερωτήσεων, οι οποίες μπορούν να είναι είτε ερωτήσεις πολλαπλής επιλογής με μία σωστή απάντηση είτε ερωτήσεις πολλαπλής επιλογής με περισσότερες από μία σωστές απαντήσεις.

Κατά τη συμμετοχή σε ένα quiz, ο χρήστης απαντά σε όλες τις διαθέσιμες ερωτήσεις και στη συνέχεια υποβάλλει την προσπάθειά του. Μετά την υποβολή, το σύστημα ελέγχει αυτόματα τις απαντήσεις, υπολογίζει τη συνολική βαθμολογία και αποθηκεύει την προσπάθεια στη βάση δεδομένων.

Για τις ερωτήσεις με μία σωστή απάντηση εφαρμόζεται απλός τρόπος βαθμολόγησης. Αν ο χρήστης επιλέξει τη σωστή απάντηση, παίρνει ολόκληρη τη μονάδα της ερώτησης, ενώ αν επιλέξει λανθασμένη απάντηση, παίρνει μηδενική βαθμολογία.

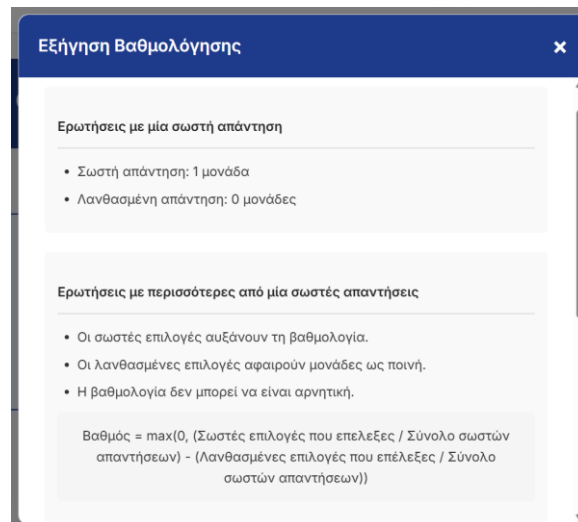
Για τις ερωτήσεις πολλαπλής επιλογής με περισσότερες από μία σωστές απαντήσεις εφαρμόζεται σύστημα ποινής. Η βαθμολογία δεν βασίζεται μόνο στον αριθμό των σωστών επιλογών που επέλεξε ο χρήστης, αλλά λαμβάνει υπόψη και τις λανθασμένες επιλογές. Με αυτόν τον τρόπο ο φοιτητής αποφεύγει να επιλέγει τυχαία απαντήσεις. Η τελική βαθμολογία κάθε τέτοιας ερώτησης περιορίζεται ώστε να μην είναι αρνητική και υπολογίζεται ως εξής:

$$\begin{aligned} \text{Βαθμός} = & \max(0, (\text{Σωστές επιλογές που επέλεξε ο φοιτητής} \\ & / \text{Σύνολο σωστών απαντήσεων}) \\ & - (\text{Λανθασμένες επιλογές που επέλεξε ο φοιτητής} \\ & / \text{Σύνολο σωστών απαντήσεων})) \end{aligned}$$

Επιπλέον, στη σελίδα εκτέλεσης του quiz υπάρχει κουμπί «Εξήγηση Βαθμολόγησης», μέσω του οποίου ο χρήστης μπορεί να ενημερωθεί για τον τρόπο υπολογισμού της βαθμολογίας πριν υποβάλει τις απαντήσεις του. Η εξήγηση παρουσιάζεται σε ξεχωριστό παράθυρο και περιγράφει την βαθμολόγηση των ερωτήσεων με μία σωστή απάντηση, την βαθμολόγηση των ερωτήσεων με περισσότερες από μία σωστές απαντήσεις και την βαθμολόγηση ολόκληρου του quiz. Με αυτόν τον τρόπο, ο χρήστης γνωρίζει από πριν ότι στις ερωτήσεις πολλαπλών σωστών επιλογών οι λανθασμένες επιλογές λειτουργούν ως ποινή.



Εικόνα 56:Κουμπί εξήγησης βαθμολόγησης



Εικόνα 57: Modal εξήγησης τρόπου βαθμολόγησης quiz

Όταν ολοκληρώσει ένα quiz, ο χρήστης μεταφέρεται στη σελίδα αποτελεσμάτων. Εκεί εμφανίζεται η συνολική του βαθμολογία, η βαθμολογία ανά ερώτηση, οι σωστές απαντήσεις και η επεξήγηση. Η ανατροφοδότηση βοηθά τον χρήστη να εντοπίσει τα λάθη του.

Αφού το σύστημα αποθηκεύει την επίδοση κάθε χρήστη στα quizzes, ο φοιτητής μπορεί να βλέπει τα αποτελέσματα του σε συγκεκριμένο quiz, ενώ ο διαχειριστής μπορεί να παρακολουθεί την απόδοση των χρηστών στα ολοκληρωμένα quizzes.

Κεφάλαιο 5

Υλοποίηση Συστήματος

5.1 Τεχνολογίες Υλοποίησης	63
5.2 Οργάνωση και Δομή Κώδικα	65
5.2.1 Δομή Φακέλων Πλατφόρμας	66
5.2.2 Υλοποίηση Controllers και Repositories	67
5.3 Υλοποίηση Συστήματος Σύνδεσης και Ελέγχου Πρόσβασης	70
5.3.1 Εγγραφή Χρήστη	71
5.3.2 Σύνδεση Χρήστη	72
5.3.3 Έλεγχος Ρόλων και Πρόσβασης	73
5.4 Υλοποίηση Διαχείρισης Θεωρίας	74
5.5 Υλοποίηση Ασκήσεων	75
5.5.1 Υλοποίηση Ασκήσεων Drag & Drop	76
5.5.2 Υλοποίηση Ασκήσεων Επιλογής Κανόνων	76
5.5.3 Υλοποίηση Ασκήσεων Εξέτασης Ισοδυναμίας	77
5.5.4 Υλοποίηση Ασκήσεων Ελέγχου Απόδειξης	78
5.6 Υλοποίηση Quizzes	78
5.6.1 Δημιουργία Quiz	79
5.6.2 Δημιουργία και Διαχείριση Ερωτήσεων	79
5.6.3 Υποβολή Απαντήσεων Quiz	79
5.6.4 Υπολογισμός Βαθμολογίας και Penalty System	80
5.7 Υλοποίηση Διαχείρισης Χρηστών	80

5.1 Τεχνολογίες Υλοποίησης

Η ανάπτυξη της πλατφόρμας πραγματοποιήθηκε με τη χρήση τεχνολογιών τόσο για το frontend όσο και για το backend, με στόχο τη δημιουργία ενός λειτουργικού περιβάλλοντος εκμάθησης. [6][7]

PHP [11]

Η πλευρά του backend της πλατφόρμας υλοποιήθηκε με τη γλώσσα προγραμματισμού PHP. Η PHP χρησιμοποιήθηκε για την υλοποίηση των λειτουργιών που εκτελούνται στον server, όπως η σύνδεση των χρηστών, η διαχείριση των sessions και η επικοινωνία με τη βάση δεδομένων. Μέσω της PHP υλοποιήθηκαν επίσης οι βασικές λειτουργίες διαχείρισης της πλατφόρμας, όπως η επεξεργασία θεωρητικού περιεχομένου, ασκήσεων, quizzes και αποτελεσμάτων χρηστών.

JavaScript

Η JavaScript χρησιμοποιήθηκε στο frontend για να γίνουν οι σελίδες πιο δυναμικές και διαδραστικές. Με αυτήν υλοποιήθηκαν λειτουργίες όπως τα modals, οι φόρμες, οι ασκήσεις τύπου drag and drop και η άμεση εμφάνιση αποτελεσμάτων χωρίς να χρειάζεται κάθε φορά ανανέωση της σελίδας. Έτσι, η χρήση της πλατφόρμας γίνεται πιο εύκολη και πιο ομαλή για τον χρήστη.

HTML και CSS

Η HTML χρησιμοποιήθηκε στο frontend για τη δημιουργία της βασικής δομής των σελίδων της εφαρμογής, δηλαδή για την οργάνωση του περιεχομένου, των φορμών, των κουμπιών και γενικά των σελίδων. Η CSS χρησιμοποιήθηκε για τη μορφοποίηση της εμφάνισης της πλατφόρμας, όπως τα χρώματα, οι αποστάσεις μεταξύ των elements, τα cards, τα κουμπιά και η γενική διάταξη των σελίδων. Με αυτόν τον τρόπο δημιουργήθηκε ένα ενιαίο και πιο καθαρό περιβάλλον χρήσης για τις σελίδες θεωρίας, τις ασκήσεις, τα quizzes και το admin panel.

Bootstrap

Το Bootstrap χρησιμοποιήθηκε υποστηρικτικά σε ορισμένα σημεία της διεπαφής χρήστη, κυρίως για βασικά στοιχεία όπως κουμπιά, φόρμες, κάρτες και διάταξη περιεχομένου. Η τελική εμφάνιση της πλατφόρμας βασίστηκε σε μεγάλο βαθμό σε custom CSS, ώστε να προσαρμοστεί στις ανάγκες και στο ύφος της εφαρμογής. Παρ' όλα αυτά, η χρήση του Bootstrap βοήθησε στην ταχύτερη ανάπτυξη κοινών στοιχείων του περιβάλλοντος χρήστη και στη διατήρηση μιας πιο οργανωμένης δομής στις σελίδες. [10] Σε μελλοντική έκδοση, η αξιοποίηση του Bootstrap θα μπορούσε να επεκταθεί περισσότερο, κυρίως ως προς το responsive design.

MySQL

Για την αποθήκευση των δεδομένων της πλατφόρμας χρησιμοποιήθηκε η MySQL. Μέσα στη βάση αποθηκεύονται όλα τα δεδομένα του συστήματος, όπως οι χρήστες, το θεωρητικό υλικό, οι ασκήσεις, τα quizzes, οι απαντήσεις και τα αποτελέσματα των χρηστών. Η MySQL

επιλέχθηκε επειδή συνεργάζεται εύκολα με την PHP και παρέχει έναν αξιόπιστο τρόπο οργάνωσης και διαχείρισης των δεδομένων της πλατφόρμας.

PDO

Η επικοινωνία της εφαρμογής με τη βάση δεδομένων έγινε μέσω του PDO (PHP Data Objects). Το PDO χρησιμοποιήθηκε για την εκτέλεση των SQL queries και την ανάκτηση ή αποθήκευση δεδομένων στη βάση. Ένα βασικό πλεονέκτημα του είναι ότι υποστηρίζει prepared statements, τα οποία βοηθούν στην ασφαλέστερη εκτέλεση των queries και μειώνουν τον κίνδυνο για SQL Injections. Επιπλέον, προσφέρει πιο οργανωμένο τρόπο διαχείρισης της σύνδεσης της PHP με τη βάση δεδομένων. [14]

jQuery

Η βιβλιοθήκη jQuery χρησιμοποιήθηκε σε ορισμένα σημεία της πλατφόρμας, κυρίως στις σελίδες διαχείρισης, για την ευκολότερη υλοποίηση διαδραστικών λειτουργιών όπως φόρμες, modals και δυναμικές αλλαγές περιεχομένου. Επιπλέον, χρησιμοποιείται για τη λειτουργία του Summernote editor, ο οποίος χρησιμοποιείται για την επεξεργασία του θεωρητικού περιεχομένου. [12]

Summernote

Το Summernote χρησιμοποιήθηκε ως επεξεργαστής κειμένου για τη διαχείριση του θεωρητικού υλικού από τον διαχειριστή. Μέσα από αυτό, ο διαχειριστής μπορεί να δημιουργεί και να επεξεργάζεται περιεχόμενο με μορφοποίηση, χωρίς να χρειάζεται να γράφει απευθείας HTML κώδικα. Έτσι, η προσθήκη και ενημέρωση της θεωρίας γίνεται πιο εύκολη και πιο πρακτική. [15]

Font Awesome

Το Font Awesome χρησιμοποιήθηκε για την προσθήκη εικονιδίων σε διάφορα σημεία της πλατφόρμας, όπως σε κουμπιά, κάρτες λειτουργιών και ενέργειες διαχείρισης. Τα εικονίδια βοηθούν τον χρήστη να αναγνωρίζει πιο εύκολα τις λειτουργίες της εφαρμογής και κάνουν το περιβάλλον πιο καθαρό και φιλικό στη χρήση. [13]

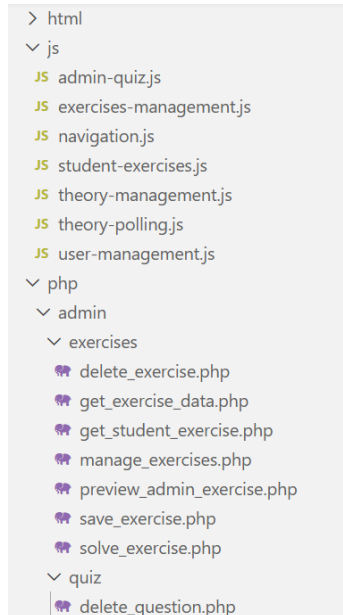
5.2 Οργάνωση και Δομή Κώδικα

Η οργάνωση του κώδικα της πλατφόρμας έγινε με στόχο να υπάρχει καθαρή δομή και να είναι πιο εύκολη η συντήρηση και η μελλοντική επέκταση της εφαρμογής. Για τον λόγο αυτό, ο

κώδικας χωρίστηκε σε διαφορετικούς φακέλους, ανάλογα με τον ρόλο κάθε αρχείου μέσα στο σύστημα.

5.2.1 Δομή Φακέλων Πλατφόρμας

Η εφαρμογή οργανώθηκε σε ξεχωριστούς φακέλους, ώστε τα αρχεία που εξυπηρετούν παρόμοιες λειτουργίες να βρίσκονται συγκεντρωμένα στο ίδιο σημείο.



Εικόνα 58: Παράδειγμα δομής φακέλων.

Οι βασικοί φάκελοι της είναι πλατφόρμας είναι οι εξής:

controllers/

Περιλαμβάνει τους controllers της εφαρμογής, οι οποίοι διαχειρίζονται τις ενέργειες του χρήστη και συντονίζουν την επικοινωνία μεταξύ διεπαφής και δεδομένων.

repositories/

Περιλαμβάνει τις κλάσεις που επικοινωνούν με τη βάση δεδομένων και εκτελούν τα απαραίτητα SQL queries και επιστρέφουν δεδομένα στους controllers.

admin/

Περιλαμβάνει τις σελίδες και τις λειτουργίες που αφορούν τον διαχειριστή.

user/

Περιλαμβάνει τις λειτουργίες που σχετίζονται με το περιβάλλον του φοιτητή, όπως θεωρία, ασκήσεις και quizzes.

html/

Περιλαμβάνει τις βασικές στατικές σελίδες εισόδου, όπως η αρχική σελίδα, η σελίδα σύνδεσης και η σελίδα εγγραφής. Οι σελίδες αυτές αποτελούν το αρχικό σημείο πρόσβασης του χρήστη στην πλατφόρμα πριν μεταφερθεί στο περιβάλλον φοιτητή ή διαχειριστή.

css/

Περιλαμβάνει τα αρχεία μορφοποίησης της πλατφόρμας όπως χρώματα, αποστάσεις, περιγράμματα.

js/

Περιλαμβάνει τα αρχεία JavaScript που χρησιμοποιούνται για δυναμικές λειτουργίες και αλληλεπιδράσεις.

config/

Περιλαμβάνει αρχεία ρυθμίσεων, όπως η σύνδεση με τη βάση δεδομένων.

5.2.2 Υλοποίηση Controllers και Repositories

Στο πλαίσιο της αρχιτεκτονικής MVC που χρησιμοποιείται στην πλατφόρμα, οι controllers αποτελούν το ενδιάμεσο επίπεδο μεταξύ της διεπαφής χρήστη και του επιπέδου πρόσβασης στα δεδομένα. Οι controllers είναι υπεύθυνοι για τη διαχείριση της λογικής της εφαρμογής, όπως η επεξεργασία δεδομένων, ο έλεγχος εγκυρότητας, η κλήση κατάλληλων repositories και ο συντονισμός των λειτουργιών του συστήματος.

TheoryController

Ο TheoryController διαχειρίζεται τη λογική που σχετίζεται με το θεωρητικό περιεχόμενο της πλατφόρμας. Είναι υπεύθυνος για την ανάκτηση ενοτήτων και κεφαλαίων θεωρίας, καθώς και για τη δημιουργία, επεξεργασία και διαγραφή τους από το περιβάλλον διαχείρισης.

Για παράδειγμα, οι συναρτήσεις getAllUnits(), getChaptersByUnit(), getChapterById() και getUnitById() χρησιμοποιούνται για τη φόρτωση του θεωρητικού περιεχομένου, ενώ οι saveUnit(), saveChapter(), deleteUnit() και deleteChapter() υποστηρίζουν τις λειτουργίες διαχείρισης θεωρίας. Επιπλέον, χρησιμοποιούνται συναρτήσεις ελέγχου εγκυρότητας, όπως οι

`validateUnitData()` και `validateChapterData()`, ώστε να διασφαλίζεται η ορθότητα των δεδομένων πριν την αποθήκευσή τους.

ExerciseController

Ο `ExerciseController` διαχειρίζεται τη λογική των ασκήσεων. Χρησιμοποιείται για την ανάκτηση ασκήσεων, τη φόρτωσή τους για επίλυση, τη δημιουργία, την επεξεργασία και τη διαγραφή τους. Οι συναρτήσεις `getExercisesForStudent()`, `getExerciseForSolving()` και `getExercisesByType()` χρησιμοποιούνται για την προβολή των ασκήσεων, ενώ οι `createExercise()`, `updateExercise()` και `deleteExercise()` χρησιμοποιούνται για τη διαχείριση ασκήσεων από τον admin. Ο controller κάνει validation στα βασικά πεδία της άσκησης, όπως τίτλος, τύπος και περιεχόμενο, και επιτρέπει συγκεκριμένους τύπους ασκήσεων όπως `drag_drop`, `logical_equivalence`, `logic_rules` και `proof_check`.

QuizController

Ο `QuizController` περιλαμβάνει τη βασική λογική που αφορά τα quizzes της πλατφόρμας. Μέσω αυτού γίνεται η φόρτωση των διαθέσιμων quizzes, η διαχείριση των ερωτήσεων, η υποβολή των απαντήσεων και ο υπολογισμός της βαθμολογίας των χρηστών. Επιπλέον, χρησιμοποιείται και στις λειτουργίες του διαχειριστή για τη δημιουργία και επεξεργασία quizzes και ερωτήσεων.

Μία από τις σημαντικότερες συναρτήσεις του controller είναι η `calculateResult()`, η οποία υπολογίζει τη βαθμολογία του quiz. Στις ερωτήσεις πολλαπλών σωστών απαντήσεων εφαρμόζεται σύστημα ποινής, μειώνοντας τη βαθμολογία όταν ο χρήστης επιλέγει λανθασμένες απαντήσεις. Επιπλέον, η συνάρτηση `saveAttempt()` χρησιμοποιείται για την αποθήκευση της προσπάθειας και των απαντήσεων του χρήστη στη βάση δεδομένων μέσω του αντίστοιχου repository.

Repositories

Τα repositories αποτελούν το επίπεδο που είναι υπεύθυνο για την επικοινωνία της πλατφόρμας με τη βάση δεδομένων. Μέσα σε αυτά συγκεντρώνονται τα SQL ερωτήματα και οι βασικές λειτουργίες ανάκτησης, εισαγωγής, ενημέρωσης και διαγραφής δεδομένων. Με αυτόν τον τρόπο, οι controllers δεν περιέχουν απευθείας SQL κώδικα, αλλά καλούν τις κατάλληλες μεθόδους των repositories για πιο ασφαλή πρόσβαση στην βάση.

Στην πλατφόρμα χρησιμοποιείται επίσης το `BaseRepository`, το οποίο λειτουργεί ως κοινή βάση για τα υπόλοιπα repositories. Μέσω αυτού αξιοποιείται η σύνδεση με τη βάση δεδομένων

και αποφεύγεται η επανάληψη κοινής λειτουργικότητας. Επιπλέον παρέχει γενικές CRUD λειτουργίες όπως `count()` για να μετρά πόσες εγγραφές υπάρχουν στον πίνακα, `create()` για να δημιουργεί νέα εγγραφή, `update()` για να ενημερώνει μια εγγραφή με βάση το `id` και `delete()` για να διαγράφει μια εγγραφή με βάση το `id`.

UserRepository

Το `UserRepository` είναι υπεύθυνο για τις λειτουργίες που αφορούν τους χρήστες της πλατφόρμας. Χρησιμοποιείται κυρίως στο περιβάλλον διαχείρισης για την προβολή, αναζήτηση και φιλτράρισμα χρηστών. Η συνάρτηση `getAllUsers()` επιστρέφει χρήστες με βάση όνομα/email και ρόλο, ενώ αποκλείει χρήστες που έχουν διαγραφεί μέσω του πεδίου `is_deleted`. Η `getUserProfile()` επιστρέφει τα στοιχεία ενός χρήστη μαζί με τις προσπάθειές του στα quiz. Επιπλέον υλοποιείται και η συνάρτηση διαγραφής ενός χρήστη.

QuizRepository

Το `QuizRepository` συγκεντρώνει όλες τις λειτουργίες πρόσβασης στη βάση για τα quiz. Χρησιμοποιείται για τη φόρτωση quiz μαζί με τις ερωτήσεις και τις επιλογές τους μέσω της `getQuizWithQuestions()`, για τη δημιουργία και ενημέρωση quiz, καθώς και για τη διαγραφή quiz μαζί με τα `questions` και τα `choices` του. Επίσης, διαχειρίζεται την αποθήκευση και ενημέρωση ερωτήσεων μέσω της `saveQuestion()`, όπου αποθηκεύονται τα δεδομένα της ερώτησης και οι επιλογές απάντησης. Περιλαμβάνει επίσης λειτουργίες για την ανάκτηση των αναλυτικών αποτελεσμάτων προσπάθειας quiz και για το `reset` μιας προσπάθειας χρήστη.

TheoryRepository

Το `TheoryRepository` είναι υπεύθυνο για την πρόσβαση στα δεδομένα της θεωρίας. Διαχειρίζεται τις ενότητες θεωρίας και τα κεφάλαια. Μέσω της `getAllUnitsWithChapters()` φορτώνει όλες τις ενότητες μαζί με τα κεφάλαια τους. Έχει επίσης μεθόδους για δημιουργία, ενημέρωση και διαγραφή ενοτήτων και κεφαλαίων, όπως `createUnit()`, `updateUnit()`, `deleteUnit()`, `createChapter()`, `updateChapter()` και `deleteChapter()`. Χρησιμοποιείται κυρίως στο admin περιβάλλον για τη διαχείριση του θεωρητικού περιεχομένου.

ExerciseRepository

Το `ExerciseRepository` διαχειρίζεται τα δεδομένα των ασκήσεων από την πλευρά του διαχειριστή. Περιλαμβάνει λειτουργίες για δημιουργία, ενημέρωση και διαγραφή ασκήσεων, καθώς και για αποθήκευση του περιεχομένου και της σωστής απάντησης κάθε άσκησης.

Επειδή οι ασκήσεις της πλατφόρμας έχουν διαφορετικούς τύπους, τα πεδία `content` και `correct_answer` αποθηκεύονται σε μορφή JSON στον πίνακα `exercises`. Το `ExerciseRepository`

αποθηκεύει αυτές τις JSON τιμές μέσω των συναρτήσεων `createExercise()` και `updateExercise()`.

UserExerciseRepository

Το `UserExerciseRepository` χρησιμοποιείται για την ανάκτηση ασκήσεων από την πλευρά του φοιτητή. Σε αντίθεση με το `ExerciseRepository`, που καλύπτει γενικές και admin λειτουργίες CRUD, το `UserExerciseRepository` εστιάζει στην προβολή των διαθέσιμων ασκήσεων για επίλυση με την μέθοδο `getAllExercisesForStudent()`.

UserTheoryRepository

Το `UserTheoryRepository` χρησιμοποιείται για την προβολή της θεωρίας από την πλευρά του φοιτητή. Η βασική του μέθοδος είναι η `getAllUnitsWithChapters()`, η οποία φορτώνει τις ενότητες και τα κεφάλαια θεωρίας. Τα δεδομένα επιστρέφονται σε δομή όπου κάθε ενότητα περιέχει τα αντίστοιχα κεφάλαια της, ώστε να μπορούν να εμφανιστούν στο sidebar και στο περιεχόμενο της student theory page.

UserQuizRepository

Το `UserQuizRepository` διαχειρίζεται τα quiz από την πλευρά του φοιτητή. Η συνάρτηση `getAllQuizzesForStudent()` επιστρέφει τα διαθέσιμα quiz μαζί με το πλήθος των ερωτήσεων τους, ώστε να εμφανίζονται στη σελίδα του φοιτητή. Η `getStudentAttemptHistory()` ανακτά το ιστορικό προσπαθειών του χρήστη, για να ελεγχθεί αν ο φοιτητής έχει ήδη ολοκληρώσει ένα quiz, ώστε να εμφανιστεί η ένδειξη «Ολοκληρωμένο» στη σελίδα των quizzes. Κατά την ολοκλήρωση ενός quiz, οι συναρτήσεις `saveAttempt()` και `saveAnswer()` χρησιμοποιούνται για την αποθήκευση της συνολικής προσπάθειας και των επιμέρους απαντήσεων στη βάση δεδομένων.

5.3 Υλοποίηση Συστήματος Σύνδεσης και Ελέγχου Πρόσβασης

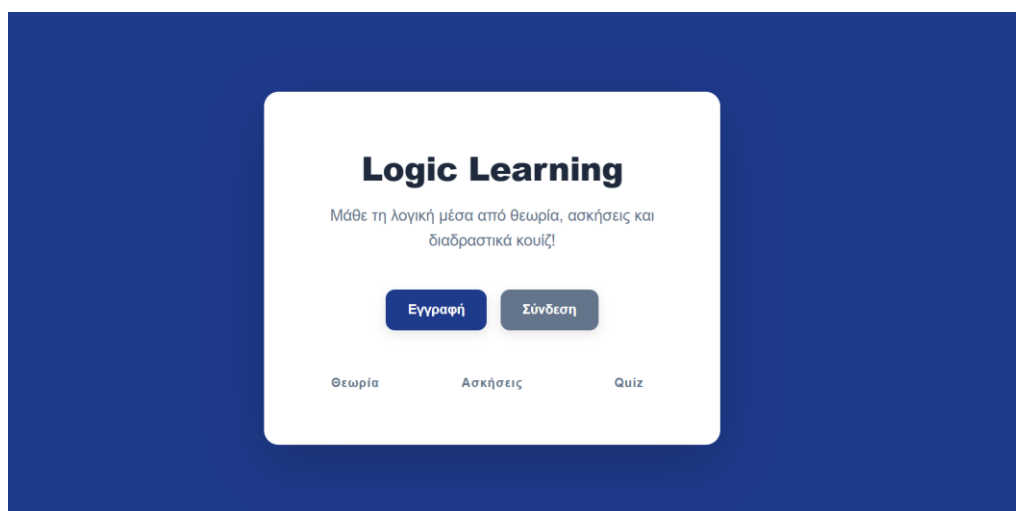
Η πλατφόρμα χρησιμοποιεί μηχανισμό σύνδεσης χρηστών βασισμένο σε PHP sessions. Η εγγραφή και η σύνδεση υλοποιούνται μέσω των αρχείων `process_register.php` και `process_login.php`, τα οποία επικοινωνούν με τη βάση δεδομένων με χρήση PDO.

Για λόγους ασφάλειας, οι κωδικοί πρόσβασης δεν αποθηκεύονται στη βάση σε απλή μορφή. Κατά την εγγραφή, ο κωδικός μετατρέπεται σε hash με τη συνάρτηση `password_hash()`, ενώ κατά τη σύνδεση η επαλήθευση του κωδικού γίνεται με τη συνάρτηση `password_verify()`. Με

αυτό τον τρόπο ακόμη και αν αποκτηθεί πρόσβαση στη βάση δεδομένων, οι πραγματικοί κωδικοί των χρηστών δεν είναι άμεσα αναγνώσιμοι.

Ο έλεγχος πρόσβασης βασίζεται στον ρόλο του χρήστη, ο οποίος αποθηκεύεται στον πίνακα users, στο πεδίο role. Η εφαρμογή υποστηρίζει δύο ρόλους, user και admin, ώστε να διαχωρίζονται οι λειτουργίες του φοιτητή από τις λειτουργίες του διαχειριστή. Μετά από μια επιτυχημένη σύνδεση, τα βασικά στοιχεία του χρήστη αποθηκεύονται στο \$_SESSION, έτσι ώστε το σύστημα να μπορεί να αναγνωρίζει τον συνδεδεμένο χρήστη και να ελέγχει την πρόσβαση του σε άλλες σελίδες.

Από την αρχική σελίδα index.html ο χρήστης μπορεί να εγγραφεί ή να συνδεθεί στο σύστημα αν έχει ήδη εγγραφεί.



Εικόνα 59: Αρχική σελίδα της πλατφόρμας index.html.

5.3.1 Εγγραφή Χρήστη

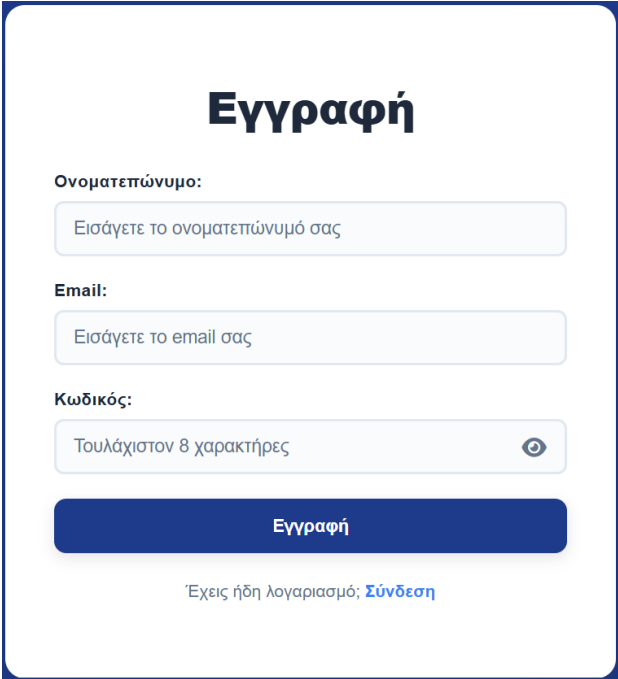
Η εγγραφή νέου χρήστη ξεκινά από τη φόρμα register.html, όπου ο χρήστης συμπληρώνει το ονοματεπώνυμο, το email και τον κωδικό πρόσβασης. Με την υποβολή της φόρμας, τα δεδομένα αποστέλλονται στο αρχείο process_register.php, όπου πραγματοποιούνται οι απαραίτητοι έλεγχοι αν είναι έγκυρα τα δεδομένα.

Συγκεκριμένα, ελέγχεται αν έχουν συμπληρωθεί όλα τα απαιτούμενα πεδία, αν το ονοματεπώνυμο και το email έχουν έγκυρη μορφή και αν ο κωδικός πρόσβασης έχει μήκος ίσο ή μεγαλύτερο από 8 χαρακτήρες. Οι έλεγχοι αυτοί πραγματοποιούνται μέσω συναρτήσεων

όπως οι `validateRequired()`, `validateName()`, `validateEmail()` και `validatePassword()`, οι οποίες ορίζονται στο αρχείο `validation.php`.

Πριν από τη δημιουργία του λογαριασμού, το σύστημα ελέγχει αν υπάρχει ήδη χρήστης με το ίδιο email. Αν το email δεν χρησιμοποιείται, ο κωδικός μετατρέπεται σε hash με τη συνάρτηση `password_hash()` και ο χρήστης αποθηκεύεται στον πίνακα `users` με προεπιλεγμένο ρόλο `ROLE_USER` δηλαδή φοιτητή.

Με την επιτυχημένη εγγραφή δημιουργείται ο νέος χρήστης στη βάση δεδομένων και εκκινείται session, ώστε το σύστημα να αναγνωρίζει ότι ο χρήστης είναι πλέον συνδεδεμένος.



The image shows a registration form titled "Εγγραφή" (Registration). It contains three input fields: "Ονοματεπώνυμο:" (Full Name) with placeholder text "Εισάγετε το ονοματεπώνυμό σας", "Email:" with placeholder text "Εισάγετε το email σας", and "Κωδικός:" (Password) with placeholder text "Τουλάχιστον 8 χαρακτήρες" and a toggle icon. Below the fields is a blue button labeled "Εγγραφή" (Register). At the bottom, there is a link: "Έχεις ήδη λογαριασμό; [Σύνδεση](#)" (Already have an account? [Login](#)).

Εικόνα 60: Φόρμα εγγραφής χρήστη

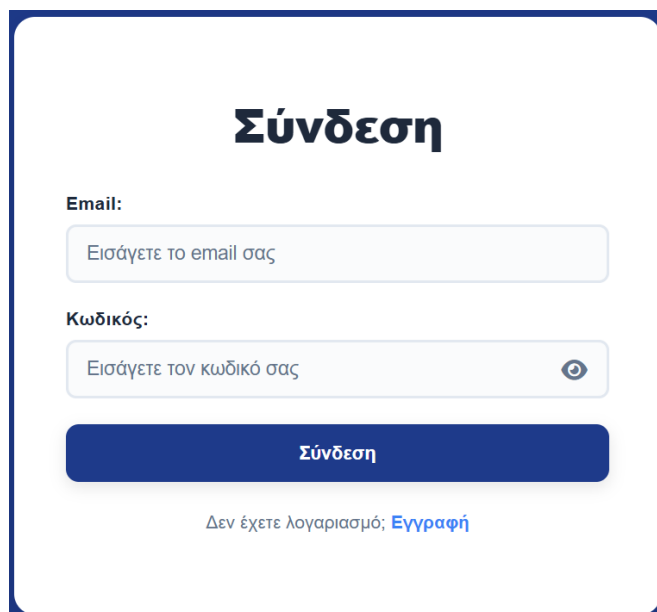
5.3.2 Σύνδεση Χρήστη

Η σύνδεση χρήστη πραγματοποιείται μέσω της φόρμας `login.html`, η οποία αποστέλλει τα στοιχεία σύνδεσης στο αρχείο `process_login.php`. Ο χρήστης εισάγει το email και τον κωδικό πρόσβασής του και το σύστημα ελέγχει αρχικά αν τα πεδία είναι έγκυρα.

Στη συνέχεια, γίνεται αναζήτηση του χρήστη στη βάση δεδομένων με βάση το email. Αν βρεθεί αντίστοιχος λογαριασμός, η επαλήθευση του κωδικού γίνεται με τη συνάρτηση `password_verify()`, η οποία συγκρίνει τον κωδικό που έβαλε ο χρήστης με το αποθηκευμένο hash.

Μετά την επιτυχημένη σύνδεση, αποθηκεύονται στο session τα βασικά στοιχεία του χρήστη. Αν ο χρήστης έχει ρόλο admin, μεταφέρεται στο admin dashboard, ενώ αν έχει ρόλο user, μεταφέρεται στο dashboard του φοιτητή.

Η αποσύνδεση υλοποιείται μέσω του αρχείου logout.php, το οποίο καταστρέφει το ενεργό session και επιστρέφει τον χρήστη στην αρχική σελίδα index.html.

The image shows a login form with a dark blue border. At the top, the title 'Σύνδεση' is centered in a bold, dark blue font. Below the title, there are two input fields. The first is labeled 'Email:' and contains the placeholder text 'Εισάγετε το email σας'. The second is labeled 'Κωδικός:' and contains the placeholder text 'Εισάγετε τον κωδικό σας', with a small eye icon to its right. Below these fields is a large, dark blue button with the text 'Σύνδεση' in white. At the bottom, there is a link that says 'Δεν έχετε λογαριασμό; Εγγραφή'.

Εικόνα 61: Φόρμα σύνδεσης χρήστη

5.3.3 Έλεγχος ρόλων και πρόσβασης

Ο έλεγχος πρόσβασης υλοποιείται με βάση τον ρόλο που είναι αποθηκευμένος στο session του χρήστη. Οι διαθέσιμοι ρόλοι ορίζονται ως ROLE_USER για τον φοιτητή και ROLE_ADMIN για τον διαχειριστή. Με αυτόν τον τρόπο, το σύστημα μπορεί να αναγνωρίζει αν ο συνδεδεμένος χρήστης είναι φοιτητής ή διαχειριστής.

Για τον έλεγχο πρόσβασης χρησιμοποιούνται βοηθητικές συναρτήσεις στο αρχείο config.php, όπως οι isAdmin(), isUser(), requireAdmin() και requireUser(). Η συνάρτηση requireAdmin() χρησιμοποιείται στις σελίδες διαχείρισης, ώστε να επιτρέπεται πρόσβαση μόνο σε χρήστες με ρόλο διαχειριστή. Αντίστοιχα, η requireUser() χρησιμοποιείται στις σελίδες που απαιτούν σύνδεση απλού χρήστη.

Η χρήση των συναρτήσεων αυτών διασφαλίζει ότι οι φοιτητές έχουν πρόσβαση μόνο στις λειτουργίες φοιτητή, όπως θεωρία, ασκήσεις και quizzes, ενώ οι διαχειριστές έχουν πρόσβαση στις λειτουργίες διαχείρισης περιεχομένου και χρηστών.

5.4 Υλοποίηση Διαχείρισης Θεωρίας

Η θεωρία στην πλατφόρμα υλοποιήθηκε με τρόπο ώστε να υποστηρίζεται τόσο η διαχείριση της από τον διαχειριστή όσο και η προβολή της από τον φοιτητή.

Το θεωρητικό υλικό οργανώνεται σε δύο επίπεδα: ενότητες και κεφάλαια. Κάθε ενότητα μπορεί να περιλαμβάνει πολλά κεφάλαια, ενώ κάθε κεφάλαιο ανήκει σε μία συγκεκριμένη ενότητα. Η δομή αυτή βοηθά στην καλύτερη οργάνωση του περιεχομένου και κάνει την πλοήγηση πιο απλή για τον χρήστη.

Στην υλοποίηση της θεωρίας χρησιμοποιούνται κυρίως τα αρχεία `manage_theory.php`, `save_unit.php`, `save_chapter.php`, `delete_unit.php`, `delete_chapter.php`, η σελίδα προβολής θεωρίας του φοιτητή, τα repositories `TheoryRepository.php` και `UserTheoryRepository.php`, καθώς και τα JavaScript αρχεία `theory-management.js` και `theory-polling.js`.

Η σελίδα διαχείρισης της θεωρίας είναι διαθέσιμη μόνο σε χρήστες με ρόλο διαχειριστή. Κατά τη φόρτωση της σελίδας, ανακτώνται οι υπάρχουσες ενότητες και τα κεφάλαια θεωρίας μέσω του `TheoryRepository`, ώστε ο διαχειριστής να μπορεί να δει και να επεξεργαστεί το ήδη καταχωρημένο υλικό.

Η δημιουργία νέας ενότητας γίνεται μέσω ενός modal φόρμας, όπου ο διαχειριστής συμπληρώνει τον τίτλο και την περιγραφή της ενότητας. Μετά την υποβολή, τα δεδομένα αποστέλλονται στο αρχείο `save_unit.php`, το οποίο πραγματοποιεί τους απαραίτητους ελέγχους και καλεί το `TheoryRepository` για την αποθήκευση της ενότητας στη βάση δεδομένων.

Αντίστοιχα, η δημιουργία κεφαλαίου πραγματοποιείται μέσω φόρμας στην οποία ο διαχειριστής επιλέγει την ενότητα στην οποία θα ανήκει το κεφάλαιο και συμπληρώνει τον τίτλο και το περιεχόμενο του. Η αποθήκευση ή ενημέρωση των κεφαλαίων γίνεται μέσω του αρχείου `save_chapter.php`. Για την επεξεργασία του περιεχομένου των κεφαλαίων χρησιμοποιείται το Summernote editor, όπου επιτρέπει την μορφοποίηση του κειμένου.

Η επεξεργασία ενοτήτων και κεφαλαίων γίνεται από τα ίδια modal που χρησιμοποιούνται και για τη δημιουργία. Όταν ο διαχειριστής επιλέξει επεξεργασία, τα υπάρχοντα δεδομένα εμφανίζονται στη φόρμα και μετά την αποθήκευση ενημερώνονται οι αντίστοιχες εγγραφές στη βάση. Η διαγραφή ενοτήτων και κεφαλαίων γίνεται μέσω των αρχείων delete_unit.php και delete_chapter.php. Στην περίπτωση διαγραφής ενότητας, διαγράφονται και τα κεφάλαια που ανήκουν σε αυτή, ώστε να διατηρείται η συνέπεια των δεδομένων.

Μετά τη δημιουργία ή ενημέρωση περιεχομένου από τον διαχειριστή, οι ενεργές ενότητες και τα κεφάλαια εμφανίζονται αυτόματα στο περιβάλλον του φοιτητή μέσω του UserTheoryRepository.

5.5 Υλοποίηση Ασκήσεων

Η υλοποίηση των ασκήσεων ακολουθεί κοινή αρχιτεκτονική ανεξάρτητα από τον τύπο της άσκησης. Κάθε άσκηση αποθηκεύεται στη βάση, εμφανίζεται δυναμικά στον φοιτητή, ελέγχεται μετά την υποβολή της απάντησης και επιστρέφει την κατάλληλη ανατροφοδότηση.

Για να υποστηρίζονται ασκήσεις με διαφορετική δομή περιεχομένου, το σύστημα αποθηκεύει τα δεδομένα κάθε άσκησης σε μορφή JSON στα πεδία content και correct_answer του πίνακα exercises. Με αυτό τον τρόπο, κάθε τύπος άσκησης μπορεί να έχει τη δική του δομή περιεχομένου, χωρίς να χρειάζεται να δημιουργηθούν ξεχωριστοί πίνακες στη βάση δεδομένων.

Κατά τη δημιουργία μιας άσκησης, ο διαχειριστής συμπληρώνει το περιεχόμενο, ορίζει τη σωστή απάντηση και την επεξήγηση που θα εμφανιστεί μετά την υποβολή. Στη συνέχεια, όταν ο φοιτητής ανοίξει την άσκηση, το σύστημα ανακτά τα αποθηκευμένα δεδομένα και εμφανίζει τη διεπαφή που αντιστοιχεί στον συγκεκριμένο τύπο άσκησης.

Για παράδειγμα, στις ασκήσεις Drag & Drop εμφανίζονται στοιχεία προς αντιστοίχιση και κατηγορίες, στις ασκήσεις Επιλογής Κανόνων εμφανίζονται βήματα με dropdown menu επιλογής κανόνα, ενώ στις ασκήσεις Εξέτασης Ισοδυναμίας και Ελέγχου Απόδειξης εμφανίζονται οι αντίστοιχες επιλογές απάντησης.

Μετά την υποβολή της απάντησης, πραγματοποιείται έλεγχος ορθότητας από την πλευρά του server, συγκρίνοντας την απάντηση του χρήστη με τη σωστή απάντηση που είναι αποθηκευμένη για τη συγκεκριμένη άσκηση. Ο έλεγχος αυτός διασφαλίζει ότι η αξιολόγηση βασίζεται στα δεδομένα του συστήματος και όχι μόνο στα δεδομένα που εμφανίζονται στο frontend.

Τέλος, το σύστημα επιστρέφει στον φοιτητή το αποτέλεσμα της προσπάθειας μαζί με ανατροφοδότηση. Παρόλο που όλοι οι τύποι ασκήσεων ακολουθούν την ίδια ροή, διαφοροποιούνται ως προς τη μορφή των δεδομένων, τον τρόπο αλληλεπίδρασης και τον τρόπο ελέγχου της απάντησης.

5.5.1 Υλοποίηση Ασκήσεων Drag & Drop

Οι ασκήσεις τύπου Drag & Drop χρησιμοποιούνται για δραστηριότητες αντιστοίχισης, όπου ο φοιτητής πρέπει να μετακινήσει στοιχεία και να τα τοποθετήσει στη σωστή κατηγορία.

Ο διαχειριστής ορίζει τα στοιχεία που θα εμφανιστούν στον φοιτητή, τις διαθέσιμες κατηγορίες και τις σωστές αντιστοιχίσεις. Με αυτόν τον τρόπο, κάθε άσκηση μπορεί να περιλαμβάνει διαφορετικό αριθμό στοιχείων και κατηγοριών.

Όταν ο φοιτητής επιλέξει μία άσκηση Drag & Drop, το σύστημα ανακτά τα αποθηκευμένα δεδομένα και δημιουργεί δυναμικά τα στοιχεία της άσκησης στην οθόνη. Τα στοιχεία που πρέπει να μετακινηθούν εμφανίζονται σε ξεχωριστή περιοχή, ενώ οι κατηγορίες εμφανίζονται ως περιοχές στις οποίες μπορεί να γίνει απόθεση. Η μεταφορά των στοιχείων και η τοποθέτηση τους στις κατηγορίες γίνεται με JavaScript, χωρίς να πρέπει να ανανεωθεί η σελίδα.

Όταν ο φοιτητής υποβάλει την απάντηση του, το σύστημα παίρνει τις αντιστοιχίσεις που δημιούργησε και τις συγκρίνει με τις σωστές αντιστοιχίσεις που είναι αποθηκευμένες στη βάση. Αν όλες οι αντιστοιχίσεις είναι σωστές, η άσκηση θεωρείται σωστή. Αν υπάρχει έστω και μία λανθασμένη αντιστοίχιση, η άσκηση θεωρείται λάθος.

5.5.2 Υλοποίηση Ασκήσεων Επιλογής Κανόνων

Οι ασκήσεις Επιλογής Κανόνα υλοποιήθηκαν ώστε ο φοιτητής να μπορεί να αναγνωρίζει ποιος λογικός κανόνας εφαρμόστηκε σε κάθε βήμα μιας λογικής ισοδυναμίας. Κάθε βήμα περιέχει

μια λογική ισοδυναμία. Ο φοιτητής εδώ επιλέγει τον κατάλληλο κανόνα από την διαθέσιμη λίστα επιλογών.

Ο διαχειριστής εισάγει σε κάθε βήμα μια λογική ισοδυναμία και επιλέγει από τη λίστα τον σωστό κανόνα για κάθε βήμα. Αν ο διαχειριστής προσθέσει νέο βήμα, δημιουργείται η αντίστοιχη επιλογή σωστής απάντησης. Έτσι, διαφορετικές ασκήσεις μπορούν να έχουν διαφορετικό αριθμό βημάτων.

Κατά την εκτέλεση της άσκησης, το σύστημα εμφανίζει δυναμικά όλα τα βήματα και κάτω από κάθε βήμα εμφανίζεται dropdown menu με τους διαθέσιμους κανόνες. Ο φοιτητής επιλέγει τον κανόνα που θεωρεί ότι εφαρμόζεται σε κάθε περίπτωση και στη συνέχεια υποβάλλει την απάντηση του.

Μετά την υποβολή, οι επιλογές του φοιτητή συγκρίνονται με τους σωστούς κανόνες που έχουν αποθηκευτεί. Η σύγκριση γίνεται ανά βήμα, δηλαδή η επιλογή του πρώτου βήματος συγκρίνεται με τη σωστή απάντηση του πρώτου βήματος, η επιλογή του δεύτερου με τη σωστή απάντηση του δεύτερου κ.ο.κ. Η άσκηση θεωρείται σωστή μόνο όταν έχουν επιλεγεί σωστά όλοι οι κανόνες.

5.5.3 Υλοποίηση Ασκήσεων Εξέτασης Ισοδυναμίας

Οι ασκήσεις Εξέτασης Ισοδυναμίας υλοποιήθηκαν ώστε ο φοιτητής να μπορεί να αναγνωρίζει ποιες λογικές προτάσεις είναι ισοδύναμες με μία αρχική πρόταση. Σε αυτόν τον τύπο άσκησης παρουσιάζεται μία λογική έκφραση και πολλές επιλογές πιθανών ισοδύναμων προτάσεων, από τις οποίες ο χρήστης πρέπει να επιλέξει όλες τις σωστές απαντήσεις.

Κατά τη δημιουργία μιας νέας άσκησης, ο διαχειριστής ορίζει την αρχική πρόταση και τις πιθανές απαντήσεις και επιλέγει ποιες από αυτές θεωρούνται σωστές. Με αυτό τον τρόπο η πλατφόρμα μπορεί να υποστηρίζει ασκήσεις με διαφορετικό επίπεδο δυσκολίας και διαφορετικό αριθμό σωστών επιλογών.

Στη διεπαφή του φοιτητή, εμφανίζεται αρχικά η πρόταση που πρέπει να ελέγξει. Στη συνέχεια παρουσιάζονται οι διαθέσιμες επιλογές με checkboxes, αφού μπορεί να υπάρχουν περισσότερες από μία σωστές απαντήσεις.

Κατά την αξιολόγηση, το σύστημα συγκρίνει τις επιλογές του φοιτητή με τις σωστές επιλογές που έχουν οριστεί για την άσκηση. Η σειρά με την οποία επιλέχθηκαν οι απαντήσεις δεν επηρεάζει το αποτέλεσμα. Αυτό που ελέγχεται είναι αν το τελικό σύνολο επιλογών του χρήστη είναι ίδιο με το σύνολο των σωστών απαντήσεων. Η απάντηση θεωρείται σωστή μόνο όταν ο φοιτητής έχει επιλέξει όλες τις σωστές επιλογές και καμία λανθασμένη. Αν λείπει κάποια σωστή επιλογή ή έχει επιλεγεί επιπλέον λανθασμένη απάντηση, η προσπάθεια θεωρείται λανθασμένη.

5.5.4 Υλοποίηση Ασκήσεων Ελέγχου Απόδειξης

Οι ασκήσεις τύπου Έλεγχος Απόδειξης δίνουν στον φοιτητή μια ολοκληρωμένη απόδειξη και πρέπει να αξιολογήσει αν είναι σωστή ή λανθασμένη. Σε αυτόν τον τύπο άσκησης, ο χρήστης δεν συμπληρώνει βήματα ούτε επιλέγει κανόνες, αλλά εξετάζει συνολικά την εγκυρότητα της δοσμένης λύσης.

Όταν ο διαχειριστής δημιουργεί νέα άσκηση, εισάγει το κείμενο της απόδειξης και ορίζει αν αυτή θεωρείται σωστή ή λανθασμένη. Επιπλέον, μπορεί να προσθέσει επεξήγηση που θα εμφανιστεί μετά την αξιολόγηση της απάντησης.

Όταν ο φοιτητής ανοίγει μία άσκηση, το σύστημα ανακτά το αποθηκευμένο περιεχόμενο και εμφανίζει την απόδειξη διατηρώντας τη μορφοποίηση και τη διάταξη των βημάτων. Κάτω από την απόδειξη παρουσιάζονται δύο διαθέσιμες επιλογές απάντησης, «Ναι» και «Όχι», μέσω radio buttons, επιτρέποντας στον χρήστη να επιλέξει μόνο μία τελική απάντηση σχετικά με την εγκυρότητα της απόδειξης.

Μετά την υποβολή, το σύστημα συγκρίνει την επιλογή του φοιτητή με τη σωστή απάντηση που έχει αποθηκευτεί για τη συγκεκριμένη άσκηση. Αν οι δύο τιμές είναι οι ίδιες, η απάντηση θεωρείται σωστή, αλλιώς λανθασμένη.

5.6 Υλοποίηση Quizzes

Η υλοποίηση των quizzes καλύπτει όλη τη διαδικασία, από τη δημιουργία τους από τον διαχειριστή μέχρι την εκτέλεσή τους από τον φοιτητή. Συγκεκριμένα, περιλαμβάνει τη διαχείριση ερωτήσεων και επιλογών απάντησης, την υποβολή των απαντήσεων, τον υπολογισμό της βαθμολογίας και την αποθήκευση των αποτελεσμάτων στη βάση δεδομένων.

Η βασική λογική υλοποιείται μέσω του QuizController, ενώ η ανάκτηση και αποθήκευση δεδομένων πραγματοποιείται μέσω των QuizRepository και UserQuizRepository. Με τον τρόπο αυτό διαχωρίζεται η λογική από την πρόσβαση στα δεδομένα.

5.6.1 Δημιουργία Quiz

Η δημιουργία ενός νέου quiz πραγματοποιείται από την σελίδα διαχείρισης. Ο διαχειριστής εισάγει βασικές πληροφορίες όπως τον τίτλο του quiz και προαιρετικά το κεφάλαιο θεωρίας με το οποίο συνδέεται.

Μετά την υποβολή της φόρμας, τα δεδομένα αποστέλλονται στον QuizController, όπου γίνεται έλεγχος εγκυρότητας και στη συνέχεια αποθηκεύονται στον πίνακα quizzes μέσω του QuizRepository.

Κάθε quiz μπορεί να περιλαμβάνει πολλές ερωτήσεις, οι οποίες προσθέτονται ξεχωριστά μετά τη δημιουργία του.

5.6.2 Δημιουργία και Διαχείριση Ερωτήσεων

Οι ερωτήσεις ενός quiz δημιουργούνται επίσης από το περιβάλλον του διαχειριστή. Για κάθε ερώτηση, ο διαχειριστής συμπληρώνει το κείμενο της ερώτησης, ορίζει τον τύπο της, δηλαδή αν έχει μία σωστή απάντηση ή περισσότερες από μία σωστές απαντήσεις, και προσθέτει τις διαθέσιμες επιλογές απάντησης. Παράλληλα, επιλέγει ποιες από τις επιλογές είναι σωστές και προσθέτει επεξήγηση, η οποία εμφανίζεται στον φοιτητή μετά την ολοκλήρωση του quiz.

Η αποθήκευση των ερωτήσεων γίνεται μέσω της συνάρτησης saveQuestion(), η οποία εισάγει ή ενημερώνει τα δεδομένα στους πίνακες questions και choices. Με αυτόν τον τρόπο, κάθε quiz μπορεί να περιλαμβάνει δυναμικό αριθμό ερωτήσεων και επιλογών, ενώ υποστηρίζονται διαφορετικοί τύποι ερωτήσεων χωρίς να απαιτούνται αλλαγές στη βασική δομή του συστήματος.

5.6.3 Υποβολή Απαντήσεων Quiz

Κατά την εκτέλεση ενός quiz, το σύστημα ανακτά από τη βάση δεδομένων τις ερωτήσεις και τις διαθέσιμες επιλογές απάντησης και τις εμφανίζει στον φοιτητή. Ο χρήστης επιλέγει τις απαντήσεις του και με την ολοκλήρωση του quiz, τις υποβάλλει για αξιολόγηση.

Μετά την υποβολή, οι απαντήσεις αποστέλλονται στον QuizController, όπου ελέγχεται κάθε ερώτηση ξεχωριστά. Για κάθε ερώτηση υπολογίζεται ο αντίστοιχος βαθμός και στη συνέχεια προκύπτει η συνολική βαθμολογία του quiz. Παράλληλα, η προσπάθεια του χρήστη αποθηκεύεται στον πίνακα quiz_attempts, ενώ οι επιμέρους απαντήσεις του χρήστη αποθηκεύονται στον πίνακα answers.

Η αποθήκευση των προσπαθειών επιτρέπει στον φοιτητή να προβάλλει αργότερα τα αποτελέσματά του μετά την ολοκλήρωση κάποιου quiz, ενώ δίνει και στον διαχειριστή την δυνατότητα να παρακολουθεί τη συμμετοχή και την επίδοση των χρηστών στα quizzes.

5.6.4 Υπολογισμός Βαθμολογίας και Penalty System

Ο υπολογισμός της βαθμολογίας πραγματοποιείται στον QuizController, μέσω της συνάρτησης calculateResult(). Η συνάρτηση αξιολογεί κάθε ερώτηση ξεχωριστά, ανάλογα με τον τύπο της.

Για ερωτήσεις με μία σωστή απάντηση εφαρμόζεται πλήρης ή μηδενική βαθμολόγηση, ανάλογα με την ορθότητα της επιλογής. Για ερωτήσεις με πολλαπλές σωστές απαντήσεις εφαρμόζεται σύστημα ποινής, το οποίο λαμβάνει υπόψη τόσο τις σωστές όσο και τις λανθασμένες επιλογές του χρήστη. Η λογική βαθμολόγησης παρουσιάστηκε στο Υποκεφάλαιο 4.5. Η υλοποίηση διασφαλίζει ότι η τελική βαθμολογία κάθε ερώτησης δεν μπορεί να γίνει αρνητική.

Μετά τον υπολογισμό της βαθμολογίας όλων των ερωτήσεων, υπολογίζεται η συνολική βαθμολογία του quiz, μετατρέπεται σε ποσοστό και αποθηκεύεται στον πίνακα quiz_attempts ενώ οι επιμέρους απαντήσεις αποθηκεύονται στον πίνακα answers.

5.7 Υλοποίηση Διαχείρισης Χρηστών

Η πλατφόρμα παρέχει στον διαχειριστή δυνατότητα προβολής των εγγεγραμμένων χρηστών μέσω της σελίδας διαχείρισης χρηστών. Τα δεδομένα ανακτώνται από τον πίνακα users μέσω του UserRepository και εμφανίζονται σε μορφή πίνακα.

Για κάθε χρήστη προβάλλονται βασικές πληροφορίες, όπως το ονοματεπώνυμο, το email, ο ρόλος και η ημερομηνία εγγραφής. Επιπλέον, παρέχονται λειτουργίες αναζήτησης και φιλτραρίσματος, διευκολύνοντας τη διαχείριση μεγάλου αριθμού χρηστών.

Η ανάκτηση των χρηστών πραγματοποιείται μέσω της μεθόδου getAllUsers(), η οποία υποστηρίζει αναζήτηση με βάση το όνομα ή το email, καθώς και φιλτράρισμα ανά ρόλο χρήστη. Παράλληλα, η μέθοδος αποκλείει χρήστες που έχουν διαγραφεί, ελέγχοντας την τιμή του πεδίου is_deleted αφού οι χρήστες διαγράφονται με soft delete.

Ο διαχειριστής έχει τη δυνατότητα προβολής των αποτελεσμάτων quiz κάθε χρήστη. Για κάθε χρήστη εμφανίζονται οι ολοκληρωμένες προσπάθειες σε quiz, μαζί με πληροφορίες όπως ο τίτλος του quiz και η τελική του βαθμολογία.

Η πλατφόρμα παρέχει στον διαχειριστή δυνατότητα επαναφοράς της προσπάθειας ενός χρήστη για συγκεκριμένο quiz. Κατά την επαναφορά, το σύστημα εντοπίζει την τελευταία καταγεγραμμένη προσπάθεια του χρήστη για το επιλεγμένο quiz. Αρχικά διαγράφονται οι σχετικές εγγραφές από τον πίνακα answers και στη συνέχεια διαγράφεται η αντίστοιχη εγγραφή από τον πίνακα quiz_attempts. Με αυτό τον τρόπο, ο χρήστης μπορεί να επαναλάβει το quiz σαν να μην είχε προηγουμένως ολοκληρώσει προσπάθεια για αυτό.

Η διαγραφή χρηστών υλοποιείται με μηχανισμό soft delete, ώστε οι εγγραφές να μην αφαιρούνται οριστικά από τη βάση δεδομένων. Ο μηχανισμός αυτός προσφέρει μεγαλύτερη ευελιξία, καθώς σε περίπτωση διαγραφής καταλάθος ο λογαριασμός μπορεί θεωρητικά να ξαναενεργοποιηθεί. Με αυτόν τον τρόπο, η διαχείριση χρηστών γίνεται πιο ασφαλής.

Κεφάλαιο 6

Συμπεράσματα

6.1 Μεθοδολογία αξιολόγησης	82
6.2 Ανάλυση αποτελεσμάτων	83
6.3 Συμπεράσματα	84
6.4 Μελλοντική Εργασία	84

6.1 Μεθοδολογία αξιολόγησης

Μετά την ολοκλήρωση της ανάπτυξης της πλατφόρμας πραγματοποιήθηκε αξιολόγηση του συστήματος, με στόχο την εξέταση της ευχρηστίας, της λειτουργικότητας και της συμβολής της πλατφόρμας στην εκμάθηση της Προτασιακής Λογικής.

Η αξιολόγηση πραγματοποιήθηκε από φοιτητές Πληροφορικής οι οποίοι έχουν παρακολουθήσει το μάθημα ΕΠΛ111. Οι συμμετέχοντες χρησιμοποίησαν τις βασικές λειτουργίες της πλατφόρμας, όπως η μελέτη θεωρίας, η επίλυση ασκήσεων και η συμμετοχή σε quizzes, πριν προχωρήσουν στην αξιολόγηση. Η επιλογή φοιτητών που έχουν ήδη παρακολουθήσει το μάθημα αυτό επέτρεψε την αξιολόγηση της πλατφόρμας από χρήστες με σχετική εμπειρία στο αντικείμενο, οι οποίοι μπορούσαν να αξιολογήσουν καλύτερα τη χρησιμότητα του εκπαιδευτικού περιεχομένου.

Η συλλογή δεδομένων πραγματοποιήθηκε μέσω ερωτηματολογίου, το οποίο περιλάμβανε ερωτήσεις κλίμακας Likert (1–5), καθώς και ανοικτές ερωτήσεις για προτάσεις βελτίωσης. Οι ερωτήσεις αξιολόγησαν στοιχεία όπως η οργάνωση του περιεχομένου, η ευκολία πλοήγησης, η χρησιμότητα των ασκήσεων, η αποτελεσματικότητα της ανατροφοδότησης, η κατανόηση του θεωρητικού υλικού, καθώς και η συνολική εκπαιδευτική εμπειρία των χρηστών. Επιπλέον, ζητήθηκαν προτάσεις σχετικά με πιθανές βελτιώσεις και μελλοντικές λειτουργίες της πλατφόρμας.

6.2 Ανάλυση αποτελεσμάτων

Η αξιολόγηση της πλατφόρμας πραγματοποιήθηκε μέσω ερωτηματολογίου. Στόχος ήταν να εξεταστεί η εμπειρία χρήσης, η ευχρηστία της πλατφόρμας, η ποιότητα του εκπαιδευτικού υλικού, καθώς και η χρησιμότητα των ασκήσεων και των quizzes. Από τις απαντήσεις προέκυψε ότι οι συμμετέχοντες είχαν γενικά θετική εικόνα για την πλατφόρμα.

Η εμφάνιση και η πλοήγηση στην πλατφόρμα αξιολογήθηκαν θετικά, καθώς οι χρήστες μπορούσαν εύκολα να εντοπίσουν τις βασικές λειτουργίες, όπως τη θεωρία, τις ασκήσεις και τα quizzes. Η καθαρή διάταξη των σελίδων και η οργάνωση του περιεχομένου συνέβαλαν στη γρήγορη εξοικείωση με το περιβάλλον της πλατφόρμας.

Το θεωρητικό υλικό θεωρήθηκε κατανοητό και καλά οργανωμένο σε ενότητες και κεφάλαια. Οι συμμετέχοντες αξιολόγησαν θετικά τη σαφήνεια των επεξηγήσεων και τη σταδιακή παρουσίαση της ύλης, πράγμα που δείχνει ότι η θεωρία μπορεί να υποστηρίξει αποτελεσματικά τη μελέτη της Προτασιακής Λογικής.

Οι ασκήσεις αξιολογήθηκαν ως χρήσιμες για την πρακτική εφαρμογή της θεωρίας. Οι διαφορετικοί τύποι δραστηριοτήτων έδωσαν στους χρήστες τη δυνατότητα να εξασκηθούν με πολλούς διαφορετικούς τρόπους, ενώ οι οδηγίες και η ανατροφοδότηση βοήθησαν στην καλύτερη κατανόηση των απαντήσεων.

Τα quizzes θεωρήθηκαν χρήσιμο εργαλείο αυτοαξιολόγησης. Οι χρήστες κατανόησαν τον τρόπο βαθμολόγησης, ενώ ο το σύστημα αρνητικής βαθμολόγησης στις ερωτήσεις με περισσότερες από μία σωστές απαντήσεις φάνηκε να ενθαρρύνει την πιο προσεκτική επιλογή απαντήσεων. Παράλληλα, η προβολή αποτελεσμάτων μετά την ολοκλήρωση του quiz συνέβαλε στην άμεση ενημέρωση του χρήστη για την επίδοση του.

Συνολικά, η εμπειρία χρήσης αξιολογήθηκε θετικά. Οι συμμετέχοντες ανέφεραν ότι η πλατφόρμα μπορεί να βοηθήσει στη μελέτη και εξάσκηση στην Προτασιακή Λογική, ενώ οι προτάσεις βελτίωσης που καταγράφηκαν αφορούσαν κυρίως την προσθήκη επιπλέον λειτουργιών, όπως flashcards, notes, rewards, ενσωμάτωση ασκήσεων μέσα στο θεωρητικό υλικό και να μπορούν να επεξεργάζονται το προφίλ τους.

6.3 Συμπεράσματα

Η ανάπτυξη της εκπαιδευτικής πλατφόρμας έδειξε ότι ο συνδυασμός θεωρίας, διαδραστικών ασκήσεων, quizzes και άμεσης ανατροφοδότησης μπορεί να λειτουργήσει υποστηρικτικά στη μελέτη της Προτασιακής Λογικής.

Από την αξιολόγηση προέκυψε ότι οι χρήστες είχαν γενικά θετική εμπειρία. Η ευχρηστία, η οργάνωση του περιεχομένου, το εκπαιδευτικό υλικό και οι λειτουργίες αυτοαξιολόγησης αξιολογήθηκαν θετικά, γεγονός που δείχνει ότι η πλατφόρμα ανταποκρίνεται σε μεγάλο βαθμό στους στόχους που είχαν τεθεί.

Σημαντικό στοιχείο αποτελεί και η θετική ανταπόκριση στους διαφορετικούς τύπους ασκήσεων και στην ανατροφοδότηση που παρέχεται μετά την υποβολή των απαντήσεων. Οι λειτουργίες αυτές φαίνεται να βοηθούν τον φοιτητή να εξασκείται πιο ενεργά και να κατανοεί καλύτερα τα λάθη του. Παράλληλα, το σύστημα βαθμολόγησης των quizzes με την ποινή για τις λανθασμένες επιλογές, φαίνεται να συμβάλλει σε πιο προσεκτική επιλογή των απαντήσεων από τους φοιτητές.

Συνολικά, η πλατφόρμα μπορεί να θεωρηθεί ένα χρήσιμο εκπαιδευτικό εργαλείο για την υποστήριξη της εκμάθησης της Προτασιακής Λογικής.

6.4 Μελλοντική Εργασία

Η πλατφόρμα μπορεί να επεκταθεί με νέες λειτουργίες που θα βελτιώσουν την εμπειρία μάθησης. Μία μελλοντική επέκταση είναι η προσθήκη εργαλείων παρακολούθησης προόδου, όπως progress bars, ποσοστά ολοκλήρωσης εννοιών και αναλυτικά στατιστικά. Με τον τρόπο αυτό, οι χρήστες θα μπορούν να εντοπίζουν ευκολότερα τα δυνατά και αδύνατα σημεία τους. Επιπλέον, θα μπορούσαν να ενσωματωθούν badges ή σύστημα επιβραβεύσεων, ώστε να αυξηθεί το κίνητρο και η αλληλεπίδραση των φοιτητών με την πλατφόρμα. Η ανάγκη για τέτοιες λειτουργίες προέκυψε και από τις προτάσεις των συμμετεχόντων στην αξιολόγηση.

Μια ακόμη μελλοντική επέκταση αφορά την προσθήκη flashcards, σημειώσεων, περιλήψεων και ηχητικού υλικού, ώστε να υποστηρίζονται διαφορετικοί τρόποι μελέτης και μάθησης. Επιπλέον, θα μπορούσε να υλοποιηθεί δυνατότητα επεξεργασίας στοιχείων προφίλ χρήστη, καθώς και πιο εξατομικευμένη διαχείριση λογαριασμού.

Επιπρόσθετα, η σύνδεση των quizzes με το θεωρητικό περιεχόμενο θα μπορούσε να επεκταθεί, ώστε κάθε quiz να σχετίζεται με ένα ή περισσότερα κεφάλαια θεωρίας. Με τον τρόπο αυτό, θα είναι δυνατή η δημιουργία πιο σύνθετων quizzes που θα καλύπτουν συνδυαστικά πολλαπλές ενότητες, προσφέροντας μια πιο ολοκληρωμένη διαδικασία αξιολόγησης των γνώσεων του χρήστη.

Μια ακόμη σημαντική επέκταση αφορά την υλοποίηση μηχανισμού δυναμικής δημιουργίας quizzes. Αντί τα quizzes να είναι προκαθορισμένα, το σύστημα θα μπορούσε να αντλεί ερωτήσεις από ένα question pool, επιλέγοντας τυχαία ερωτήσεις διαφορετικών επιπέδων δυσκολίας. Με αυτόν τον τρόπο, κάθε νέα προσπάθεια θα οδηγεί στη δημιουργία ενός διαφορετικού quiz, ενισχύοντας την ποικιλία των ασκήσεων και μειώνοντας την πιθανότητα αποστήθισης των απαντήσεων.

Παράλληλα, η ανατροφοδότηση θα μπορούσε να βελτιωθεί μέσω της αξιοποίησης τεχνικών Τεχνητής Νοημοσύνης. Αντί της εμφάνισης στατικών επεξηγήσεων, το σύστημα θα μπορούσε να παράγει εξατομικευμένες επεξηγήσεις προσαρμοσμένες στα λάθη και στις ανάγκες κάθε χρήστη. Μια τέτοια προσέγγιση θα μπορούσε να προσφέρει πιο αναλυτικές διευκρινίσεις, εναλλακτικούς τρόπους κατανόησης των εννοιών και στοχευμένες επεξηγήσεις, συμβάλλοντας σε μια πιο προσωποποιημένη μαθησιακή εμπειρία.

Τέλος, μελλοντική εργασία μπορεί να αποτελέσει η επέκταση της πλατφόρμας ώστε να υποστηρίζει επιπλέον θεματικές ενότητες λογικής της Πληροφορικής, μετατρέποντας την σε ένα πιο ολοκληρωμένο εκπαιδευτικό περιβάλλον.

Βιβλιογραφία

- [1] Κανόνες Ευχρηστίας κατά Nielsen, Διάλεξη μαθήματος ΕΠΛ435
- [2] Kenneth H. Rosen - Discrete Mathematics and Its Applications 7th Edition
- [3] Διάλεξη μαθήματος ΕΠΛ111 - Προτασιακή Λογική
- [4] Ian Sommerville - Software Engineering 10th Edition
- [5] Διαλέξεις μαθήματος ΕΠΛ342 - Βάσεις Δεδομένων - ER διαγράμματα
- [6] Διαλέξεις μαθήματος ΕΠΛ425 - Internet Technologies
- [7] Jon Duckett - HTML & CSS Design and build Websites
- [8] Modeliosoft, *Modelio Documentation*, [Ηλεκτρονικό]
Διαθέσιμο από: <https://www.modelio.org/>
- [9] Mokhtar A. A., *ERDPlus: Entity Relationship Diagram Tool* [Ηλεκτρονικό]
Διαθέσιμο από: <https://erdplus.com/>
- [10] Bootstrap Bootstrap Team, *Bootstrap Documentation* [Ηλεκτρονικό]
Διαθέσιμο από: <https://getbootstrap.com/docs/>
- [11] PHP PHP Documentation Team, *PHP Manual* [Ηλεκτρονικό]
Διαθέσιμο από: <https://www.php.net/docs.php>
- [12] jQuery OpenJS Foundation, *jQuery API Documentation* [Ηλεκτρονικό]
Διαθέσιμο από: <https://api.jquery.com/>
- [13] Font Awesome Fonticons Inc., *Font Awesome Documentation* [Ηλεκτρονικό]
Διαθέσιμο από: <https://fontawesome.com/docs>
- [14] PHP Documentation Team, PHP Data Objects (PDO) [Ηλεκτρονικό]

Διαθέσιμο από: <https://www.php.net/manual/en/book.pdo.php>

[15] Summernote Team, Summernote: Super Simple WYSIWYG Editor, [Ηλεκτρονικό]

Διαθέσιμο από: <https://summernote.org/>

Παράρτημα Α

Ερωτηματολόγιο αξιολόγησης πλατφόρμας από τους φοιτητές.

Ερωτηματολόγιο Αξιολόγησης Εκπαιδευτικής Πλατφόρμας

Το παρόν ερωτηματολόγιο δημιουργήθηκε στο πλαίσιο αξιολόγησης της εκπαιδευτικής πλατφόρμας εκμάθησης Λογικής. Στόχος του είναι η συλλογή απόψεων σχετικά με την ευχρηστία, τη λειτουργικότητα και τη συνολική εμπειρία χρήσης της εφαρμογής. Οι απαντήσεις είναι ανώνυμες και θα χρησιμοποιηθούν αποκλειστικά για σκοπούς αξιολόγησης της παρούσας πτυχιακής εργασίας.

Για τις παρακάτω ερωτήσεις χρησιμοποιείται η ακόλουθη κλίμακα αξιολόγησης:

- 1 = Διαφωνώ απόλυτα
- 2 = Διαφωνώ
- 3 = Ούτε συμφωνώ ούτε διαφωνώ
- 4 = Συμφωνώ
- 5 = Συμφωνώ απόλυτα

[Επόμενο](#)[Εκκαθάριση φόρμας](#)

Ενότητα 1: Δημογραφικά Στοιχεία

Έχετε προηγούμενη εμπειρία με εκπαιδευτικές πλατφόρμες;

☐ Ναι

☐ Όχι

Πόσο εξοικειωμένος/η είστε με την Προτασιακή Λογική πριν χρησιμοποιήσετε την πλατφόρμα;

☐ Καθόλου

☐ Λίγο

☐ Μέτρια

☐ Πολύ

☐ Πάρα πολύ

Ενότητα 2: UI και γενική εμφάνιση

Η διάταξη των σελίδων της πλατφόρμας ήταν οργανωμένη και κατανοητή. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η εμφάνιση της πλατφόρμας συνέβαλε θετικά στην εμπειρία χρήσης. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η χρωματική παλέτα και τα οπτικά στοιχεία διευκόλυναν την ανάγνωση και πλοήγηση. *

1	2	3	4	5
☐	☐	☐	☐	☐

Τα κουμπιά και οι λειτουργίες ήταν εύκολα αναγνωρίσιμα. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η παρουσίαση του περιεχομένου ήταν συνεπής μεταξύ διαφορετικών σελίδων. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η μετάβαση μεταξύ θεωρίας, ασκήσεων και quizzes ήταν εύκολη και ξεκάθαρη. *

1	2	3	4	5
☐	☐	☐	☐	☐

Ενότητα 3: Ευχρηστία και πλοήγηση

Η πλοήγηση στην πλατφόρμα ήταν εύκολη χωρίς να απαιτούνται επιπλέον οδηγίες. *

1	2	3	4	5
☐	☐	☐	☐	☐

Μπόρεσα να εντοπίσω γρήγορα τη λειτουργία που αναζητούσα. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η δομή της πλατφόρμας βοήθησε στη γρήγορη εξοικείωση με το περιβάλλον. *

1	2	3	4	5
☐	☐	☐	☐	☐

Ενότητα 4: Θεωρητικό υλικό

Η οργάνωση του θεωρητικού υλικού σε ενότητες και κεφάλαια ήταν κατανοητή. *

1	2	3	4	5
☐	☐	☐	☐	☐

Οι εξηγήσεις του θεωρητικού περιεχομένου ήταν σαφείς. *

1	2	3	4	5
☐	☐	☐	☐	☐

Το θεωρητικό υλικό βοήθησε στην κατανόηση βασικών εννοιών της λογικής. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η δομή των κεφαλαίων υποστήριζε τη σταδιακή εκμάθηση. *

1	2	3	4	5
☐	☐	☐	☐	☐

Ενότητα 5: Ασκήσεις

Οι διαθέσιμοι τύποι ασκήσεων (Drag & Drop, Επιλογή Κανόνων, Εξέταση Ισοδυναμίας, Έλεγχος απόδειξης) ήταν χρήσιμοι για την εξάσκηση. *

1	2	3	4	5
☐	☐	☐	☐	☐

Οι ασκήσεις Drag & Drop ήταν χρήσιμες για την κατανόηση εννοιών. *

1	2	3	4	5
☐	☐	☐	☐	☐

Οι ασκήσεις επιλογής κανόνων βοήθησαν στην εξοικείωση με τους κανόνες λογικής *

1	2	3	4	5
☐	☐	☐	☐	☐

Οι ασκήσεις εξέτασης ισοδυναμίας συνέβαλαν στην ανάπτυξη αναλυτικής σκέψης. *

1	2	3	4	5
☐	☐	☐	☐	☐

Οι ασκήσεις ελέγχου απόδειξης βελτίωσαν την κατανόηση διαδικασιών λογικής απόδειξης. *

1	2	3	4	5
☐	☐	☐	☐	☐

Οι οδηγίες και το περιεχόμενο κάθε άσκησης ήταν σαφής. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η δομή παρουσίασης των ασκήσεων ήταν οργανωμένη. *

1	2	3	4	5
☐	☐	☐	☐	☐

Ενότητα 6: Quizzes

Τα quizzes βοήθησαν στην αξιολόγηση των γνώσεών μου. *

1	2	3	4	5
☐	☐	☐	☐	☐

Οι ερωτήσεις και το περιεχόμενο των quizzes ήταν κατανοητές. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η δομή των quizzes ήταν οργανωμένη. *

1	2	3	4	5
☐	☐	☐	☐	☐

Το σύστημα βαθμολόγησης ήταν κατανοητό. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η αρνητική βαθμολόγηση στις λανθασμένες απαντήσεις συνέβαλαν σε πιο προσεκτική επιλογή απαντήσεων. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η εμφάνιση των αποτελεσμάτων μετά την ολοκλήρωση του quiz ήταν σαφής. *

1	2	3	4	5
☐	☐	☐	☐	☐

Ενότητα 7: Ανατροφοδότηση και γενική εμπειρία

Η ανατροφοδότηση μετά την ολοκλήρωση ασκήσεων ήταν χρήσιμη. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η ανατροφοδότηση μετά τα quizzes ήταν κατανοητή. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η εμφάνιση αποτελεσμάτων με βοήθησε να εντοπίσω τυχόν λάθη. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η πλατφόρμα ενίσχυσε τη διαδικασία αυτοαξιολόγησης. *

1	2	3	4	5
☐	☐	☐	☐	☐

Η συνολική εμπειρία μάθησης μέσω της πλατφόρμας ήταν θετική. *

1	2	3	4	5
☐	☐	☐	☐	☐

Θα χρησιμοποιούσα ξανά την πλατφόρμα για μελέτη λογικής. *

1	2	3	4	5
☐	☐	☐	☐	☐

Θα πρότεινα την πλατφόρμα και σε άλλους φοιτητές. *

1	2	3	4	5
☐	☐	☐	☐	☐

Υπήρξαν δυσκολίες κατά τη χρήση; Αν ναι, περιγράψτε τις.

Η απάντησή σας

Τι βελτιώσεις θα προτείνατε;

Η απάντησή σας

Τι βελτιώσεις θα προτείνατε;

Η απάντησή σας

Τι λειτουργίες θα προτείνατε να προστεθούν στο μέλλον;

Η απάντησή σας

Παράρτημα Β

Θεωρητικό εκπαιδευτικό υλικό Προτασιακής Λογικής που ενσωματώθηκε στην πλατφόρμα.

Κεφάλαιο 1

1. Προτάσεις (Propositions)

Ορισμός: Πρόταση

Πρόταση είναι μία δήλωση που έχει συγκεκριμένη τιμή αληθείας: είναι είτε αληθής (True, T) είτε ψευδής (False, F), αλλά ποτέ και τα δύο ταυτόχρονα.

Οι προτάσεις συμβολίζονται συνήθως με κεφαλαία γράμματα: P, Q, R, S, ... Η τιμή αληθείας (truth value) κάθε πρότασης είναι **T** (αληθής) ή **F** (ψευδής).

1.1 Παραδείγματα Προτάσεων

Εξετάζουμε τα παρακάτω παραδείγματα και αποφαινόμαστε αν είναι προτάσεις:

Δήλωση	Είναι πρόταση;	Τιμή
Το Παρίσι είναι η πρωτεύουσα της Γαλλίας.	Ναι	True
Η Βαρκελώνη είναι η πρωτεύουσα της Ισπανίας.	Ναι	False
$5 + 3 = 8$	Ναι	True
$7 > 10$	Ναι	False
$x + 4 = 9$	Όχι	-
Κλείσε την πόρτα!	Όχι	-
Τι ώρα είναι;	Όχι	-

Οι εντολές και οι ερωτήσεις δεν είναι προτάσεις, επειδή δεν μπορούν να χαρακτηριστούν αληθείς ή ψευδείς.

Επίσης, εκφράσεις που περιέχουν ελεύθερες μεταβλητές όπως " $x + 4 = 9$ " δεν είναι προτάσεις, αφού δεν γνωρίζουμε την τιμή της μεταβλητής.

Κεφάλαιο 2

2. Λογικοί Τελεστές (Logical Operators)

Από υπάρχουσες προτάσεις μπορούμε να κατασκευάσουμε νέες, πιο σύνθετες προτάσεις χρησιμοποιώντας λογικούς τελεστές. Ακολουθεί συνοπτικός πίνακας όλων των τελεστών που θα μελετήσουμε:

Τελεστής	Σύμβολο	Αγγλικός Όρος	Προτεραιότητα
Άρνηση	$\neg$	Negation	1 (υψηλότερη)
Σύζευξη	$\wedge$	Conjunction	2
Διάζευξη	$\vee$	Disjunction	3
Αποκλειστική Διάζευξη	$\oplus$	Exclusive OR	4
Συνεπαγωγή	$\rightarrow$	Implication	5
Διπλή Συνεπαγωγή	$\leftrightarrow$	Biconditional	6 (χαμηλότερη)

2.1 Άρνηση (Negation) - $\neg P$

Ορισμός: Άρνηση

Η άρνηση της πρότασης P, που συμβολίζεται $\neg P$, είναι η πρόταση "Δεν ισχύει το P". Είναι αληθής όταν η P είναι ψευδής, και ψευδής όταν η P είναι αληθής.

Η άρνηση είναι μοναδιαίος (unary) τελεστής δηλαδή ενεργεί πάνω σε μία μόνο πρόταση.

P	$\neg P$
T	F
F	T

Παράδειγμα: Έστω P: "Ο αριθμός 12 είναι άρτιος". Τότε $\neg P$: "Ο αριθμός 12 δεν είναι άρτιος" (F).

2.2 Σύζευξη (Conjunction) - $P \wedge Q$

Ορισμός: Σύζευξη

Η σύζευξη των προτάσεων P και Q, που συμβολίζεται $P \wedge Q$, είναι αληθής αν και μόνο αν και οι δύο προτάσεις είναι αληθείς.

P	Q	$P \wedge Q$
T	T	T
T	F	F
F	T	F
F	F	F

Παράδειγμα: P: "Ο Νίκος γνωρίζει Python", Q: "Ο Νίκος γνωρίζει Java". Η $P \wedge Q$ ("Ο Νίκος γνωρίζει Python και Java") είναι αληθής μόνο αν γνωρίζει και τις δύο γλώσσες.

2.3 Διάζευξη (Disjunction) - $P \vee Q$

Ορισμός: Διάζευξη

Η διάζευξη των προτάσεων P και Q, που συμβολίζεται $P \vee Q$, είναι αληθής αν τουλάχιστον μία από τις δύο προτάσεις είναι αληθής.

P	Q	$P \vee Q$
T	T	T
T	F	T
F	T	T
F	F	F

Παράδειγμα: P: "Ο Κώστας είναι μηχανολόγος", Q: "Ο Κώστας είναι ηλεκτρολόγος". Η $P \vee Q$ είναι αληθής αν είναι τουλάχιστον ένα από τα δύο ή και τα δύο.

2.4 Αποκλειστική Διάζευξη (Exclusive OR) - $P \oplus Q$

Ορισμός: Αποκλειστική Διάζευξη

Η αποκλειστική διάζευξη των P και Q, που συμβολίζεται $P \oplus Q$, είναι αληθής αν ακριβώς μία από τις δύο προτάσεις είναι αληθής, όχι και οι δύο.

P	Q	$P \oplus Q$
T	T	F
T	F	T
F	T	T
F	F	F

Παράδειγμα: P: "Η Μαρία κέρδισε το χρυσό μετάλλιο", Q: "Η Μαρία κέρδισε το ασημένιο μετάλλιο". Η $P \oplus Q$ είναι αληθής αν κέρδισε ακριβώς ένα από τα δύο (δεν μπορεί να πάρει και τα δύο μετάλλια).

2.5 Συνεπαγωγή (Implication) - $P \rightarrow Q$

Ορισμός: Συνεπαγωγή

Η συνεπαγωγή $P \rightarrow Q$, είναι ("αν P , τότε Q ") είναι ψευδής αποκλειστικά όταν η υπόθεση P είναι αληθής και το συμπέρασμα Q είναι ψευδές. Σε κάθε άλλη περίπτωση είναι αληθής.

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Η P ονομάζεται υπόθεση (hypothesis/antecedent) και η Q συμπέρασμα (conclusion/consequent).

Παράδειγμα: "Αν σπουδάζεις στην Αθήνα, τότε έχεις ταξιδέψει στην Ελλάδα." Αυτή η πρόταση είναι ψευδής μόνο αν κάποιος σπουδάζει στην Αθήνα αλλά δεν έχει ταξιδέψει στην Ελλάδα - σενάριο εξαιρετικά απίθανο αλλά αυτό αποτελεί τον ορισμό της λογικής ψευδούς συνεπαγωγής.

Από μια συνεπαγωγή $P \rightarrow Q$ παράγουμε τρεις συναφείς προτάσεις:

Όνομα	Πρόταση	Είναι ισοδύναμη με $P \rightarrow Q$;
Αρχική (Original)	$P \rightarrow Q$	ΝΑΙ
Αντίστροφη (Converse)	$Q \rightarrow P$	ΟΧΙ, δεν είναι πάντα ισοδύναμη
Αντίθετη (Inverse)	$\neg P \rightarrow \neg Q$	ΟΧΙ, δεν είναι πάντα ισοδύναμη
Αντιθετοαντίστροφη (Contrapositive)	$\neg Q \rightarrow \neg P$	ΝΑΙ, πάντα ισοδύναμη

Ισοδύναμοι τρόποι έκφρασης της $P \rightarrow Q$:

- Αν P , τότε Q (If P , then Q)
- P συνεπάγεται Q (P implies Q)
- P είναι ικανή συνθήκη για το Q (P is sufficient for Q)
- P μόνο αν Q (P only if Q)
- Q αν P (Q if P)
- Q είναι αναγκαία συνθήκη για το P (Q is necessary for P)
- Q , εκτός αν $\neg P$ (Q unless $\neg P$)

2.6 Διπλή Συνεπαγωγή (Implication) - $P \leftrightarrow Q$

Ορισμός: Διπλή Συνεπαγωγή

Η πρόταση $P \leftrightarrow Q$ (" P αν και μόνο αν Q ") είναι αληθής αν και μόνο αν οι P και Q έχουν την ίδια τιμή αληθείας (και οι δύο T ή και οι δύο F).

P	Q	$P \leftrightarrow Q$
T	T	T
T	F	F
F	T	F
F	F	T

Σημαντικό: $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$. Δηλαδή η διπλή συνεπαγωγή ισχύει όταν η συνεπαγωγή ισχύει και προς τις δύο κατευθύνσεις

Παράδειγμα: P : "Το αρχείο αποθηκεύεται", Q : "Ο χρήστης πατά Ctrl+S". Οπότε $P \leftrightarrow Q$: "Το αρχείο αποθηκεύεται αν και μόνο αν ο χρήστης πατά Ctrl+S."

Τρόποι έκφρασης του $P \leftrightarrow Q$:

- P αν και μόνο αν Q (P if and only if Q / P iff Q)
- P είναι αναγκαία και ικανή συνθήκη για Q (P is necessary and sufficient for Q)
- Αν P τότε Q , και αντίστροφα (If P then Q , and conversely)

Κεφάλαιο 3

3. Προτεραιότητα Λογικών Τελεστών

Για να αποφεύγεται η ασάφεια σε σύνθετες προτάσεις, ορίζεται σειρά προτεραιότητας (precedence) για τους λογικούς τελεστές, ανάλογα με τα μαθηματικά (πχ. ο πολλαπλασιασμός πριν την πρόσθεση).

Τελεστής	Σύμβολο	Αγγλικός Όρος	Προτεραιότητα
Άρνηση	$\neg$	Negation	1 (υψηλότερη)
Σύζευξη	$\wedge$	Conjunction	2
Διάζευξη	$\vee$	Disjunction	3
Αποκλειστική Διάζευξη	$\oplus$	Exclusive OR	4
Συνεπαγωγή	$\rightarrow$	Implication	5
Διπλή επαγωγή	$\leftrightarrow$	Biconditional	6 (χαμηλότερη)

Παράδειγμα: Ανάλυση σύνθετων προτάσεων με βάση προτεραιότητα

Πρόταση (Χωρίς παρενθέσεις)	Ισοδύναμη (με παρενθέσεις)
$\neg P \wedge Q$	$(\neg P) \wedge Q$ η άρνηση $\neg$ δεσμεύει μόνο το P
$P \vee Q \wedge \neg R$	$P \vee (Q \wedge (\neg R))$ πρώτα $\neg$, μετά $\wedge$, μετά $\vee$
$P \vee Q \rightarrow R$	$(P \vee Q) \rightarrow R$ πρώτα $\vee$, μετά $\rightarrow$
$P \rightarrow Q \rightarrow R$	$(P \rightarrow Q) \rightarrow R$ Ξεκινούμε από αριστερά
$P \rightarrow Q \leftrightarrow R$	$(P \rightarrow Q) \leftrightarrow R$
$\neg P \vee Q \leftrightarrow R \wedge S$	$((\neg P) \vee Q) \leftrightarrow (R \wedge S)$

Είναι καλό να χρησιμοποιούμε παρενθέσεις για να αποφεύγουμε λάθη, ακόμα και αν η προτεραιότητα θα έδινε το ίδιο αποτέλεσμα. Η αναγνωσιμότητα είναι πολύ σημαντική.

Κεφάλαιο 4

4. Πίνακες αληθείας Σύνθετων προτάσεων

Έννοια του Πίνακα Αληθείας

Ο πίνακας αληθείας είναι ένας δομημένος πίνακας που παρουσιάζει όλες τις δυνατές περιπτώσεις τιμών αλήθειας των μεταβλητών μιας λογικής πρότασης και δείχνει ποια είναι η τελική τιμή της πρότασης σε κάθε περίπτωση.

Αν μια σύνθετη πρόταση περιέχει:

- 1 μεταβλητή, τότε ο πίνακας αληθείας έχει 2 γραμμές
- 2 μεταβλητές, τότε έχει 4 γραμμές
- 3 μεταβλητές, τότε έχει 8 γραμμές

Γενικά, για n προτασιακές μεταβλητές, ο πίνακας αληθείας περιέχει 2^n γραμμές.

Η διαδικασία κατασκευής ενός πίνακα αληθείας βασίζεται στην ανάλυση της πρότασης στα επιμέρους τμήματά της. Για κάθε ενδιάμεση λογική έκφραση δημιουργείται ξεχωριστή στήλη, ώστε να είναι δυνατός ο υπολογισμός της τελικής τιμής βήμα προς βήμα.

Παράδειγμα: Κατασκευή Πίνακα Αληθείας της Πρότασης

$(\neg P \wedge Q) \rightarrow (P \vee Q)$

Η πρόταση αυτή περιέχει δύο προτασιακές μεταβλητές, τις P και Q . Επομένως, ο πίνακας αληθείας θα περιλαμβάνει $2^2 = 4$ γραμμές.

P	Q	$\neg P$	$\neg P \wedge Q$	$P \vee Q$	$(\neg P \wedge Q) \rightarrow (P \vee Q)$
T	T	F	F	T	T
T	F	F	F	T	T
F	T	T	T	T	T
F	F	T	F	F	T

Από τον πίνακα αληθείας παρατηρείται ότι η πρόταση $(\neg P \wedge Q) \rightarrow (P \vee Q)$ είναι αληθής για κάθε δυνατό συνδυασμό τιμών αλήθειας των P και Q .

Επομένως, η πρόταση αυτή αποτελεί ταυτολογία, αφού είναι πάντοτε αληθής.

Κεφάλαιο 5

5. Εφαρμογές της Προτασιακής Λογικής

5.1 Γιατί είναι χρήσιμη η προτασιακή λογική;

Στην καθημερινή γλώσσα, πολλές προτάσεις είναι ασαφείς ή εξαρτώνται από το πλαίσιο στο οποίο χρησιμοποιούνται. Η προτασιακή λογική προσφέρει έναν τυπικό τρόπο έκφρασης, ώστε οι προτάσεις να αποκτούν σαφή και ξεκάθαρη μορφή.

Η προτασιακή λογική είναι χρήσιμη επειδή:

- μειώνει την ασάφεια της φυσικής γλώσσας
- επιτρέπει την ακριβή αναπαράσταση προτάσεων και κανόνων
- διευκολύνει τον έλεγχο ορθότητας μιας διατύπωσης
- υποστηρίζει τη λογική ανάλυση και τη διαδικασία συμπερασμού
- βοηθά στη μοντελοποίηση πραγματικών προβλημάτων

5.2 Μετάφραση προτάσεων από φυσική γλώσσα σε λογικές εκφράσεις

1. Αρχικά εντοπίζονται τα **επιμέρους νοηματικά τμήματα** μιας πρότασης, τα οποία μπορούν να εκφραστούν ως ανεξάρτητες προτασιακές μεταβλητές.
2. Κάθε βασική πρόταση αντιστοιχίζεται σε ένα **σύμβολο**, όπως P , Q , R .
3. Στη συνέχεια, επιλέγονται οι **κατάλληλοι λογικοί τελεστές**, ώστε να αποδοθεί σωστά η σημασία της αρχικής πρότασης.

Παράδειγμα: Ο χρήστης μπορεί να ολοκληρώσει την αγορά μόνο αν έχει προσθέσει προϊόν στο καλάθι και έχει επιλέξει τρόπο πληρωμής.

Βήμα 1: Ορίζουμε τις προτασιακές μεταβλητές

- p : Ο χρήστης μπορεί να ολοκληρώσει την αγορά
- q : Ο χρήστης έχει προσθέσει προϊόν στο καλάθι
- r : Ο χρήστης έχει επιλέξει τρόπο πληρωμής

Βήμα 2: Αναγνώριση λογικής σύνδεσης

- Η φράση "μόνο αν" δείχνει ότι πρέπει να χρησιμοποιήσουμε συνεπαγωγή.
- Επειδή πρέπει να ισχύουν και οι δύο προϋποθέσεις, χρησιμοποιείται η σύζευξη $\wedge$.

Λογική έκφραση: $p \rightarrow (q \wedge r)$

Ερμηνεία:

Αν ο χρήστης μπορεί να ολοκληρώσει την αγορά, τότε πρέπει να ισχύουν ταυτόχρονα και οι δύο παρακάτω συνθήκες:

- έχει προσθέσει προϊόν στο καλάθι
- έχει επιλέξει τρόπο πληρωμής

5.3 Η ασάφεια της φυσικής γλώσσας

Η φυσική γλώσσα είναι πολλές φορές αμφίσημη. Λέξεις όπως:

- "ή"
- "μόνο αν"
- "εκτός αν"
- "αν... τότε"

δεν χρησιμοποιούνται πάντα με τον ίδιο ακριβώς τρόπο στην καθημερινή επικοινωνία. Για παράδειγμα, η λέξη "ή" μπορεί άλλοτε να σημαίνει διάζευξη και άλλοτε αποκλειστική επιλογή.

Η προτασιακή λογική βοηθά στην αποσαφήνιση τέτοιων περιπτώσεων, επειδή κάθε σύνδεση ανάμεσα σε προτάσεις αποδίδεται με συγκεκριμένο τελεστή και συγκεκριμένους κανόνες αληθείας.

Κεφάλαιο 6

6. Προτασιακές Ισοδυναμίες (Propositional Equivalences)

Στην προτασιακή λογική, πολλές φορές δύο διαφορετικές λογικές εκφράσεις μπορεί να έχουν ακριβώς την ίδια τιμή αλήθειας για κάθε δυνατό συνδυασμό τιμών των μεταβλητών τους. Όταν συμβαίνει αυτό, λέμε ότι οι δύο προτάσεις είναι **λογικά ισοδύναμες**.

6.1 Κατηγορίες σύνθετων προτάσεων

Ανάλογα με το πώς συμπεριφέρεται μια σύνθετη πρόταση ως προς την αλήθεια της, μπορούμε να τη διακρίνουμε σε τρεις βασικές κατηγορίες:

1. **Ταυτολογία:** Μια πρόταση είναι ταυτολογία όταν είναι πάντα αληθής, ανεξάρτητα από τις τιμές αλήθειας των μεταβλητών της.
2. **Αντίφαση:** Μια πρόταση είναι αντίφαση όταν είναι πάντα ψευδής.
3. **Ενδεχομενική πρόταση:** Μια πρόταση είναι ενδεχομενική όταν σε κάποιες περιπτώσεις είναι αληθής και σε άλλες ψευδής.

6.2 Λογική Ισοδυναμία

Ορισμός: Λογική Ισοδυναμία

Δύο σύνθετες προτάσεις P και Q λέγονται λογικά ισοδύναμες $P \equiv Q$ αν και μόνο αν η πρόταση $P \leftrightarrow Q$ είναι ταυτολογία, δηλαδή όταν έχουν ίδιες τιμές αλήθειας σε όλες τις πιθανές περιπτώσεις.

6.3 Πώς ελέγχουμε αν δύο προτάσεις είναι ισοδύναμες;

Υπάρχουν δύο μέθοδοι απόδειξης ισοδυναμίας:

1. Πίνακες Αληθείας
2. Χρήση Λογικών Κανόνων

1. Πίνακες αληθείας:

Κατασκευάζουμε τον πίνακα αληθείας για τις προτάσεις P και Q και ελέγχουμε αν οι στήλες για P και Q είναι οι ίδιες.

Παράδειγμα: Να αποδειχθεί ότι οι προτάσεις $P \vee (Q \wedge R)$ και $(P \vee Q) \wedge (P \vee R)$ είναι λογικά ισοδύναμες. (Νόμος επιμεριστικότητας)

P	Q	R	$Q \wedge R$	$P \vee (Q \wedge R)$	$P \vee Q$	$P \vee R$	$(P \vee Q) \wedge (P \vee R)$
T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F
F	F	T	F	F	F	T	F
F	F	F	F	F	F	F	F

Συγκρίνουμε τις στήλες $P \vee (Q \wedge R)$ και $(P \vee Q) \wedge (P \vee R)$ και παρατηρούμε ότι οι δύο στήλες έχουν ακριβώς τις ίδιες τιμές αληθείας σε όλες τις γραμμές του πίνακα.

Συμπέρασμα: Αφού οι δύο προτάσεις έχουν ίδιες τιμές αληθείας για κάθε δυνατό συνδυασμό των P, Q και R, συμπεραίνουμε ότι είναι λογικά ισοδύναμες.

$$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$$

2. Χρήση Λογικών Κανόνων

Μετασχηματίζουμε βήμα-βήμα τη μια πρόταση στην άλλη, εφαρμόζοντας γνωστούς λογικούς κανόνες.

Λογικοί Κανόνες:

Identity laws	$P \wedge T \equiv P$ $P \vee F \equiv P$
Domination Laws	$P \wedge F \equiv F$ $P \vee T \equiv T$
Idempotent laws	$P \vee P \equiv P$ $P \wedge P \equiv P$
Νόμος διπλής άρνησης (Double Negation)	$\neg(\neg P) \equiv P$
Νόμοι αντιμεταθετικότητας (Commutative Laws)	$P \wedge Q \equiv Q \wedge P$ $P \vee Q \equiv Q \vee P$
Νόμοι Προσεταιριστικότητας (Associative Laws)	$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$ $(P \vee Q) \vee R \equiv P \vee (Q \vee R)$
Νόμοι επιμεριστικότητας (Distributive Laws)	$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$ $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$
De Morgan's Laws	$\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ $\neg(P \vee Q) \equiv \neg P \wedge \neg Q$
Absorption Laws	$P \wedge (P \vee Q) \equiv P$ $P \vee (P \wedge Q) \equiv P$
Negation Laws	$P \wedge \neg P \equiv F$ (Αντίφαση) $P \vee \neg P \equiv T$ (Ταυτολογία)

Παράδειγμα: Απόδειξη 1

Να αποδείξετε ότι: $\neg(\neg P \vee Q) \vee (P \wedge Q) \equiv P$

$$\begin{aligned}
 \neg(\neg P \vee Q) \vee (P \wedge Q) &\equiv (\neg(\neg P) \wedge \neg Q) \vee (P \wedge Q) && \text{De Morgan} \\
 &\equiv (P \wedge \neg Q) \vee (P \wedge Q) && \text{Διπλή άρνηση} \\
 &\equiv P \wedge (\neg Q \vee Q) && \text{Νόμος επιμεριστικότητας} \\
 &\equiv P \wedge T && \neg Q \vee Q \text{ είναι ταυτολογία} \\
 &\equiv P && \text{Identity law}
 \end{aligned}$$

Παράδειγμα: Απόδειξη 2

Να αποδείξετε ότι: $(P \rightarrow Q) \wedge (P \rightarrow R) \equiv P \rightarrow (Q \wedge R)$

$$\begin{aligned}
 (P \rightarrow Q) \wedge (P \rightarrow R) &\equiv (\neg P \vee Q) \wedge (\neg P \vee R) && \text{Ορισμός συνεπαγωγής} \\
 &\equiv \neg P \vee (Q \wedge R) && \text{Νόμος επιμεριστικότητας} \\
 &\equiv P \rightarrow (Q \wedge R) && \text{Ορισμός συνεπαγωγής}
 \end{aligned}$$

Κεφάλαιο 7

7. Συναρτησιακή Πληρότητα

7.1 Ορισμός Συναρτησιακής Πληρότητας

Στην προτασιακή λογική έχουμε δει αρκετούς λογικούς τελεστές, όπως $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \oplus$.

Ερώτηση: Χρειαζόμαστε πραγματικά όλους αυτούς τους τελεστές για να εκφράσουμε κάθε λογική πρόταση;

Απάντηση: Όχι, δεν είναι απαραίτητοι όλοι. Ορισμένα σύνολα τελεστών αρκούν για να κατασκευάσουμε οποιαδήποτε σύνθετη λογική πρόταση. Αυτή η ιδιότητα ονομάζεται συναρτησιακή πληρότητα.

Ορισμός: Συναρτησιακή Πληρότητα

Μια συλλογή λογικών τελεστών λέγεται **συναρτησιακά πλήρης** όταν κάθε σύνθετη πρόταση της προτασιακής λογικής μπορεί να γραφτεί με χρήση **μόνο** των τελεστών αυτής της συλλογής.

Για να κατανοήσουμε τη συναρτησιακή πληρότητα, πρέπει πρώτα να θυμόμαστε ότι αρκετοί τελεστές μπορούν να εκφραστούν μέσω άλλων.

- **Συνεπαγωγή:** $P \rightarrow Q \equiv \neg P \vee Q$
- **Ισοδυναμία:** $P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$ ή $P \leftrightarrow Q \equiv (\neg P \vee Q) \wedge (\neg Q \vee P)$
- **Αποκλειστική Διάζευξη:** $P \oplus Q \equiv (P \wedge \neg Q) \vee (\neg P \wedge Q)$

Από τις παραπάνω ισοδυναμίες φαίνεται ότι οι τελεστές $\rightarrow, \leftrightarrow, \oplus$ δεν είναι απαραίτητο να θεωρούνται βασικοί, αφού μπορούν να εκφραστούν με πιο θεμελιώδεις τελεστές.

Παράδειγμα: Το σύνολο $\{\neg, \wedge, \vee\}$ συναρτησιακά πλήρες, διότι με αυτούς τους τρεις τελεστές μπορούμε να αναπαραστήσουμε οποιαδήποτε λογική συνάρτηση, δηλαδή οποιονδήποτε πίνακα αληθείας. Αυτό γίνεται επειδή κάθε λογική πρόταση μπορεί να κατασκευαστεί από τις γραμμές ενός πίνακα αληθείας στις οποίες η τελική τιμή είναι αληθής.

7.2 Πώς κατασκευάζουμε πρόταση από πίνακα αληθείας

Αν έχουμε έναν πίνακα αληθείας και θέλουμε να φτιάξουμε μια λογική έκφραση που να έχει ακριβώς την ίδια συμπεριφορά, ακολουθούμε την παρακάτω ιδέα:

- Κοιτάζουμε τις γραμμές στον πίνακα αληθείας όπου η τιμή είναι True.
- Για κάθε τέτοια γραμμή κατασκευάζουμε μία σύζευξη ($\wedge$) μεταβλητών ή αρνήσεων τους.
- Τέλος, ενώνουμε όλες αυτές τις συζεύξεις με διάζευξη ($\vee$).

Παράδειγμα: Έστω ότι έχουμε τρεις μεταβλητές P, Q, R και μια έξοδο S.

Θέλουμε να βρούμε μια πρόταση ισοδύναμη με το S χρησιμοποιώντας μόνο $\neg$, $\wedge$, $\vee$.

P	Q	R	S
T	T	T	F
T	T	F	T
T	F	T	F
T	F	F	T
F	T	T	F
F	T	F	F
F	F	T	T
F	F	F	F

Βήμα 1: Εντοπίζουμε τις γραμμές όπου $S = T$

$P = T, Q = T, R = F$

$P = T, Q = F, R = F$

$P = F, Q = F, R = T$

Βήμα 2: Γράφουμε μία σύζευξη για κάθε γραμμή

$P \wedge Q \wedge \neg R$

$P \wedge \neg Q \wedge \neg R$

$\neg P \wedge \neg Q \wedge R$

Βήμα 3: Ενώνουμε με διάζευξη

$S \equiv (P \wedge Q \wedge \neg R) \vee (P \wedge \neg Q \wedge \neg R) \vee (\neg P \wedge \neg Q \wedge R)$

Άρα κατασκευάσαμε μια λογική πρόταση που χρησιμοποιεί μόνο $\neg$, $\wedge$, $\vee$ και είναι ισοδύναμη με την έξοδο S. Αυτό δείχνει στην πράξη γιατί το σύνολο $\{\neg, \wedge, \vee\}$ είναι συναρτησιακά πλήρες.

Ερώτηση: Είναι το σύνολο $\{\neg, \wedge\}$ συναρτησιακά πλήρες;

Απάντηση: Ναι, το σύνολο $\{\neg, \wedge\}$ είναι συναρτησιακά πλήρες διότι αν και λείπει η διάζευξη $\vee$, μπορούμε να την κατασκευάσουμε με χρήση του νόμου του De Morgan:

$P \vee Q \equiv \neg(\neg P \wedge \neg Q)$

Άρα, αφού μπορούμε να φτιάξουμε και τη διάζευξη ($\vee$) μόνο με $\neg$ και $\wedge$, μπορούμε τελικά να εκφράσουμε οποιαδήποτε λογική πρόταση.

Δηλώνω ότι η παρούσα διατριβή και οποιοδήποτε συνοδευτικό λογισμικό αποτελούν δικό μου πρωτότυπο έργο, εκπονημένο σε πλήρη συμμόρφωση με τον Κώδικα Δεοντολογίας για την Έρευνα του Πανεπιστημίου Κύπρου (<https://www.ucy.ac.cy/legislation/kodikas-deontologias-kai-politikes/>), και με πλήρη ευθύνη εκ μέρους μου για την ακρίβεια, την ακεραιότητα και την εγκυρότητα όλου του περιεχομένου.

I declare that this dissertation and any accompanying software are my own original work, conducted in full compliance with the University of Cyprus Code of Conduct for Research (<https://www.ucy.ac.cy/legislation/kodikas-deontologias-kai-politikes/>), and with full accountability on my part for the accuracy, integrity, and validity of all content.