Εφαρμογή Υπηρεσιών Φοιτητή - academia

Εργασία στα πλαίσια του μαθήματος Βάσεις Δεδομένων (2024-2025) – Ομάδα 16.

Αλέξανδρος, ΑΚ, Κανούσης

Προπτυχιακός φοιτητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών Παν. Πατρών, a.kanousis@ac.upatras.gr

Γεώργιος, ΓΚ, Κόκκινος

Προπτυχιακός φοιτητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών Παν. Πατρών, up1083871@ac.upatras.gr

1. ΠΕΡΙΛΗΨΗ

Στα πλαίσια της διδασκαλίας του μαθήματος «Βάσεις δεδομένων» κατά το ακαδημαϊκό έτος 2024-2025, ως Ομάδα 16, αναλάβαμε τον σχεδιασμό μίας Βάσης Δεδομένων που θα υποστηρίζει μια εφαρμογή για υπηρεσίες φοιτητή. Συγκεκριμένα επιλέξαμε να επικεντρωθούμε στην δημιουργία μιας Βάσης SQLite, με την οποία ο χρήστης (φοιτητής), αλληλοεπιδρώντας μέσω προγράμματος γραφικής διεπαφής, θα μπορεί να αντλεί τα δεδομένα σχετικά με τα συγγράμματα που είναι διαθέσιμα προς επιλογή για το κάθε μάθημα που παρακολουθεί έχοντας και περαιτέρω χρήσιμες για αυτόν δυνατότητες, όπως η αξιολόγηση των συγγραμμάτων, η επιστροφή αυτών και η επιλογή του σημείου παραλαβής που τον εξυπηρετεί. Επιπλέον οι καθηγητές των μαθημάτων θα μπορούν, με την σειρά τους, να επιλέγουν τα διαθέσιμα αυτά συγγράμματα για το κάθε μάθημα που διδάσκουν.

Στόχος της εργασίας ήταν να παρέχουμε στον φοιτητή μία ολοκληρωμένη εμπειρία που θα περιλαμβάνει χρήσιμες λειτουργίες που θα θέλαμε εμείς ως φοιτητές του τμήματος να διαθέτη μία τέτοια εφαρμογή.

1. ΜΕΘΟΔΟΛΟΓΙΑ ΕΠΙΛΥΣΗΣ

Η μεθοδολογία που ακολουθήσαμε για την διεκπεραίωση της παρούσας εργασίας, ακολουθεί την δομή των παρακάτω υπό ενοτήτων. Αρχικά ασχοληθήκαμε με την περιγραφή του μικρόκοσμου της βάσης δεδομένων και με βάση αυτόν, ορίσαμε το εννοιολογικό μοντέλο και το σχεσιακό σχήμα. Στην συνέχεια δημιουργήσαμε την βάση δεδομένων και εισάγαμε δεδομένα σε αυτή. Τέλος ασχοληθήκαμε με την δημιουργία της γραφικής διεπαφής μέσω της οποίας οι χρήστες, θα μπορούν να διαβάζουν τα δεδομένα και να τα τροποποιούν.

* 1. Περιγραφή Μικρόκοσμου

Στην εφαρμογή που δημιουργήσαμε (academia) οι φοιτητές, λαμβάνουν τους προσωπικούς κωδικούς τους από την γραμματεία του τμήματος κατά την εισαγωγή τους σε αυτό. Με τους κωδικούς αυτούς μπορούν να εισέλθουν στην εφαρμογή και μέσω αυτής να τροποποιήσουν τον κωδικό πρόσβασης που τους δόθηκε, καθώς και άλλα προσωπικά τους στοιχεία.

Μέσω της εφαρμογής, οι φοιτητές, σε κάθε ακαδημαϊκό εξάμηνο, θα μπορούν να δηλώσουν τα συγγράμματα τα οποία επιθυμούν να παραλάβουν και να προσθέσουν στην βιβλιοθήκη τους για τα μαθήματα τα οποία παρακολουθούν. Σημαντική προσθήκη στην εφαρμογή μας είναι η δυνατότητα που δίνεται στους φοιτητές να επιλέξουν πιο σημείο παραλαβής τους εξυπηρετεί για κάθε ένα σύγγραμμα που θα επιλέξουν τελικά στην δήλωση τους.

Μια χρήσιμη λειτουργία της εφαρμογής είναι αυτή της βαθμολογίας των συγγραμμάτων. Οι φοιτητές θα μπορούν για κάθε ένα από τα συγγράμματα τα οποία διαθέτουν, να προσθέσουν μια βαθμολογία και ένα σχόλιο για το κάθε βιβλίο της βιβλιοθήκης τους. Αυτά τα δεδομένα στην συνέχεια θα είναι ορατά και στους υπόλοιπους φοιτητές που ενδιαφέρονται να επιλέξουν τα αντίστοιχα συγγράμματα. Αυτό θα είναι πολύ χρήσιμο για την επιλογή τους.

Μία ακόμα επιλογή που δίνεται μέσω της εφαρμογής, είναι η δυνατότητα της επιστροφής των συγγραμμάτων που οι φοιτητές διαθέτουν από προηγούμενα μαθήματα που παρακολούθησαν. Και πάλι η επιλογή του σημείου επιστροφής γίνεται από τους φοιτητές ανάλογα με το ποιο σημείο τους εξυπηρετεί καλύτερα. Όταν πραγματοποιηθεί μια επιστροφή οι φοιτητές θα πιστώνονται στον λογαριασμό τους τις πιστωτικές μονάδες που αντιστοιχούν στο σύγγραμμα που επέλεξαν για επιστροφή. Οι πιστωτικές μονάδες αυτές θα μπορούν να αξιοποιηθούν στην συνέχεια ώστε οι φοιτητές να προσθέσουν σε επόμενη δήλωση κάποιο ή κάποια συγγράμματα τα οποία δεν αντιστοιχούν απαραίτητα σε μαθήματα τα οποία παρακολουθούν αλλά τους ενδιαφέρει να αποκτήσουν.

Τέλος οι καθηγητές θα μπορούν πριν την αρχή κάθε ακαδημαϊκού έτους να επιλέξουν, μέσω της εφαρμογής, ποιο ή ποια συγγράμματα επιθυμούν να είναι διαθέσιμα για κάθε ένα από τα μαθήματα τα οποία είναι υπεύθυνοί να διδάξουν. Τα διαθέσιμα συγγράμματα από τα οποία θα μπορούν να επιλέξουν θα είναι συγκεκριμένα. Μια επιπλέον λειτουργία που δίνεται στους καθηγητές είναι η δυνατότητα της επιλογής ενός εκ των συγγραμμάτων που επέλεξαν για το κάθε μάθημα ως την προσωπική τους επιλογή και σύσταση προς τους φοιτητές. Η επιλογή αυτή θα είναι φανερή στην συνέχεια στους φοιτητές την ώρα που κάνουν την δήλωση τους, διευκολύνοντάς τους ως προς την απόφαση τους.

* 1. Εννοιολογικό μοντέλο: Διάγραμμα οντοτήτων – συσχετίσεων (ERD)

Βασιζόμενοι στην περιγραφή του μικρόκοσμου μας προχωρήσαμε στον σχεδιασμό του διαγράμματος οντοτήτων-συσχετίσεων (ERD) το οποίο αποτελείται από 10 οντότητες οι οποίες σε συνδυασμό με τις συσχετίσεις τους περιγράφουν την λύση που δώσαμε στο πρόβλημα μας. Η κάθε μια οντότητα αντιπροσωπεύει μια έννοια η οποία χαρακτηρίζεται από κάποια γνωρίσματα που την περιγράφουν. Στο διάγραμμα που παρουσιάζουμε δεν φαίνονται όλα τα γνωρίσματα αυτά, με στόχο η ανάγνωση του να είναι πιο εύκολη και κατανοητή.

A diagram of a flowchart

Description automatically generated

Εικόνα 1: ERD για περιγραφή μικρόκοσμου. Δημιουργήθηκε με εργαλείο [erdmaker](https://hci.ece.upatras.gr/erdmaker/)

Στο διάγραμμα αυτό, και συγκεκριμένα στις συσχετίσεις των οντοτήτων έχουν επιλεγεί οι κατάλληλοι δομικοί περιορισμοί στους λόγους πληθικότητας που ακολουθούν την σύμβαση Crow’s Foot Notation.

* 1. Σχεσιακό Σχήμα

Από το ERD που παρουσιάστηκε νωρίτερα, προχωρήσαμε στην δημιουργία του σχεσιακού σχήματος. Σε αυτό, όλες οι οντότητες αποτελούν έναν ξεχωριστό πίνακα με χρώμα κίτρινο και περιλαμβάνει όλα τα γνωρίσματα που τις περιγράφουν. Ακόμα στην περίπτωση των συσχετίσεων με πληθικότητες Μ:Ν, αυτές αναπαρίστανται με ξεχωριστό πίνακα και χρώμα μπλε. Στο σχήμα μπορούμε να δούμε τα πρωτεύοντα κλειδιά (σύνθετα ή απλά) του κάθε πίνακα και τα ξένα κλειδιά που ορίζουν τις συσχετίσεις.



Εικόνα 2: Σχεσιακό σχήμα. Δημιουργήθηκε με το εργαλείο [shema maker](https://150.140.186.221:1337/)

* 1. Δημιουργία Βάσης Δεδομένων

Σε αυτό το σημείο, αφού ολοκληρώσαμε ουσιαστικά τον σχεδιασμό της βάσης δεδομένων, προχωρήσαμε στην δημιουργία αυτής, σε γλώσσα SQL. Ακολουθώντας πιστά το σχεσιακό μοντέλο που παρουσιάσαμε νωρίτερα, με χρήση των κατάλληλων εντολών SQL, υλοποιήσαμε την βάση μας.

Ο κώδικας που δημιουργεί την βάση δεδομένων μας βρίσκεται στο αρχείο academia.sql. Στο αρχείο αυτό γίνεται αποκλειστικά η δημιουργία της δομής της βάσης και όχι η εισαγωγή των δεδομένων η οποία γίνεται από ξεχωριστά αρχεία, όπως θα δούμε και παρακάτω. Εισάγοντας το αρχείο sql στο DB Browser for SQLite (DBMS) δημιουργήσαμε την βάση δεδομένων (εικόνα 3) η οποία βρίσκεται στο αρχείο academia.db.

Για τον ορισμό κάθε πίνακα της βάσης, χρειάστηκε να προσθέσουμε όλα τα γνωρίσματα που φαίνονται στο σχεσιακό σχήμα και για κάθε ένα από αυτά τον κατάλληλο τύπο δεδομένων ανάλογα με την χρήση του. Χρησιμοποιώντας το NOT NULL δίπλα από τα γνωρίσματα, δεν επιτρέπουμε σε αυτά να μείνουν κενά κατά την προσθήκη μίας πλειάδας στην βάση.

Στην συνέχεια ορίζουμε το primary key (περισσότερα αν είναι σύνθετο) και τα foreign keys αν αυτά υπάρχουν. Εξασφαλίζουμε την ακεραιότητα των δεδομένων περιλαμβάνοντας τις κατάλληλες ενέργειες σε περίπτωση ενημέρωσης ή διαγραφής των τιμών των ξένων κλειδιών στους πίνακες «γονείς» με τις εντολές ON UPDATE CASCADE και ON UPDATE RESTRICT. Ένα παράδειγμα της διαδικασίας φαίνεται στην εικόνα 4, όπου ορίσαμε τον πίνακα STUDENT. Με αντίστοιχο τρόπο, όπως φαίνεται και στον κώδικα μας, δημιουργήσαμε και τους υπόλοιπους πίνακες.

A screenshot of a cell phone

Description automatically generated

Εικόνα 3:Δομή της βάσης στο DBMS

A screen shot of a computer program

Description automatically generated

Εικόνα 4: Δημιουργία πίνακα STUDENT (academia.sql)

* 1. Εισαγωγή Δεδομένων

Στην συνέχεια, ασχοληθήκαμε με την δημιουργία και την εισαγωγή των δεδομένων στην βάση. Αυτό έγινε με χρήση διάφορων εργαλείων τα οποία έκαναν την διαδικασία πιο εύκολη. Περισσότερα για αυτό θα αναφέρουμε σε ξεχωριστή ενότητα της παρούσας αναφοράς.

* 1. Δημιουργία Γραφικής Διεπαφής και SQL Queries

Το τελικό κομμάτι της εργασίας αφορούσε την δημιουργία της γραφικής διεπαφής μέσω της οποίας θα αξιοποιηθεί η βάση δεδομένων που δημιουργήσαμε. Για την γραφική διεπαφή χρησιμοποιήσαμε την Python και συγκεκριμένα την βιβλιοθήκη tkinter. Ο κώδικας δομήθηκε σε Κλάσεις που βρίσκονται σε ξεχωριστά αρχεία και οι οποίες δημιουργούν τα απαραίτητα Frames για κάθε μια λειτουργία της εφαρμογής. Η αρχικοποίηση της κλάσης academia γίνεται στο αρχικό αρχείο main.py το οποίο καλεί το πρώτο Frame (LoginFrame) που βρίσκεται στο αρχείο login\_page.py. Αφού ο χρήστης πραγματοποιήσει την είσοδο του στην εφαρμογή, ακολουθεί το βασικό WelcomeFrame που βρίσκεται στο αρχείο home\_page.py (ή home\_page\_prof.py) και από εκεί και πέρα ανάλογα με τις επιλογές του χρήστη καλούνται τα αντίστοιχα αρχεία που δημιουργούν τα αντίστοιχα Frames.

Αναλυτική παρουσίαση της χρήσης της εφαρμογής θα γίνει στην αντίστοιχη ενότητα της αναφοράς.

A screenshot of a computer program

Description automatically generated

Εικόνα 5: Αρχεία python της εφαρμογής

Σημαντικό κομμάτι τις εργασίας αυτής αποτέλεσε η δημιουργία των απαραίτητων SQL Queries με τα οποία μπορούσαμε να έχουμε πρόσβαση, να τροποποιούμε και να προσθέτουμε δεδομένα στην βάση δεδομένων. Τα Queries αυτά τα δοκιμάζαμε αρχικά στο πεδίο “Execute SQL” του DBMS και στην συνέχεια έγινε η ενσωμάτωση τους στα κατάλληλα αρχεία Python. Η υλοποίηση τους έγινε παράλληλα με την δημιουργία της γραφικής διεπαφής κάνοντας την διαδικασία πιο εύκολη. Αναλυτική περιγραφή των Queries αυτών θα γίνει στην αντίστοιχη ενότητα της αναφοράς.

A screenshot of a computer

Description automatically generated

Εικόνα 6: Επιλεγμένα Frames της εφαρμογής

1. ΔΕΔΟΜΕΝΑ

Για την ορθή λειτουργία και για την επίδειξη μιας ολοκληρωμένης εφαρμογής ήταν αναγκαίο να εισάγουμε στην βάση μας απαραίτητα δεδομένα με χρήση κατάλληλων εντολών SQL.

* 1. Δημιουργία Δεδομένων

Η επίλυση του προβλήματος επιλέξαμε να γίνει μόνο για το τμήμα στο οποίο ανήκουμε (Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών), ώστε το πλήθος των δεδομένων να είναι όσο το δυνατόν πιο διαχειρίσιμος. Για τα δεδομένα που χρησιμοποιήσαμε επιλέξαμε την χρήση εργαλείων τεχνητής νοημοσύνης (ChatGPT, Copilot) και βιβλιοθηκών την python (random). Τα μοναδικά δεδομένα που είναι πραγματικά είναι αυτά των μαθημάτων (πίνακας COURSE), όπου δημιουργήσαμε ένα csv αρχείο (ECE\_all\_courses.csv) που περιέχει για κάθε μάθημα του τμήματος τον κωδικό του, τον τίτλο του και το εξάμηνο στο οποίο διδάσκεται. Τα βιβλία και οι φοιτητές που επιλέξαμε να εισάγουμε στην βάση είναι τεχνητά και δημιουργήθηκαν σε μορφή αρχείων .csv (Books\_final.csv, Students.csv) μέσω του ChatGPT. Το κάθε βιβλίο (BOOK) περιέχει τα γνωρίσματα content (πίνακας περιεχομένων) και cover (εξώφυλλο). Για να προσθέσουμε αυτά τα δεδομένα, κατεβάσαμε 30 content αρχεία .pdf και 30 cover αρχείο .jpg (βρίσκονται στον φάκελο /data/cover\_toc), από πραγματικά βιβλία που διατίθενται προς επιλογή στους φοιτητές και για κάθε ένα βιβλίο που προσθέσαμε στην βάση μας, επιλέγαμε τυχαία ένα από αυτά τα αρχεία για τα περιεχόμενα και το εξώφυλλο χωρίς αυτά να αντιστοιχούν στον πραγματικό τίτλο και κωδικό ISBN του βιβλίου.

A screenshot of a computer program

Description automatically generated

Εικόνα 7: Αποσπάσματα των αρχείων .csv

Τα υπόλοιπα δεδομένα που εισάγαμε στην βάση δημιουργήθηκαν είτε τοπικά σε κάθε αρχείο (φάκελος imports) με χρήση του Copilot στο Visual Studio Code είτε με χρήση της βιβλιοθήκης random της python συνδυάζοντας τα είδη υπάρχοντα δεδομένα μεταξύ τους, όπως φαίνεται και στα αντίστοιχα αρχεία.

* 1. Εισαγωγή Δεδομένων

Για την εισαγωγή των δεδομένων που δημιουργήσαμε στην βάση μας, υλοποιήσαμε κάποια python αρχεία (φάκελος imports) που περιέχουν εντολές INSERT για να γεμίσουν τους κατάλληλους πίνακες. Κάποια από τα αρχεία αυτά όπως αναφέραμε και νωρίτερα περιέχουν και εντολές για την δημιουργία δεδομένων.

A screen shot of a computer program

Description automatically generated

Εικόνα 8: Εισαγωγή δεδομένων στον πίνακα MEMBER και STUDENT

A screenshot of a computer program

Description automatically generated

Εικόνα 9: Εισαγωγή δεδομένων στον πίνακα BOOK

Στις εικόνες 8 και 9 βλέπουμε κάποια παραδείγματα εισαγωγής δεδομένων στην βάση δεδομένων. Συγκεκριμένα στην εικόνα 8 (import\_members.py) βλέπουμε τον τρόπο με τον οποίο γίνεται η εισαγωγή των δεδομένων των φοιτητών ενώ στην εικόνα 9 (import\_books\_to\_db.py) βλέπουμε την εισαγωγή των δεδομένων των διαθέσιμων συγγραμμάτων στην βάση δεδομένων. Με αντίστοιχο τρόπο υλοποιούνται και τα υπόλοιπα αρχεία του φακέλου imports που καταθέσαμε.

Σημαντικό είναι να αναφέρουμε πως η σειρά με την οποία εισάγονται τα δεδομένα αυτά στην βάση είναι πολύ σημαντική ώστε να μην δημιουργούνται προβλήματα σχετικά με τα ξένα κλειδιά. Για παράδειγμα πρέπει πρώτα να είναι συμπληρωμένος ο πίνακας MEMBER και μετά να συμπληρωθεί ο πίνακας APPLICATION καθώς δεν γίνεται να συμπληρώσουμε την αίτηση ενός φοιτητή, τα στοιχεία του οποίου δεν είναι αποθηκευμένα στην βάση μας. Συγκεκριμένα η σειρά με την οποία πρέπει να τρέξουν τα αρχεία python για την εισαγωγή των δεδομένων στην βάση χωρίς να δημιουργηθεί κάποιο πρόβλημα είναι η εξής:

import\_courses\_to\_db.py

import\_books\_to\_db.py

import\_members.py

import\_applications.py

import\_course\_teaching.py

import\_participates.py

import\_pickup\_point.py

import\_rates.py

import\_return.py

1. SQL QUERIES σΤΟ ΠΡΟγραμμα

Η εφαρμογή που δημιουργήσαμε για την αλληλεπίδραση της με την βάση δεδομένων χρησιμοποιεί διάφορα SQL Queries μέσω των οποίων μπορούμε να διαβάσουμε δεδομένα, να τα τροποποιήσουμε ή να εισάγουμε νέα. Χρησιμοποιήσαμε τους εξής τέσσερις τύπους εντολών: INSERT INTO, UPDATE, SELECT και DELETE. Συγκεκριμένα χρησιμοποιήσαμε 45 εντολές SELECT, 7 εντολές INSERT, 7 εντολές UPDATE και 6 εντολές DELETE. Προφανώς δεν γίνεται να αναλυθούν ένα προς ένα όλα τα Queries αυτά. Γι’ αυτό επιλέξαμε να παρουσιάσουμε κάποια ενδεικτικά παραδείγματα από κάθε κατηγορία. Με αντίστοιχο τρόπο έχουν υλοποιηθεί και τα υπόλοιπα.

SELECT:

Στην εικόνα 10 βλέπουμε τον τρόπο με τον οποίο επιλέγεται από την βάση δεδομένων το σύνολο των συγγραμμάτων που διαθέτει ο φοιτητής στην βιβλιοθήκη του από δηλώσεις που έχει κάνει σε προηγούμενα εξάμηνα. Στο πρώτο SELECT βλέπουμε πως γίνεται η επιλογή των ISBN και των τίτλων των βιβλίων. Έχουμε συσχέτιση μεταξύ τριών πινάκων και γι’ αυτό τον λόγο χρησιμοποιούμε τρία διαδοχικά INNER JOIN έχοντας την συνθήκη το εξάμηνο αναζήτησης να είναι μικρότερο από το τωρινό εξάμηνο στο οποίο βρίσκεται ο φοιτητής. Στο δεύτερο SELECT επιλέγουμε τα ISBN και τους αντίστοιχους τίτλους των βιβλίων για τους οποίους ο φοιτητής έχει πραγματοποιήσει επιστροφή. Τέλος αφαιρούμε τις δυο αυτές λίστες για να πάρουμε τελικά το πλήθος των συγγραμμάτων που διαθέτει πραγματικά ο φοιτητής στην βιβλιοθήκη του αυτή την στιγμή. Έτσι τελικά διαμορφώνεται το τελικό “My Book Library” Frame.

A screenshot of a computer

Description automatically generated

Εικόνα 10: Επιλογή από την βάση δεδομένων των συγγραμμάτων που διαθέτει στην βιβλιοθήκη του ο φοιτητής

INSERT INTO:

Στην εικόνα 11 βλέπουμε πως αποθηκεύεται στην βάση δεδομένων η βαθμολογία και το σχόλιο που καταθέτει ένας φοιτητής για ένα από τα συγγράμματα που διαθέτει στην βιβλιοθήκη του.

A screenshot of a computer

Description automatically generated

Εικόνα 11: Εισαγωγή στην βάση δεδομένων μιας βαθμολογίας συγγράμματος

UPDATE:

Στην εικόνα 12, βλέπουμε ένα παράδειγμα τροποποίησης των προσωπικών δεδομένων του χρήστη στην βάση. Προφανώς στον κώδικα υπάρχει και ο κατάλληλος έλεγχος της μορφής των νέων δεδομένων, όπως αν το Νew Password και το Confirm New Password έχουν την ίδια τιμή, όμως εδώ επικεντρωνόμαστε στα Queries που εκτελούν την αλλαγή των δεδομένων αυτών στην βάση δεδομένων.

A screenshot of a computer

Description automatically generated

Εικόνα 12: Αλλαγή δεδομένων του χρήστη στην βάση δεδομένων

DELETE:

Η χρήση της εντολής DELETE γίνεται στα σημεία που ο χρήστης θέλει να τροποποιήσει μία είδη υπάρχουσα δήλωση που έχει πραγματοποιήσει. Για παράδειγμα έστω ένας καθηγητής ο οποίος έχει επιλέξει τα συγγράμματα που είναι διαθέσιμα για επιλογή στο μάθημα που θα διδάξει αυτό το έτος αλλά θέλει να αλλάξει αυτή την δήλωση. Έτσι, πρώτού κατατεθεί η νέα λίστα επιλογών στην βάση δεδομένων πρέπει να διαγραφούν οι προηγούμενες εγγραφές. Αυτό γίνεται με την χρήση του Query που φαίνεται στην εικόνα 13. Προφανώς υπάρχουν και εντολές INSERT που ακολουθούν για την εγγραφή στην βάση δεδομένων της νέας επιλογής των συγγραμμάτων από τον καθηγητή, αλλά εδώ επικεντρωνόμαστε στην εντολή DELETE.

Μια αντίστοιχη λογική υπάρχει και στο σημείο που ένας φοιτητής θέλει να αλλάξει την δήλωση των συγγραμμάτων που έχει πραγματοποιήσει (προφανώς εντός της περιόδου δήλωσης που ανακοινώνεται).



Εικόνα 13: Διαγραφή προηγούμενης επιλογής από την βάση δεδομένων

1. ΛΕΙΤΟΥΡΓΙΑ ΟΜΑΔΑΣ - Χρονοδιαγραμμα

Για την ολοκλήρωση της παρούσας εργασίας με επιτυχία, εργαστήκαμε καθ’ όλη την διάρκεια του εξαμήνου ακολουθώντας το χρονοδιάγραμμα του μαθήματος.

Ξεκινώντας από την ημέρα που μας ανατέθηκε το θέμα, πραγματοποιήσαμε την πρώτη συνάντηση όπου αποφασίσαμε την κατεύθυνση που θέλουμε να πάρουμε. Αφού εξοικειωθήκαμε με τις έννοιες και τα εργαλεία που συντελούν το ERD, προχωρήσαμε από κοινού στον ορισμού του μικρόκοσμου και στον σχεδιασμού του εννοιολογικού μοντέλου.

Στην συνέχεια ακολούθησε η ενδιάμεση παρουσίαση όπου παρουσιάσαμε την μέχρι τότε δουλειά μας και αξιοποιώντας τα σχόλια των διδασκόντων, κάναμε τις απαραίτητες αλλαγές στο ERD. Στην συνέχεια δημιουργήσαμε και πάλι από κοινού το σχεσιακό σχήμα που είδαμε νωρίτερα και προχωρήσαμε στην σύνταξη των κατάλληλων εντολών SQL για την δημιουργία της βάσης δεδομένων.

Έπειτα έπρεπε να χωρίσουμε τις αρμοδιότητες ώστε να δουλεύουμε παράλληλα και ο καθένας με τον δικό του ρυθμό. Εξάλλου, όπως αναφέρθηκε και νωρίτερα, ο κώδικας μας είναι δομημένος σε κλάσεις οι οποίες δημιουργούν Frames και ανήκουν σε ένα κεντρικό παράθυρο. Έτσι ο καθένας ασχολήθηκε με τα Frames που ήταν υπεύθυνος υλοποιώντας παράλληλα τα Queries που τα αποτελούν. Η δουλειά τελικά ήταν πολύ εύκολο να ενσωματωθεί και να προκύψει το τελικό αποτέλεσμα.

Συγκεκριμένα:

Ο ΑΚ, ανέλαβε αρχικά την δημιουργία και την εισαγωγή των δεδομένων στην βάση. Στην συνέχεια ήταν υπεύθυνος για την υλοποίηση του Login Frame, την παρουσίαση των δεδομένων του χρήστη καθώς και την τροποποίηση αυτών. Ακόμη υλοποίησε το παράθυρο σχετικά με την καταχώρηση αξιολογήσεων για τα συγγράμματα και τέλος πραγματοποίηση την δήλωση των διαθέσιμων συγγραμμάτων από τον καθηγητή.

Ο ΓΚ, από την πλευρά του, ασχολήθηκε με την υλοποίηση του παράθυρου με το οποίο οι φοιτητές μπορούν να πραγματοποιούν τις δηλώσεις για τα συγγράμματα που θέλουν να αποκτήσουν, ενώ υλοποίησε και το παράθυρο που οι φοιτητές πραγματοποιούν επιστροφές των συγγραμμάτων που διαθέτουν στην κατοχή τους.

Καθ’ όλη την διάρκεια του εξαμήνου πραγματοποιούσαμε τακτά meeting μέσω της πλατφόρμας του Discord όπου μπορούσαμε να συζητάμε την πρόοδο που ο καθένας έχει κάνει έχοντας έτσι και οι δύο συνολική εποπτεία και κατανόηση του κώδικα, παίρνοντας τις κρίσιμες αποφάσεις από κοινού.

Αξιοποιήσαμε επιπλέον το OneDrive για τον διαμοιρασμό των απαραίτητων αρχείων κάτι που βοήθησε πολύ την διαδικασία.

Τέλος συντάξαμε την παρούσα αναφορά και την παρουσίαση της εργασίας μας.

1. ΧΡΗΣΗ ΕΦΑΡΜΟΓΗΣ

Στο σημείο αυτό, ακολουθεί αναλυτική περιγραφή όλων των λειτουργιών της εφαρμογής. Για διευκόλυνση σας, μπορείτε να παρακολουθήσετε και το [demo video](https://youtu.be/Jg19dTnPrB0) στο οποίο φαίνονται όλες οι δυνατότητες της εφαρμογής που δημιουργήσαμε.

* 1. Χρήση από τον καθηγητή

Ο καθηγητής εισέρχεται στην εφαρμογή χρησιμοποιώντας τα προσωπικά του διαπιστευτήρια. Μετά την επιτυχημένη είσοδο, οδηγείται στην κεντρική σελίδα (homepage), όπου εμφανίζονται τρία βασικά κουμπιά:

Προφίλ (Profile): Πατώντας το κουμπί αυτό, ο καθηγητής μεταφέρεται στη σελίδα του προφίλ του, όπου μπορεί να δει πληροφορίες όπως ονοματεπώνυμο, email, τηλέφωνο, διεύθυνση κ.ά. Από τη σελίδα του προφίλ, μπορεί είτε να επιστρέψει στην κεντρική σελίδα είτε να επιλέξει να επεξεργαστεί τα δεδομένα του πατώντας το κουμπί "Edit". Η επιλογή αυτή τον μεταφέρει σε ειδική σελίδα επεξεργασίας δεδομένων. Μετά την πραγματοποίηση αλλαγών, τα νέα δεδομένα αποθηκεύονται πατώντας το κουμπί "Save".

Επιλογή Συγγραμμάτων για Μαθήματα (Choose Books for Courses): Πατώντας το κουμπί αυτό, ο καθηγητής βλέπει μια λίστα με τα μαθήματα που διδάσκει, όπου εμφανίζονται ο κωδικός του μαθήματος, το όνομα, και το εξάμηνο διδασκαλίας. Επιλέγοντας ένα μάθημα από τη λίστα, ανοίγει μια νέα σελίδα που περιέχει όλα τα διαθέσιμα βιβλία της βάσης δεδομένων της εφαρμογής, με πληροφορίες όπως ο τίτλος και το ISBN. Στη σελίδα αυτή, ο καθηγητής μπορεί να επιλέξει ένα ή περισσότερα συγγράμματα για το συγκεκριμένο μάθημα και να ορίσει ποιο από αυτά προτείνει προσωπικά. Η επιλογή του αποθηκεύεται πατώντας το κουμπί "Finish". Πατώντας το κουμπί "Back", επιστρέφει στη λίστα με τα μαθήματα που διδάσκει, ώστε να επαναλάβει τη διαδικασία για άλλο μάθημα, αν το επιθυμεί.

Έξοδος (Logout): Πατώντας το κουμπί αυτό, ο καθηγητής αποσυνδέεται από την εφαρμογή και επιστρέφει στην αρχική σελίδα σύνδεσης.

Η διαδικασία έχει σχεδιαστεί ώστε να είναι απλή και εύχρηστη, προσφέροντας στον καθηγητή όλα τα απαραίτητα εργαλεία για τη διαχείριση των μαθημάτων και των προσωπικών του δεδομένων.

* 1. Χρήση από τον φοιτητή

Ο φοιτητής εισέρχεται στην εφαρμογή χρησιμοποιώντας τα προσωπικά του διαπιστευτήρια. Μετά την επιτυχημένη είσοδο, οδηγείται στην κεντρική σελίδα (homepage), όπου εμφανίζονται τα εξής βασικά κουμπιά:

Προφίλ (Profile): Πατώντας το κουμπί αυτό, ο φοιτητής μεταφέρεται στη σελίδα του προφίλ του, όπου μπορεί να δει πληροφορίες όπως ονοματεπώνυμο, email, τηλέφωνο, διεύθυνση κ.ά. Από τη σελίδα του προφίλ, μπορεί είτε να επιστρέψει στην κεντρική σελίδα είτε να επιλέξει να επεξεργαστεί τα δεδομένα του πατώντας το κουμπί "Edit". Η επιλογή αυτή τον μεταφέρει σε ειδική σελίδα επεξεργασίας δεδομένων. Μετά την πραγματοποίηση αλλαγών, τα νέα δεδομένα αποθηκεύονται πατώντας το κουμπί "Save".

Αίτηση Συγγραμμάτων (Make an Application): Πατώντας το κουμπί αυτό, αν ο φοιτητής δεν έχει υποβάλει ήδη δήλωση για το τρέχον εξάμηνο, οδηγείται κανονικά στη σελίδα δήλωσης, όπως περιγράφεται παρακάτω. Αν όμως έχει ήδη υποβάλει δήλωση, εμφανίζεται ένα νέο παράθυρο, το οποίο δείχνει την προηγούμενη δήλωση και τον ρωτά αν επιθυμεί να προχωρήσει στη δημιουργία νέας δήλωσης. Σε αυτή την περίπτωση, ενημερώνεται ότι η προηγούμενη δήλωση θα διαγραφεί. Εάν ο φοιτητής επιλέξει να προχωρήσει, τυχόν επιπλέον βιβλία που είχε "αγοράσει" με πιστωτικές μονάδες στην προηγούμενη δήλωση θα αφαιρεθούν, και οι αντίστοιχες μονάδες θα επιστραφούν στο λογαριασμό του.

Στη σελίδα δήλωσης, εμφανίζονται δύο λίστες. Στην πρώτη λίστα περιλαμβάνονται τα μαθήματα του τρέχοντος εξαμήνου, ενώ στη δεύτερη όλα τα μαθήματα της βάσης δεδομένων. Στην πρώτη φάση της διαδικασίας, ο φοιτητής δηλώνει συγγράμματα για τα μαθήματα του τρέχοντος εξαμήνου. Πατώντας το κουμπί "i" δίπλα σε κάθε βιβλίο, εμφανίζεται ένα παράθυρο με λεπτομέρειες όπως ISBN, εξώφυλλο, συγγραφέας, σελίδες και ο μέσος όρος βαθμολογίας του βιβλίου. Κάνοντας hover στη βαθμολογία, εμφανίζεται λίστα με τις κριτικές άλλων φοιτητών. Στη δεύτερη φάση, ο φοιτητής μπορεί να δηλώσει επιπλέον βιβλία, εφόσον διαθέτει τις απαραίτητες πιστωτικές μονάδες, οι οποίες υπολογίζονται πατώντας το κουμπί "Check Credits". Με το κουμπί "Finish", ελέγχονται οι κανόνες της δήλωσης και αν πληρούνται, εμφανίζεται η τελική λίστα με τα επιλεγμένα βιβλία. Ο φοιτητής επιλέγει σημείο παραλαβής για κάθε βιβλίο μέσω dropdown με τα διαθέσιμα βιβλιοπωλεία, και ολοκληρώνει τη διαδικασία πατώντας το κουμπί "Complete Application".

Προγραμματισμός Επιστροφής (Plan a Return): Πατώντας το κουμπί αυτό, ο φοιτητής βλέπει μια λίστα με τα βιβλία που έχει παραλάβει στο παρελθόν. Δίπλα σε κάθε βιβλίο υπάρχει dropdown με διαθέσιμα βιβλιοπωλεία για επιστροφή. Με το κουμπί "Complete Application", ολοκληρώνεται η δήλωση επιστροφής, και οι αντίστοιχες πιστωτικές μονάδες προστίθενται στο λογαριασμό του φοιτητή.

Βαθμολόγηση Βιβλίου (Rate a Book): Ο φοιτητής βλέπει μια λίστα με τα βιβλία που έχει παραλάβει στο παρελθόν. Πατώντας σε κάποιο βιβλίο, εμφανίζεται μια κάρτα με πληροφορίες όπως εξώφυλλο, συγγραφέας κ.λπ., καθώς και δύο πεδία: ένα dropdown για τη βαθμολογία (από 0 έως 10, με βήμα 0.5) και ένα πεδίο κειμένου για σχόλια. Με το κουμπί "Submit", η βαθμολογία και το σχόλιο αποθηκεύονται στη βάση δεδομένων.

Έξοδος (Logout): Πατώντας το κουμπί αυτό, ο φοιτητής αποσυνδέεται από την εφαρμογή και επιστρέφει στην αρχική σελίδα σύνδεσης.

Η διαδικασία χρήσης της εφαρμογής έχει σχεδιαστεί για να προσφέρει ευκολία και λειτουργικότητα, καλύπτοντας πλήρως τις ανάγκες των φοιτητών για διαχείριση δηλώσεων, επιστροφών και αξιολογήσεων συγγραμμάτων.

REFERENCES

1. Ramez Elmasri and Shamkant B.Navathe. 2016. Fundamentals of Darabase Systems, Seventh Edition. Addison-Wesley

A  ΠΑραρτηματα

A.1 Οδηγίες εγκατάστασης και εκτέλεσης

Για να λειτουργήσει η εφαρμογή σωστά, πρέπει να γίνουν τα παρακάτω βήματα:

1. Να υπάρχει κάποια έκδοση της Python εγκατεστημένη στον υπολογιστή
2. Να γίνουν εγκατάσταση οι αναγκαίες βιβλιοθήκες με τις παρακάτω εντολές:

pip install pillow

pip install datetime

pip install Image

pip install messagebox

1. To τρέξιμο του κώδικα να γίνει στον σωστό φάκελο που θα περιέχει όλα τα αρχεία. Συγκεκριμένα, πρέπει να ανοίξετε έναν φάκελο στο IDE της επιλογής σας, που να περιέχει όλα τα αρχεία.
2. Για να χρησιμοποιήσετε την Βάση δεδομένων που έχουμε δημιουργήσει με όλα τα δεδομένα φορτωμένα, κατεβάστε το αρχείο academia.db από το [Google Drive](https://drive.google.com/drive/folders/14t3iYWmINEeDy4GtP4KpzJpK8wQP6uKD?usp=sharing) και προσθέστε το αρχείο στον φάκελο που έχετε ανοίξει νωρίτερα στο IDE.
3. Για την εκτέλεση του προγράμματος τρέξτε το αρχείο main.py και το πρόγραμμα πρέπει να ανοίξει.

Για να γίνει δοκιμή της εφαρμογής μπορούν να χρησιμοποιηθούν οι παρακάτω demo λογαριασμοί:

Για λειτουργίες καθηγητή:

member\_id : 1111111

password: 1111

Για λειτουργίες φοιτητή:

Φοιτητής 9ο εξάμηνο:

member\_id : 1

password: 1234

Φοιτητής 3ο εξάμηνο:

member\_id : 2

password: 1235

Ο κώδικας που υλοποιήθηκε για την εργασία αυτή, μπορεί να βρεθεί στο [GitHub](https://github.com/kanousis/DataBases_project_team16/tree/main).