

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΡΟΗ Λ, ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

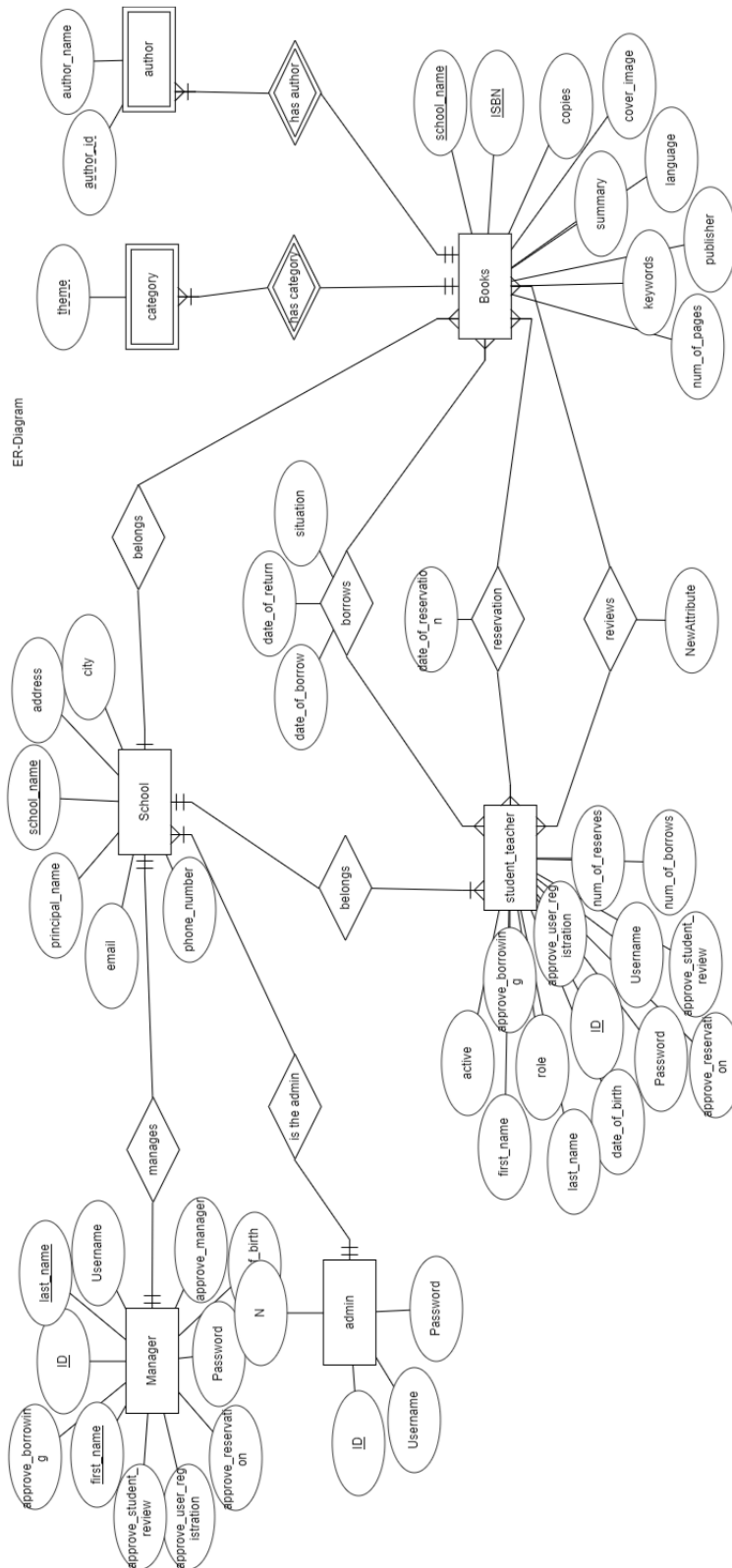


# BiblioTECH

Αναφορά  
Εξαμηνιαίας Εργασίας  
Ομάδα: 2

**Μέλη:**

Γκίνη Δήμητρα 03120128  
Θάνου Παρασκευή 03120225  
Σαμπάνη Μαρία 03120117



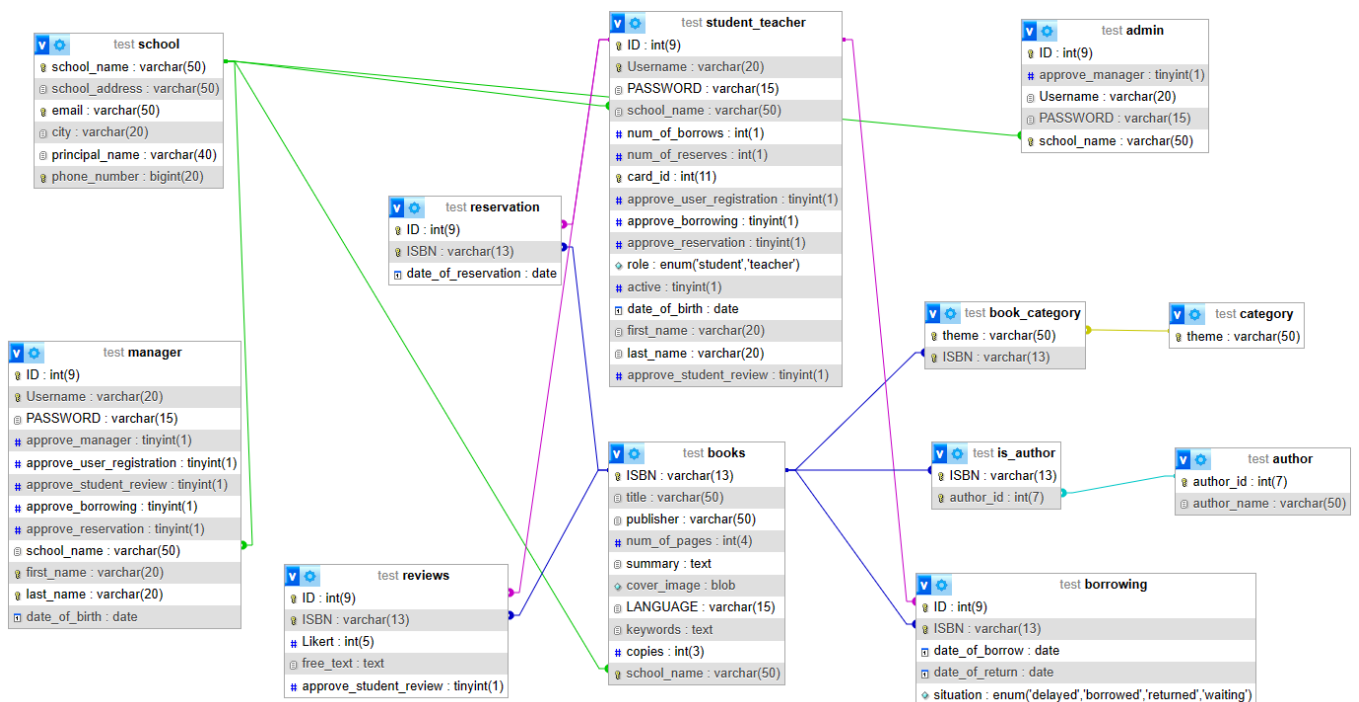
Από την περιγραφή της άσκησης καταλήξαμε στο παραπάνω ER διάγραμμα το οποίο περιλαμβάνει τις οντότητες : school, student, teachers, admin, manager, books, category, author από τις οποίες οι 2 τελευταίες είναι weak entities των books. Αυτό συμβαίνει καθώς δεν μπορεί να υπάρξει ούτε συγγραφέας ούτε κατηγορία χωρίς το αντίστοιχο βιβλίο.

Έχουμε επίσης τις παρακάτω σχέσεις:

- 1) Manages (one-to-one) μεταξύ school - manager. Κάθε σχολείο έχει μόνο έναν χειριστή και ένας χειριστής μπορεί να έχει την εποπτεία μόνο ενός σχολείου. Τόσο ο χειριστής όσο και το σχολείο συνδέονται με ολική συμμετοχή (total participation) καθώς δεν μπορεί να υπάρξει σχολείο χωρίς τον χειριστή του. Αντίστοιχα κανένας manager δεν μπορεί να εγγραφεί στο σύστημα χωρίς το σχολείο του.
- 2) Is the admin (many-to-one) μεταξύ school - admin. Ο γενικός διαχειριστής είναι υπεύθυνος για όλα τα σχολεία που είναι εγγεγραμμένα στο σύστημα. Έτσι, πολλά σχολεία έχουν μόνο έναν κεντρικό διαχειριστή, ο οποίος συνδέεται με περισσότερα από ένα σχολεία. Η σχέση είναι και σε αυτή την περίπτωση ολική καθώς δεν μπορεί να υπάρξει σχολείο που δεν διευθύνεται από τον κεντρικό διαχειριστή.
- 3) Belongs (many-to-one) μεταξύ student\_teacher – school. Πολλοί μαθητές και εκπαιδευτικοί (χρήστη) ανήκουν σε 1 σχολείο, και κάθε σχολείο έχει περισσότερους από έναν χρήστες. Και εδώ οι οντότητες συμμετέχουν στη σχέση ολικά, εφόσον κάθε σχολείο πρέπει να έχει υποχρεωτικά τουλάχιστον έναν μαθητή-εκπαιδευτικό και καθένας από αυτούς ανήκει σε ακριβώς 1 σχολείο.
- 4) Belong (many-to-many) μεταξύ books – school. Πολλά βιβλία ανήκουν σε ένα σχολείο και το ίδιο βιβλίο μπορεί να υπάρχει διαθέσιμο σε περισσότερα από 1 σχολεία.
- 5) Borrows (many-to-many), Reserves (many-to-many) και Reviews (many-to-many) μεταξύ books – student\_teacher. Ένας χρήστης μπορεί να δανειστεί, να κρατήσει και να αξιολογήσει περισσότερα από ένα βιβλία και αντίστοιχα ένα βιβλίο μπορεί να δανειστεί, να κρατηθεί και να αξιολογηθεί από πολλούς χρήστες.

- 6) Has category, has author (weak relationships many-to-many) μεταξύ books και category, author αντίστοιχα. Ένα βιβλίο μπορεί να ανήκει σε πολλές κατηγορίες και να έχει περισσότερους από έναν συγγραφείς. Αντίστοιχα, μια κατηγορία χαρακτηρίζει πολλά βιβλία και ένας συγγραφέας μπορεί να έχει γράψει πολλά βιβλία.

## 1.2 Σχεσιακό Διάγραμμα



Για την εξαγωγή του παραπάνω σχεσιακού διαγράμματος ξεκινήσαμε από το ER και μετατρέψαμε όλες τις οντότητες και μερικές από τις σχέσεις σε πίνακες. Οι σχέσεις many-to-many έγιναν ξεχωριστοί πίνακες με foreign keys τα primary keys των οντοτήτων που συνδέουν. Έτσι προέκυψαν οι πίνακες borrowing, review, reservation, is\_author και book\_category. Όσον αφορά τις σχέσεις one-to-many, many-to-one εξαλείφονται με την εισαγωγή του primary key της οντότητας από την πλευρά one, ως foreign key στην πλευρά many. Για τον λόγο αυτό έχουμε τοποθετήσει το school\_name ως foreign key στους πίνακες student\_teacher και

admin. Τέλος, όσον αφορά την σχέση one-to-one αναπαρίσταται με ένα foreign key σε οποιαδήποτε από τις 2 συσχετιζόμενες οντότητες. Έτσι, τοποθετήσαμε στον πίνακα manager το school\_name, που είναι το primary key του πίνακα school.

## 2. DDL και DML script

### 2.1 DDL script

```
DDL.sql
1 CREATE TABLE IF NOT EXISTS school(
2     school_name VARCHAR(50) NOT NULL,
3     school_address VARCHAR(50) NOT NULL,
4     email VARCHAR(50) NOT NULL,
5     city VARCHAR(20) NOT NULL,
6     principal_name VARCHAR(40) NOT NULL,
7     phone_number bigint(20) NOT NULL,
8     UNIQUE(email),
9     UNIQUE(phone_number),
10    PRIMARY KEY(school_name)
11 );
```

Λαμβάνοντας υπόψιν τους περιορισμούς κλειδιών, στον πίνακα των σχολείων έχουμε ορίσει ως primary key το όνομα του σχολείου, καθώς δεν γίνεται να υπάρχουν δύο σχολεία με το ίδιο όνομα.

Λαμβάνοντας υπόψιν τους περιορισμούς ακεραιότητας πεδίου τιμών για όλα τα attributes που έχουν χαρακτήρες στο όνομα τους, έχουμε χρησιμοποιήσει τον τύπο VARCHAR. Στο όνομα του σχολείου έχουμε δώσει μέγιστο μέγεθος 50, όπως και στη διεύθυνση του σχολείου και στο email ενώ στο όνομα της πόλης μέγιστο μέγεθος 20 και στον διευθυντή του σχολείου 40. Ο αριθμός τηλεφώνου του σχολείου έχει τύπο ακεραίου (int) και μέγιστο μέγεθος 11.

Το email και το τηλέφωνο του σχολείου είναι unique διότι δεν είναι δυνατό δύο διαφορετικά σχολεία να έχουν ίδιο κάτι από τα δύο.

Για όλα τα παραπάνω πεδία έχουμε ορίσει ότι δεν γίνεται να είναι κενά (NOT NULL), καθώς όλα είναι στοιχεία απαραίτητα για την ένταξη ενός σχολείου στην εφαρμογή μας.

```

13 CREATE TABLE IF NOT EXISTS student_teacher(
14     ID int(9) NOT NULL AUTO_INCREMENT,
15     Username VARCHAR(20) NOT NULL,
16     Password VARCHAR(15) NOT NULL,
17     school_name VARCHAR(50) NOT NULL,
18     num_of_borrows int(1) NOT NULL DEFAULT 0,
19     num_of_reserves int(1) NOT NULL DEFAULT 0,
20     approve_user_registration BOOLEAN,
21     approve_borrowing BOOLEAN,
22     approve_reservation BOOLEAN,
23     role enum('student','teacher'),
24     active BOOLEAN,
25     date_of_birth date NOT NULL,
26     first_name VARCHAR(20) NOT NULL,
27     last_name VARCHAR(20) NOT NULL,
28     approve_student_review BOOLEAN DEFAULT NULL,
29     PRIMARY KEY(ID),
30     UNIQUE(Username),
31     CONSTRAINT school_name_fk FOREIGN KEY(school_name) REFERENCES school(school_name)
32 );

```

Ομοίως με πριν, στον πίνακα των μαθητών-καθηγητών έχουμε ορίσει ως primary key το ID και έχουμε δώσει unique values στο username. Αυτό το κάναμε προκειμένου οι χρήστες να ξεχωρίζουν μεταξύ τους από το ID τους και δυο χρήστες να μην μπορούν να έχουν το ίδιο username, αφού μέσω αυτού και του password πραγματοποιείται το login.

Τα πεδία που ορίσαμε ως varchar πρόκειται για string μεταβλητού μεγέθους.

Τα πεδία που ορίσαμε ως INT, πρόκειται για ακέραιους αριθμούς.

Τα πεδία που ορίσαμε ως BOOLEAN παίρνουν τιμές μόνο 1 και 0 και πρόκειται για εγκρίσεις.

Ο ρόλος είναι τύπου enum καθώς παίρνει προκαθορισμένες τιμές.

Το πεδίο που ορίσαμε με τύπο date πρόκειται για ημερομηνία.

Για όλα τα παραπάνω πεδία εκτός από τα Boolean attributes, έχουμε ορίσει ότι δεν γίνεται να είναι κενά (NOT NULL), καθώς όλα είναι στοιχεία απαραίτητα για την ένταξη ενός σχολείου στην εφαρμογή μας.

```

35 CREATE TABLE IF NOT EXISTS admin (
36     ID int(9) NOT NULL,
37     approve_manager BOOLEAN,
38     Username VARCHAR(20) NOT NULL,
39     Password VARCHAR(15) NOT NULL,
40     school_name VARCHAR(50) NOT NULL,
41     PRIMARY KEY(ID,school_name),
42     FOREIGN KEY(school_name) REFERENCES school(school_name)
43 );

```

Στον πίνακα του διαχειριστή έχουμε ορίσει ως primary key set το ID και το school\_name. Αυτό το κάναμε έτσι ώστε ο μοναδικός admin να ελέγχει όλα τα σχολεία.

Τα πεδία που ορίσαμε ως varchar πρόκειται για string μεταβλητού μεγέθους.

Τα πεδία που ορίσαμε ως INT, πρόκειται για ακέραιους αριθμούς.

Τα πεδία που ορίσαμε ως BOOLEAN παίρνουν τιμές μόνο 1 και 0 και πρόκειται για εγκρίσεις.

Για όλα τα παραπάνω πεδία εκτός από τα Boolean attributes, έχουμε ορίσει ότι δεν γίνεται να είναι κενά (NOT NULL), καθώς όλα είναι στοιχεία απαραίτητα για την ένταξη ενός σχολείου στην εφαρμογή μας.

```
45 CREATE TABLE IF NOT EXISTS manager(  
46     ID int(9) NOT NULL AUTO_INCREMENT,  
47     Username VARCHAR(20) NOT NULL,  
48     Password VARCHAR(15) NOT NULL,  
49     approve_manager BOOLEAN,  
50     approve_user_registration BOOLEAN,  
51     approve_student_review BOOLEAN,  
52     approve_borrowing BOOLEAN,  
53     approve_reservation BOOLEAN,  
54     school_name VARCHAR(50) NOT NULL,  
55     first_name VARCHAR(20) NOT NULL,  
56     last_name VARCHAR(20) NOT NULL,  
57     date_of_birth date NOT NULL,  
58     PRIMARY KEY(ID,first_name,last_name),  
59     UNIQUE(Username),  
60     CONSTRAINT school_name__fk FOREIGN KEY(school_name) REFERENCES school(school_name) ON UPDATE CASCADE  
61 );  
62
```

Στον πίνακα του χειριστή ορίσαμε ως primary key set το ID, το first\_name και το last\_name. Αυτό το κάναμε προκειμένου οι χειριστές να ξεχωρίζουν μεταξύ τους μέσω των στοιχείων αυτών. Δυο χειριστές μπορούν να έχουν ίδιο first\_name και last\_name σε περίπτωση συνωνυμίας. Επίσης, έχουμε ορίσει το username ως unique value, έτσι ώστε δυο χειριστές να μην μπορούν να έχουν το ίδιο username.

Τα πεδία που ορίσαμε ως varchar πρόκειται για string μεταβλητού μεγέθους.

Τα πεδία που ορίσαμε ως INT, πρόκειται για ακέραιους αριθμούς.

Τα πεδία που ορίσαμε ως BOOLEAN παίρνουν τιμές μόνο 1 και 0 και πρόκειται για εγκρίσεις.

Για όλα τα παραπάνω πεδία εκτός από τα Boolean attributes, έχουμε ορίσει ότι δεν γίνεται να είναι κενά (NOT NULL), καθώς όλα είναι στοιχεία απαραίτητα για την ένταξη ενός σχολείου στην εφαρμογή μας.

```

63 CREATE TABLE IF NOT EXISTS books(
64     ISBN VARCHAR(13) NOT NULL,
65     title VARCHAR(50) NOT NULL,
66     publisher VARCHAR(50) NOT NULL,
67     num_of_pages int(4) NOT NULL,
68     summary TEXT,
69     cover_image BLOB NOT NULL,
70     language VARCHAR(15) NOT NULL,
71     keywords TEXT NOT NULL,
72     copies int(3),
73     school_name VARCHAR(50) NOT NULL,
74     PRIMARY KEY(ISBN,school_name),
75     FOREIGN KEY (school_name) REFERENCES school(school_name)
76 );

```

Στον πίνακα των βιβλίων, ορίσαμε ως primary key set το ISBN και το school\_name. Αυτό το κάναμε προκειμένου τα βιβλία να ξεχωρίζουν μεταξύ τους μέσω των στοιχείων αυτών και να μπορεί ένα βιβλίο μπορεί να ανήκει σε πάνω από ένα σχολείο. Η εικόνα του βιβλίου είναι τύπου BLOB, δηλαδή δυαδικής μορφής. Η περίληψη του βιβλίου είναι τύπου text και όχι varchar, καθώς σε κάποια βιβλία είναι πιθανό το μέγεθος της περίληψης να υπερβαίνει το μέγιστο μέγεθος του varchar.

Τα πεδία που ορίσαμε ως varchar πρόκειται για string μεταβλητού μεγέθους. Τα πεδία που ορίσαμε ως INT, πρόκειται για ακέραιους αριθμούς.

Για όλα τα παραπάνω πεδία, έχουμε ορίσει ότι δεν γίνεται να είναι κενά (NOT NULL), καθώς όλα είναι στοιχεία απαραίτητα για την ένταξη ενός βιβλίου στην εφαρμογή μας.

```

77
78 CREATE TABLE IF NOT EXISTS reviews(
79     ID int(9) NOT NULL,
80     ISBN VARCHAR(13) NOT NULL,
81     Likert int(5) NOT NULL,
82     free_text TEXT NOT NULL DEFAULT '',
83     approve_student_review BOOLEAN DEFAULT NULL,
84     PRIMARY KEY(ID,ISBN),
85     FOREIGN KEY(ID) REFERENCES student_teacher(ID),
86     FOREIGN KEY(ISBN) REFERENCES books(ISBN)
87 );
88

```

Στον πίνακα των αξιολογήσεων ορίσαμε ως primary key set το ISBN του βιβλίου και το ID του χρήστη που κάνει αξιολόγηση. Αυτό το κάναμε προκειμένου ένας χρήστης να μπορεί να αξιολογήσει πολλά διαφορετικά βιβλία. Η γραπτή αξιολόγηση που κάνει ο χρήστης (free\_text) είναι τύπου text και όχι varchar, καθώς σε κάποια βιβλία είναι πιθανό το μέγεθος της αξιολόγησης να υπερβαίνει το μέγιστο μέγεθος του varchar. Τα πεδία που ορίσαμε ως varchar πρόκειται για string μεταβλητού μεγέθους. Τα πεδία που ορίσαμε ως INT, πρόκειται για ακέραιους αριθμούς.



Τα πεδία που ορίσαμε ως BOOLEAN παίρνουν τιμές μόνο 1 και 0 και πρόκειται για εγκρίσεις.

Για όλα τα παραπάνω πεδία εκτός από τις εγκρίσεις, έχουμε ορίσει ότι δεν γίνεται να είναι κενά (NOT NULL), καθώς όλα είναι στοιχεία απαραίτητα για την ένταξη μιας αξιολόγησης στην εφαρμογή μας.

```
89 CREATE TABLE IF NOT EXISTS reservation(  
90     ID int(9) NOT NULL,  
91     ISBN VARCHAR(13) NOT NULL,  
92     date_of_reservation date,  
93     PRIMARY KEY(ID,ISBN),  
94     FOREIGN KEY(ID) REFERENCES student_teacher(ID),  
95     FOREIGN KEY(ISBN) REFERENCES books(ISBN)  
96 );
```

Στον πίνακα των κρατήσεων ορίσαμε ως primary key set το ISBN του βιβλίου και το ID του χρήστη που κάνει κράτηση. Αυτό το κάναμε προκειμένου ένας χρήστης να μπορεί να κάνει κράτηση πολλά διαφορετικά βιβλία.

Τα πεδία που ορίσαμε ως varchar πρόκειται για string μεταβλητού μεγέθους. Τα πεδία που ορίσαμε ως INT, πρόκειται για ακέραιους αριθμούς.

Το πεδίο που ορίσαμε με τύπο date πρόκειται για ημερομηνία.

Για όλα τα παραπάνω πεδία εκτός από την ημερομηνία, έχουμε ορίσει ότι δεν γίνεται να είναι κενά (NOT NULL), καθώς όλα είναι στοιχεία απαραίτητα για την ένταξη μιας κράτησης στην εφαρμογή μας.

```
98 CREATE TABLE IF NOT EXISTS borrowing(  
99     ID int(9) NOT NULL,  
100     ISBN VARCHAR(13) NOT NULL,  
101     date_of_borrow date,  
102     date_of_return date,  
103     situation enum('delayed','borrowed','returned','waiting') DEFAULT NULL,  
104     PRIMARY KEY(ID,ISBN),  
105     FOREIGN KEY(ID) REFERENCES student_teacher(ID),  
106     FOREIGN KEY(ISBN) REFERENCES books(ISBN)  
107 );
```

Στον πίνακα των δανεισμών ορίσαμε ως primary key set το ISBN του βιβλίου και το ID του χρήστη που κάνει δανεισμό. Αυτό το κάναμε προκειμένου ένας χρήστης να μπορεί να δανειστεί πολλά διαφορετικά βιβλία. Το situation το ορίσαμε enum, καθώς παίρνει προκαθορισμένες τιμές. Επίσης το ορίσαμε DEFAULT NULL καθώς όταν ένας χρήστης κάνει αίτηση για δανεισμό θέλουμε να μπαίνει στον πίνακα με situation NULL μέχρι ο χειριστής να εγκρίνει, να απορρίψει ή να θέσει σε αναμονή τον δανεισμό.

Τα πεδία που ορίσαμε ως varchar πρόκειται για string μεταβλητού μεγέθους. Τα πεδία που ορίσαμε ως INT, πρόκειται για ακέραιους αριθμούς.

Το πεδίο που ορίσαμε με τύπο date πρόκειται για ημερομηνία. Ομοίως και με το situation θέλουμε την ημερομηνία DEFAULT NULL.

Για το ID και το ISBN έχουμε ορίσει ότι δεν γίνεται να είναι NULL, καθώς είναι στοιχεία απαραίτητα για την ένταξη ενός δανεισμού στην εφαρμογή μας.

```
108
109 CREATE TABLE IF NOT EXISTS category(
110     theme VARCHAR(50) NOT NULL,
111     PRIMARY KEY(theme)
112 );
```

Στον πίνακα της κατηγορίας, έχουμε primary key το theme το οποίο είναι varchar, δηλαδή string μεταβλητού μεγέθους.

```
114 CREATE TABLE IF NOT EXISTS book_category(
115     theme VARCHAR(50) NOT NULL,
116     ISBN VARCHAR(13) NOT NULL,
117     PRIMARY KEY(theme, ISBN),
118     FOREIGN KEY(ISBN) REFERENCES books(ISBN),
119     FOREIGN KEY(theme) REFERENCES category(theme) ON UPDATE CASCADE
120 );
```

Στον πίνακα αυτό, συσχετίζουμε την κατηγορία με τα βιβλία. Έχουμε ορίσει ως primary key set το theme και το ISBN, αφού ένα βιβλίο μπορεί να ανήκει σε πάνω από μια κατηγορίες και ένα θέμα μπορεί να χαρακτηρίζει πάνω από ένα βιβλίο. Τα πεδία αυτά τα ορίσαμε ως varchar αφού πρόκειται για string μεταβλητού μεγέθους και επιπλέον δεν γίνεται να είναι NULL.

```
121
122 CREATE TABLE IF NOT EXISTS author(
123     author_id int(7) NOT NULL auto_increment,
124     author_name VARCHAR(50) NOT NULL,
125     PRIMARY KEY(author_id)
126 );
```

Στον πίνακα των συγγραφέων, έχουμε ορίσει ως primary key το author\_id, το οποίο χαρακτηρίζει τον κάθε συγγραφέα και για αυτό το ορίσαμε auto\_increment. Το όνομα του συγγραφέα είναι τύπου varchar καθώς πρόκειται για string μεταβλητού μεγέθους και δεν μπορεί να είναι NULL.

```

128 CREATE TABLE IF NOT EXISTS is_author(
129     ISBN VARCHAR(13) NOT NULL,
130     author_id int(7) NOT NULL AUTO_INCREMENT,
131     PRIMARY KEY(ISBN,author_id),
132     FOREIGN KEY(author_id) REFERENCES author(author_id),
133     FOREIGN KEY(ISBN) REFERENCES books(ISBN)
134 );
135

```

Στον πίνακα αυτό, συσχετίζουμε τους συγγραφείς με τα βιβλία που έχουν γράψει. Ορίσαμε ως primary key set το ISBN και το author\_id, καθώς ένας συγγραφέας μπορεί να έχει γράψει πάνω από ένα βιβλίο και ένα βιβλίο μπορεί να έχει πάνω από έναν συγγραφέα. Το ISBN είναι τύπου varchar καθώς πρόκειται για string μεταβλητού μεγέθους και δεν μπορεί να είναι NULL.

Τα επιπλέον indexes που χρησιμοποιήσαμε:

```

136 CREATE INDEX idx_situation ON borrowing(situation);
137
138 CREATE INDEX idx_role ON student_teacher(role);
139

```

Για τη δημιουργία της βάσης μας χρησιμοποιήσαμε MariaDB η οποία ορίζει ως indexes σε κάθε πίνακα τα primary keys, foreign keys και unique values.

Παρατηρήσαμε τα queries που υλοποιήσαμε και αποφασίσαμε να δημιουργήσουμε indexes στα attributes που χρησιμοποιούνται συχνότερα σε WHERE, JOIN, ORDER BY, GROUP BY.

Με την χρήση indexes γίνεται πιο γρήγορα η εύρεση των attributes που χρησιμοποιούνται συχνότερα και έτσι αυξάνεται η απόδοση των query.

- Στον πίνακα student\_teacher έχουμε ορίσει primary key το ID, το username ως unique value και foreign key το school\_name στον πίνακα των σχολείων. Η επιλογή του primary key έγινε καθώς θέλουμε να είναι μοναδικό στοιχείο για κάθε χρήστη της εφαρμογής μας, είναι σταθερό, να μην παίρνει NULL τιμές και να διευκολύνει τη σύνταξη και χρήση queries. Εκτός από τα παραπάνω indexes, προσθέσαμε ως index τον ρόλο στον πίνακα των μαθητών-καθηγητών καθώς χρειαζόταν για την διεκπεραίωση των queries. Πιο συγκεκριμένα, η επιλογή των παραπάνω indexes βοήθησε στην ταχύτερη εκτέλεση των queries 3.1.1, 3.1.2, 3.1.3, 3.2.2.
- Στον πίνακα books, έχουμε ορίσει primary key τα ISBN και school\_name. Η επιλογή του ISBN έγινε προκειμένου να ξεχωρίζει το ένα βιβλίο από το άλλο και η επιλογή του school\_name έγινε με σκοπό να μπορεί ένα βιβλίο να ανήκει σε πάνω από ένα σχολείο.

Πιο συγκεκριμένα, αυτά τα indexes βοήθησαν πολύ στην ταχύτερη εκτέλεση των queries: 3.2.3, 3.1.7, 3.1.3, 3.1.2 αλλά και στην γενικότερη υλοποίηση της εφαρμογής μας.

- Στον πίνακα school, όπως αναφέραμε παραπάνω, έχουμε επιλέξει ως primary key το school\_name και ως unique values τα email και phone\_number. Αυτά τα indexes βοήθησαν πολύ στην ταχύτερη εκτέλεση του query 3.1.1.
- Στον πίνακα borrowing, όπως αναφέραμε παραπάνω, έχουμε επιλέξει ως primary keys το ID του δανειζόμενου και το ISBN του βιβλίου. Αυτά τα indexes βοήθησαν πολύ στην ταχύτερη εκτέλεση των queries: 3.1.2, 3.1.3, 3.2.2. Ομοίως και στους πίνακες reservation και reviews.

Εκτός από τα παραπάνω indexes, προσθέσαμε ως index το situation του δανεισμού, γεγονός που βοήθησε στην ταχύτερη εκτέλεση των queries: 3.1.6, 3.2.2.

- Στον πίνακα manager, όπως αναφέραμε παραπάνω έχουμε primary key τα ID, first\_name, last\_name και unique value το username. Επιπλέον έχουμε foreign key το school\_name. Το index στο school\_name βοήθησε πολύ στην ταχύτερη υλοποίηση του query 3.2.2 αλλά και στην γενικότερη υλοποίηση της εφαρμογής μας, αφού σε πολλά σημεία του κώδικα υλοποιήσαμε query για να βρούμε το σχολείο στο οποίο ανήκει κάποιος manager.

### Τα triggers που χρησιμοποιήσαμε:

```
139
140 DELIMITER $$
141 CREATE OR REPLACE TRIGGER update_manager_approval_trigger
142 AFTER UPDATE ON admin
143 FOR EACH ROW
144 BEGIN
145     IF NEW.approve_manager = 1 THEN
146         UPDATE manager
147         SET approve_manager = 1
148         WHERE school_name = NEW.school_name;
149     END IF;
150 END $$
151 DELIMITER ;
```

Με το trigger αυτό, εξασφαλίζουμε ότι μόλις αλλάξει η στήλη approve\_manager στον admin και γίνει 1, τότε αυτόματα θα αλλάξει και η ίδια στήλη στον manager σε 1. Τότε, ο manager θα μπορεί να συνδεθεί στον λογαριασμό του.

### Τα views που χρησιμοποιήσαμε:

```
176
177 CREATE VIEW manager_borrowing AS
178     SELECT m.ID, m.first_name, m.last_name, b.ISBN, br.date_of_borrow
179     FROM manager m
180     INNER JOIN school s ON m.school_name = s.school_name
181     INNER JOIN student_teacher st ON st.school_name = s.school_name
182     INNER JOIN borrowing br ON st.ID = br.ID
183     INNER JOIN books b ON br.ISBN = b.ISBN
184     WHERE br.date_of_borrow IS NOT NULL AND b.school_name = m.school_name;
185
```

Με αυτό το view συσχετίζουμε τον πίνακα manager και τον πίνακα borrowing, για να πάρουμε ποιά βιβλία έχει δανείσει ένας manager. Το view αυτό το χρησιμοποιούμε στο query 3.1.5.

```
169
170 CREATE VIEW book_author(ISBN,title,publisher,num_of_pages,summary,cover_image,language,keywords,copies,school_name,author_id,author_name) AS
171     SELECT b.ISBN,b.title,b.publisher,b.num_of_pages,b.summary,b.cover_image,b.language,b.keywords,b.copies,b.school_name,a.author_id,a.author_name FROM books b
172     INNER JOIN is_author ia
173     ON b.ISBN = ia.ISBN
174     INNER JOIN author a
175     ON ia.author_id = a.author_id;
176
```

Με το view αυτό συνδέουμε τον πίνακα books με τους πίνακες is\_author και author, για να πάρουμε τον συγγραφέα του κάθε βιβλίου. Το χρησιμοποιήσαμε στο query 3.3.1. και 3.2.1.

```
185
186 CREATE VIEW school_borrows AS
187     SELECT s.school_name, br.ISBN ,br.date_of_borrow FROM school s
188     INNER JOIN student_teacher st ON st.school_name = s.school_name
189     INNER JOIN borrowing br ON st.ID = br.ID
190     WHERE br.date_of_borrow is NOT NULL ;
191
```

Με το view αυτό συσχετίζουμε τον πίνακα school με τον πίνακα borrowing για να πάρουμε το σχολείο στο οποίο βρίσκεται ένας μαθητής που έχει δανειστεί ένα συγκεκριμένο βιβλίο καθώς και την ημερομηνία δανεισμού. Το χρησιμοποιούμε στο query 3.3.1.

## Περιορισμοί Ορισμένοι από τον χρήστη

### **Συγγραφείς και Κατηγορίες**

Όταν ο manager εισάγει ένα καινούριο βιβλίο με περισσότερους από έναν συγγραφείς ή κατηγορίες, τους καταχωρεί με κόμμα χωρίς ενδιάμεσα κενά.

### **Εισαγωγή Manager**

Αν ένας χρήστης πάει να εγγραφεί ως manager σε σχολείο που ήδη έχει manager που έχει εγκριθεί, απορρίπτεται και δεν είναι δυνατή η εγγραφή του στο σύστημα.

### **Δανεισμοί**

Αν ένας χρήστης δεν μπορεί να δανειστεί ένα βιβλίο λόγω των δανεισμών του, τότε το αίτημά του απορρίπτεται και διαγράφεται από τον πίνακα.

Αν δεν μπορεί να δανειστεί λόγω των αντιτύπων, γίνεται update με situation = 'waiting'.

Όταν πραγματοποιείται δανεισμός, γίνεται update με situation = 'borrowed' και date\_of\_borrow = current\_date() ΚΑΙ update των αντιτύπων τα οποία μειώνονται κατά ένα και update των num\_of\_books τα οποία αυξάνονται κατά ένα.

### **Δανεισμός ίδιου βιβλίου δύο φορές**

Έχοντας ορίσει primary keys το ISBN και το ID του χρήστη στον πίνακα των δανεισμών, δεν γίνεται ένας χρήστης να δανειστεί το ίδιο βιβλίο δύο φορές ακόμα κι αν το έχει επιστρέψει.

### **Δανεισμός και κράτηση ίδιου βιβλίου**

Δεν επιτρέπεται ένας χρήστης να κάνει αίτηση για δανεισμό και αίτηση για κράτηση για το ίδιο βιβλίο.

### **Τι γίνεται όταν εγγράφεται ένας χρήστης;**

Γίνεται insert στον πίνακα με approve\_user\_registration = null. Ο manager βλέπει ποιοι χρήστες από τον πίνακα student\_teacher εκκρεμούν (approve\_user\_registration = null) και επιλέγει αν θα τους εγκρίνει (1) ή όχι (0).

Επιπλέον, όταν κάποιος εγκριθεί γίνεται το active = 1.

## **Κρατήσεις**

Άμα έχει εγκριθεί η κράτηση, τότε αυτή προστίθεται και στον πίνακα των δανεισμών με situation “waiting”. Μόλις εγκριθεί ο συγκεκριμένος δανεισμός, η κράτηση διαγράφεται από τον πίνακα των κρατήσεων

## **Διαγραφή Κράτησης μετά από 7 ημέρες**

Οι κρατήσεις διαγράφονται αν έχουν περάσει πάνω από 7 ημέρες από την ημερομηνία κράτησης. Ωστόσο, ο μάνατζερ την 8<sup>η</sup> ημέρα μόλις μπει στη σελίδα Approve Reservations, χρειάζεται να κάνει refresh για να βλέπει ποιες κρατήσεις είναι σε ισχύ. Αυτό συμβαίνει διότι πρώτα εμφανίζονται οι κρατήσεις και μετά διαγράφονται όσες έχουν διάρκεια πάνω από 7 ημέρες.

## **Εισαγωγή δεδομένων**

Για να μπορεί να εγγραφεί ένας μαθητής ή καθηγητής ή χειριστής και να ενταχθεί στη βάση μας, θα πρέπει πρώτα να υπάρχει στην βάση το σχολείο στο οποίο ανήκει.

## **Τι θα συμβεί στην περίπτωση που διαγράψουμε ή απενεργοποιήσουμε έναν μαθητή;**

Σε περίπτωση διαγραφής ή απενεργοποίησης ενός μαθητή ή καθηγητή, το boolean attribute του με όνομα “active” θα γίνεται false. Ο χρήστης και όλα τα δεδομένα του δεν θα διαγράφεται από τη βάση μας, απλώς δεν θα έχει τη δυνατότητα να κάνει οποιαδήποτε δραστηριότητα. Η επιλογή αυτή έγινε καθώς ακόμα και όταν ένας χρήστης διαγραφτεί, εμείς θέλουμε να συνυπολογίζουμε τους δανεισμούς και τις αξιολογήσεις του στα queries.

## **Εισαγωγή Βιβλίων**

Στο DML οι εισαγωγές των βιβλίων γίνονται με cover\_image =1 επειδή είναι τύπου δυαδικής μορφής (blob). Εάν κάποιος χειριστής θέλει να προσθέσει φωτογραφία σε κάποιο βιβλίο, έχει τη δυνατότητα να το κάνει μέσω της εφαρμογής. Πιο συγκεκριμένα, αναζητά το συγκεκριμένο βιβλίο και επιλέγει το κουμπί «Update Book». Στο πεδίο της

εικόνας, εισάγει την εικόνα που επιθυμεί, η οποία πρέπει να είναι τύπος «.jpeg»

## 2.2 DML script

```
Ε DML.txt
1  INSERT INTO school (school_name, school_address, email, city, principal_name, phone_number)
2  VALUES ('18o Gel Triandrias', 'Ethnikis Antistasews', '18o@hotmail.com', 'Thessaloniki', 'Petros Piliouris', 2319876543);
3
4  INSERT INTO school (school_name, school_address, email, city, principal_name, phone_number)
5  VALUES ('1o Gel Halandriou', 'Diogenous 2', '1ogel@gmail.com', 'Athens', 'Nikos Petridis', 2106196365);
6
7  INSERT INTO school (school_name, school_address, email, city, principal_name, phone_number)
8  VALUES ('2o Gel Thivas', 'Aulidos 2', '2ogel@gmail.com', 'Thiva', 'Koutsikos', 2262087615);
9
10 INSERT INTO school (school_name, school_address, email, city, principal_name, phone_number)
11 VALUES ('7o Gymnasio Haniwn', 'Nikhs 128', '7o@gmail.com', 'Crete', 'Apostolos Mavrovirogianakis', 2821457209);
12
13 INSERT INTO admin(ID,Username,password,school_name) VALUES(0,'administrator','000','18o Gel Triandrias');
14 INSERT INTO admin(ID,Username,password,school_name) VALUES(0,'administrator','000','1o Gel Halandriou');
15 INSERT INTO admin(ID,Username,password,school_name) VALUES(0,'administrator','000','2o Gel Thivas');
16 INSERT INTO admin(ID,Username,password,school_name) VALUES(0,'administrator','000','7o Gymnasio Haniwn');
17
18 INSERT INTO manager(ID, Username, Password, approve_manager, approve_user_registration, approve_student_review, approve_borrowing, approve_reservation, school_name, first_name, last_name)
19 VALUES (7, 'MarM', '123', 1, 1, NULL, NULL, NULL, '7o Gymnasio Haniwn', 'Marianthi', 'Metaxaki', '1980-09-26');
20
21 INSERT INTO manager(ID, Username, Password, approve_manager, approve_user_registration, approve_student_review, approve_borrowing, approve_reservation, school_name, first_name, last_name)
22 VALUES (8, 'dimitra', '456', 1, 1, NULL, NULL, NULL, '1o Gel Halandriou', 'Dimitra', 'Gini', '1983-06-27');
23
24 INSERT INTO manager(ID, Username, Password, approve_manager, approve_user_registration, approve_student_review, approve_borrowing, approve_reservation, school_name, first_name, last_name)
25 VALUES (9, 'Helen', '789', 1, 1, NULL, NULL, NULL, '2o Gel Thivas', 'Eleni', 'Sabani', '1988-03-27');
26
27 INSERT INTO manager (ID, Username, Password, approve_manager, approve_user_registration, approve_student_review, approve_borrowing, approve_reservation, school_name, first_name, last_name)
28 VALUES (11, 'petran', '567', 1, 1, NULL, NULL, NULL, '18o Gel Triandrias', 'Petros', 'Piliouris', '1997-03-01');
29
30 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
31 VALUES ('978006025665', 'Where the Wild Things Are', 'HarperCollins', 48, 'This children's picture book tells the imaginative story of Max, a mischievous boy who journeys to an island and tames various creatures.', '978006025665.jpg', 'English', 'Children's Literature', 10, '18o Gel Triandrias');
32
33 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
34 VALUES ('9780060288075', 'The Giver', 'HMH Books for Young Readers', 240, 'In a seemingly perfect society, Jonas discovers the dark secrets behind his utopian world, challenging the status quo.', '9780060288075.jpg', 'English', 'Young Adult Fiction', 15, '18o Gel Triandrias');
35
36 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
37 VALUES ('9780140494929', 'The Republic', 'Penguin Classics', 416, '"The Republic" is a philosophical work by Plato, exploring topics such as justice, governance, and the ideal society. It is one of Plato's most famous dialogues.', '9780140494929.jpg', 'English', 'Classics', 20, '18o Gel Triandrias');
38
39 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
40 VALUES ('978047089452', 'Mathematical Mindsets', 'Jossey-Bass', 288, '"Mathematical Mindsets" explores the concept of developing a growth mindset in mathematics education, providing strategies for teachers and students alike.', '978047089452.jpg', 'English', 'Education', 12, '18o Gel Triandrias');
41
42 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
43 VALUES ('978054757731', 'The Odyssey', 'Mariner Books', 550, 'An epic poem attributed to Homer, following the adventures of Odysseus as he attempts to return home after the Trojan War, facing numerous challenges and monsters.', '978054757731.jpg', 'English', 'Classics', 18, '18o Gel Triandrias');
```

```
45 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
46 VALUES ('978069115514', 'The Elements of Euclid', 'Princeton University Press', 856, '"Euclid's Elements" is a foundational mathematical work, providing a comprehensive introduction to geometry and number theory.', '978069115514.jpg', 'English', 'Mathematics', 25, '18o Gel Triandrias');
47
48 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
49 VALUES ('9780838026756', 'French: Discovering French, Nouveau!', 'This French language textbook is designed for students learning French as a second language. It covers vocabulary, grammar, and culture.', '9780838026756.jpg', 'French', 'Language Learning', 10, '18o Gel Triandrias');
50
51 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
52 VALUES ('9780060732592', 'Ariel: The Restored Edition', 'Harper Perennial Modern Classics', 224, 'Ariel is a collection of poems by Sylvia Plath. The collection, published posthumously, explores themes of identity, mental health, and the female experience.', '9780060732592.jpg', 'English', 'Poetry', 15, '18o Gel Triandrias');
53
54 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
55 VALUES ('9780060750503', 'The Love Poems of Rumi', 'HarperOne', 240, 'The Love Poems of Rumi is a collection of poems by the renowned Persian poet Rumi. These poems explore themes of love, spirituality, and the human condition.', '9780060750503.jpg', 'English', 'Poetry', 12, '18o Gel Triandrias');
56
57 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
58 VALUES ('9780060838652', 'A People's History of the United States', 'Harper Perennial', 729, 'A People's History of the United States presents a different perspective on American history, focusing on the experiences of ordinary people and social movements.', '9780060838652.jpg', 'English', 'History', 18, '18o Gel Triandrias');
59
60 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
61 VALUES ('9780061120080', 'The Alchemy of Air', 'Broadway Books', 336, 'The Alchemy of Air explores the fascinating story of the Haber-Bosch process, a scientific discovery that revolutionized agriculture and industry.', '9780061120080.jpg', 'English', 'Science', 10, '18o Gel Triandrias');
62
63 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
64 VALUES ('9780061122415', 'To Kill a Mockingbird', 'Harper Perennial', 336, 'Set in the 1930s, this Pulitzer Prize-winning novel explores racial injustice and the loss of innocence through the eyes of a young girl in Alabama.', '9780061122415.jpg', 'English', 'Fiction', 20, '18o Gel Triandrias');
65
66 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
67 VALUES ('9780061122415', 'To Kill a Mockingbird', 'Harper Perennial', 336, 'Set in the 1930s, this Pulitzer Prize-winning novel explores racial injustice and the loss of innocence through the eyes of a young girl in Alabama.', '9780061122415.jpg', 'English', 'Fiction', 20, '18o Gel Triandrias');
68
69 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
70 VALUES ('9780061122415', 'To Kill a Mockingbird', 'Harper Perennial', 336, 'Set in the 1930s, this Pulitzer Prize-winning novel explores racial injustice and the loss of innocence through the eyes of a young girl in Alabama.', '9780061122415.jpg', 'English', 'Fiction', 20, '18o Gel Triandrias');
71
72 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
73 VALUES ('9780061122415', 'To Kill a Mockingbird', 'Harper Perennial', 336, 'Set in the 1930s, this Pulitzer Prize-winning novel explores racial injustice and the loss of innocence through the eyes of a young girl in Alabama.', '9780061122415.jpg', 'English', 'Fiction', 20, '18o Gel Triandrias');
74
75 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
76 VALUES ('9780062316097', 'Sapiens: A Brief History of Humankind', 'Harper Perennial', 996, 'Sapiens offers a thought-provoking exploration of the history of humankind, from the emergence of modern humans to the present day.', '9780062316097.jpg', 'English', 'History', 15, '18o Gel Triandrias');
77
78 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
79 VALUES ('9780062509598', 'The Essential Rumi', 'HarperOne', 416, '"The Essential Rumi" is a collection of poems by the Persian poet Jalaluddin Rumi, translated by Coleman Barks. Rumi's poetry explores themes of love, spirituality, and the human condition.', '9780062509598.jpg', 'English', 'Poetry', 12, '18o Gel Triandrias');
80
81 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
82 VALUES ('9780071315012', 'Discrete Mathematics and Its Applications', 'McGraw-Hill Education', 972, 'This book explores various topics in discrete mathematics, including logic, sets, functions, and graph theory.', '9780071315012.jpg', 'English', 'Mathematics', 10, '18o Gel Triandrias');
83
84 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
85 VALUES ('9780070021756', 'Economics: Principles, Problems, and Policies', 'McGraw-Hill Education', 864, 'This economics textbook introduces fundamental economic concepts and principles, covering microeconomics and macroeconomics.', '9780070021756.jpg', 'English', 'Economics', 10, '18o Gel Triandrias');
```



```

86 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
87 VALUES ('9780078245836', 'Geography: The World and Its People', 'Glencoe/McGraw-Hill', 912, 'This biology textbook introduces students to the fundamental principles and concepts of biology. It covers topics such as the history of life, the structure and function of cells, and the interactions between organisms and their environment. This textbook is designed for use in high school biology courses and provides a comprehensive overview of the field.'),
88
89 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
90 VALUES ('9780078745607', 'Geography', 'Glencoe/McGraw-Hill', 728, 'This geography textbook offers a comprehensive exploration of the world's regions, physical landscapes, and human interactions. It covers topics such as the history of geography, the physical environment, and the impact of human activities on the environment. This textbook is designed for use in high school geography courses and provides a comprehensive overview of the field.'),
91
92 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
93 VALUES ('9780078746376', 'Chemistry: Matter and Change', 'McGraw-Hill Education', 996, 'This chemistry textbook provides a comprehensive introduction to the principles and applications of chemistry. It covers topics such as the history of chemistry, the structure and properties of matter, and the interactions between different chemical systems. This textbook is designed for use in high school chemistry courses and provides a comprehensive overview of the field.'),
94
95 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
96 VALUES ('9780078772375', 'Chemistry for Junior High School', 'McGraw-Hill Education', 400, 'Chemistry for Junior High School' is a comprehensive textbook that introduces basic concepts of chemistry, including the structure and properties of matter, chemical reactions, and the periodic table. It is designed for use in junior high school chemistry courses and provides a comprehensive overview of the field.'),
97
98 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
99 VALUES ('9780131481930', 'Introduction to Environmental Engineering and Science', 'Pearson', 729, 'This introductory textbook explores the principles and applications of environmental engineering and science. It covers topics such as the history of environmental engineering, the physical and chemical properties of the environment, and the impact of human activities on the environment. This textbook is designed for use in high school environmental engineering and science courses and provides a comprehensive overview of the field.'),
100
101 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
102 VALUES ('978013401127', 'Environmental Science: Toward a Sustainable Future', 'Pearson', 672, 'This environmental science textbook explores the interconnections between humans and the environment. It covers topics such as the history of environmental science, the physical and chemical properties of the environment, and the impact of human activities on the environment. This textbook is designed for use in high school environmental science courses and provides a comprehensive overview of the field.'),
103
104 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
105 VALUES ('9780134065455', 'Biology: Concepts and Connections', 'Pearson', 896, 'This biology textbook introduces students to the fundamental principles and concepts of biology. It covers topics such as the history of biology, the structure and function of cells, and the interactions between organisms and their environment. This textbook is designed for use in high school biology courses and provides a comprehensive overview of the field.'),
106
107 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
108 VALUES ('9780140275360', 'Iliad', 'Penguin Classics', 704, 'Iliad' is an ancient Greek epic poem attributed to Homer. It tells the story of the Trojan War, focusing on the events surrounding the death of Hector and the fall of Troy. This book is a classic of ancient Greek literature and is widely studied in high school and college courses.'),
109
110 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
111 VALUES ('9780140421996', 'Leaves of Grass', 'Penguin Classics', 704, 'Leaves of Grass' is a groundbreaking collection of poetry by Walt Whitman. Celebrated for its boldness and celebration of the human condition, it includes poems such as 'Song of Myself' and 'Drum-Drum'. This book is a classic of American literature and is widely studied in high school and college courses.'),
112
113 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
114 VALUES ('9780140441185', 'Thus Spoke Zarathustra', 'Penguin Classics', 352, 'Thus Spoke Zarathustra' is a philosophical novel by Friedrich Nietzsche. It presents a series of poetic and philosophical reflections on the human condition, the nature of power, and the concept of the Übermensch. This book is a classic of modern philosophy and is widely studied in high school and college courses.'),
115
116 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
117 VALUES ('9780140443161', 'Meditations', 'Penguin Classics', 304, 'Meditations' is a collection of personal writings by Marcus Aurelius, a Roman emperor and philosopher. It offers reflections on the nature of life, the human condition, and the pursuit of wisdom. This book is a classic of Stoic philosophy and is widely studied in high school and college courses.'),
118
119 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
120 VALUES ('978014044929', 'The Republic', 'Penguin Classics', 416, 'The Republic' is a foundational work of Western philosophy by Plato. It presents a dialogue between Socrates and other characters, exploring the nature of justice, the ideal state, and the role of the philosopher. This book is a classic of ancient Greek philosophy and is widely studied in high school and college courses.'),
121
122 INSERT INTO books (ISBN, title, publisher, num_of_pages, summary, cover_image, language, keywords, copies, school_name)
123 VALUES ('9780140449334', 'Meditations', 'Penguin Classics', 304, 'Meditations' is a collection of personal writings by Roman Emperor Marcus Aurelius. It offers reflections on Stoic philosophy, the human condition, and the pursuit of wisdom. This book is a classic of Stoic philosophy and is widely studied in high school and college courses.'),
124
125

```

Παραθέτουμε ενδεικτικά μερικά INSERT queries από το DML μας. Λόγω μεγέθους, το υπόλοιπο το έχουμε ανεβάσει στο git repo link που θα βρείτε στο τέλος της αναφοράς.

### 3.Ακολουθεί το User Manual της εφαρμογής μας:

# User Manual

**BiblioTECH**

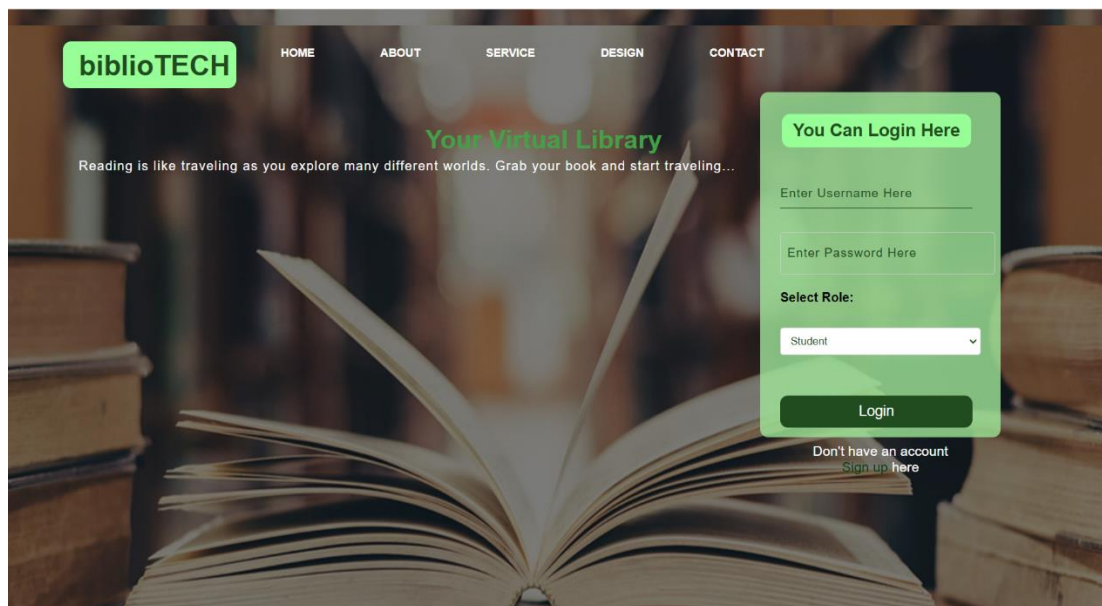
**Ομάδα 2**

**Δήμητρα Γκίνη 20128**

**Θάνου Παρασκευή 20225**

**Σαμπάνη Μαρία 20117**

## Οδηγίες της εφαρμογής για κάθε νέο χρήστη



### Σύνδεση

Κάθε χρήστης συνδέεται στην εφαρμογή με τον λογαριασμό του, συμπληρώνοντας το username και το password του και επιλέγοντας τον ρόλο του.

### Εγγραφή

Ο Admin της εφαρμογής μας είναι ένας και είναι εξ αρχής μέσα στην βάση μας και κανένας χρήστης δεν μπορεί να κάνει εγγραφή ως administrator.

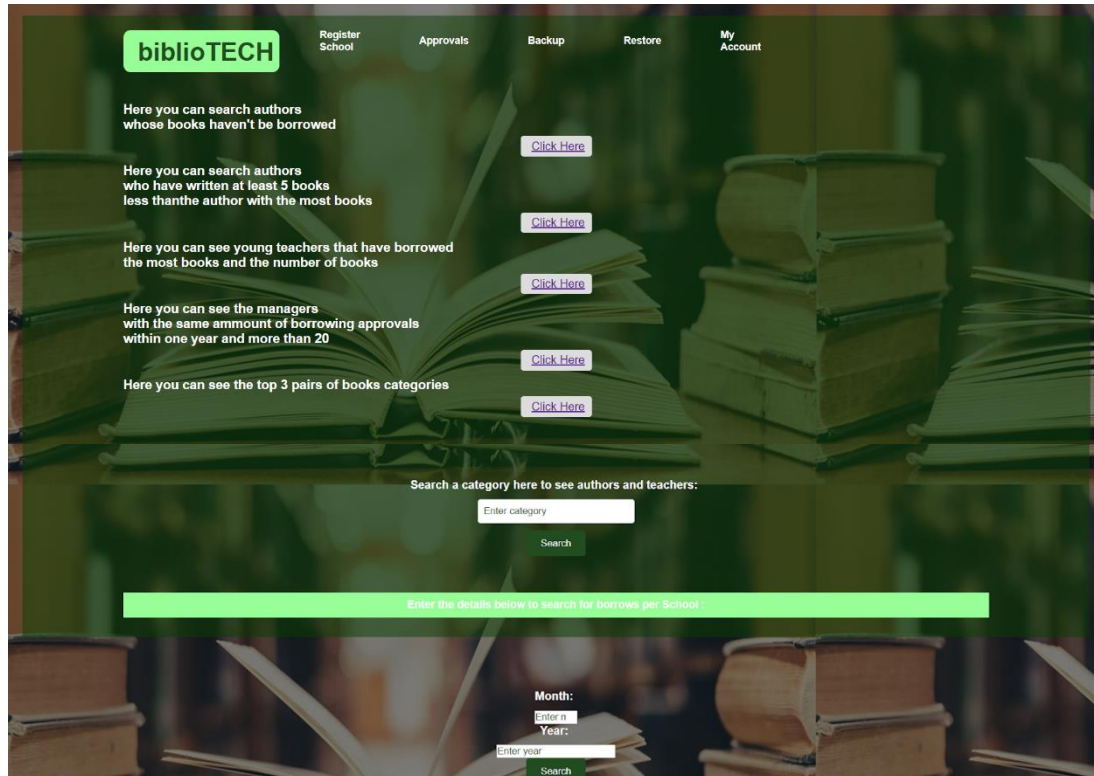
Για τη δημιουργία λογαριασμού θα πρέπει ο χρήστης να επιλέξει το «sign up» και να δημιουργήσει τον προσωπικό του λογαριασμό, συμπληρώνοντας τα στοιχεία του. Βασική προϋπόθεση για να γίνει αυτό είναι να έχει ήδη καταχωριστεί το σχολείο από τον admin.

Μόλις ένας χειριστής κάνει εγγραφή στο σύστημα, θα πρέπει για να μπορεί να συνδεθεί να εγκριθεί πρώτα από τον διαχειριστή.

Μόλις ένας μαθητής ή καθηγητής κάνει εγγραφή στο σύστημα, θα πρέπει να περιμένει την έγκριση της εγγραφής του από τον χειριστή του σχολείου του.

## Οδηγίες Χρήσης της Εφαρμογής για τον Admin

Όταν ο admin μπαίνει στον λογαριασμό του βλέπει τα εξής:



**Στο πάνω μέρος την οθόνης υπάρχει ένα μενού που αποτελείται από 5 κουμπιά:**

1. Από το κουμπί Register School μπορεί να εγγράψει ένα νέο σχολείο.
2. Από το κουμπί Approvals μπορεί να δει όλους τους χειριστές που έχουν κάνει sign up και περιμένουν να εγγραφούν. Έχει την επιλογή είτε να τους αποδεχτεί ( Approve) είτε να μην τους αποδεχτεί (Reject).
3. Από το κουμπί Backup μπορεί να δημιουργήσει αντίγραφο ασφαλείας της βάσης.
4. Από το κουμπί restore, μπορεί να επαναφέρει την εφαρμογή χρησιμοποιώντας το αντίγραφο ασφαλείας.
5. Από το κουμπί My Account, μπορεί να δει τα προσωπικά του στοιχεία τα οποία είναι το ID του, το Username του και τον κωδικό του.

**Κάτω από το μενού υπάρχουν 5 κάθετα κουμπιά:**

1. Πατώντας το πρώτο κουμπί μπορεί να δει όλους τους συγγραφείς των οποίων κανένα βιβλίο δεν έχει τύχει δανεισμού.
2. Πατώντας το δεύτερο κουμπί μπορεί να δει ποιοί συγγραφείς έχουν γράψει τουλάχιστον 5 βιβλία λιγότερα από τον συγγραφέα που έχει γράψει τα περισσότερα βιβλία
3. Πατώντας το τρίτο κουμπί μπορεί να δει τους νέους εκπαιδευτικούς ( κάτω των 40 ετών) που έχουν δανειστεί τα περισσότερα βιβλία, καθώς και τον αριθμό των βιβλίων.
4. Πατώντας το τέταρτο κουμπί, μπορεί να δει τους χειριστές που έχουν δανείσει τον ίδιο αριθμό βιβλίων μέσα σε διάστημα ενός έτους και με περισσότερους από 20 δανεισμούς.
5. Πατώντας το πέμπτο κουμπί μπορεί να δει τα 3 κορυφαία ζεύγη κατηγοριών που εμφανίστηκαν σε δανεισμούς.

#### **Παρακάτω υπάρχουν δύο φόρμες:**

1. Στην πρώτη φόρμα θα πρέπει υποχρεωτικά να εισάγει μια κατηγορία και θα μπορεί για αυτή την κατηγορία να δει ποιοι συγγραφείς ανήκουν σε αυτή και ποιοι εκπαιδευτικοί έχουν δανειστεί βιβλία αυτής της κατηγορίας το τελευταίο έτος.
2. Στην δεύτερη φόρμα μπορεί να δει τον συνολικό αριθμό δανεισμό ανά σχολείο με κριτήρια αναζήτησης έτος και ημερολογιακό μήνα.  
Αμα δεν εισάγει κανένα από τα δύο πεδία, τότε εμφανίζεται ο συνολικός αριθμός δανεισμών για όλα τα σχολεία, για όλους τους μήνες και όλα τα έτη.  
Αμα εισάγει μόνο τον μήνα τότε εμφανίζεται ο συνολικός αριθμός δανεισμών για όλα τα σχολεία, για όλα τα έτη αλλά για τον συγκεκριμένο μήνα.  
Αμα εισάγει μόνο το έτος τότε εμφανίζεται ο συνολικός αριθμός δανεισμών για όλα τα σχολεία, για όλους τους μήνες αλλά για το συγκεκριμένο έτος.



## Οδηγίες Χρήσης της Εφαρμογής για τους Χειριστές

Μόλις συνδεθεί στον λογαριασμό του θα πρέπει να βλέπει το εξής:

The screenshot displays the biblioTECH application interface with a dark green header. The header contains the logo 'biblioTECH' and seven navigation links: Users, User Approvals, Add Book, My Account, Borrowing Approvals/Return Registrations, Reservation Approvals, and Review Approvals. Below the header, there are four distinct search forms, each with a green title bar and a green search button. The first form is for searching books, with fields for Title, Author, Category, and Available Copies. The second form is for searching user ratings, with fields for Username and Category. The third form is for searching borrows, with a field for Username. The fourth form is for searching users with delayed books, with fields for First Name, Last Name, and Days Of Delay. The background of the interface is a blurred image of books.

**Search for books:**

Enter the details below to search for books :

Title: Enter title

Author: Enter author

Category: Enter category

Available Copies: Enter copies

Search

**Search for average rating of user:**

Enter the details below to search for average rating of user :

Username: Enter Username

Category: Enter category

Search

**Search for borrows for every user:**

Enter the details below to see the borrows for every user :

Username: Enter Username

Search

**Search for users with delayed books:**

Enter the details below to search for users with delayed books :

First Name: Enter first\_name

Last Name: Enter last\_name

Days Of Delay: Enter days

Search

**Στο πάνω μέρος την οθόνης υπάρχει ένα μενού που αποτελείται από 7 κουμπιά:**

1. Πατώντας το κουμπί Users, μπορεί να δει όλους τους ενεργούς χρήστες του σχολείου του (μαθητές και καθηγητές) που είναι εγγεγραμμένοι στην εφαρμογή.

School: 1o gel							
User ID	First Name	Last Name	Username	School	Role	Delete User	Print Card
2	Maria	Sabani	Maria	1o gel	student	<input type="button" value="Delete"/>	<input type="button" value="See Card"/>
3	Dimitra	Gkini	Dimitra	1o gel	student	<input type="button" value="Delete"/>	<input type="button" value="See Card"/>
5	Maria	Gkini	Marianthi	1o gel	teacher	<input type="button" value="Delete"/>	<input type="button" value="See Card"/>

- Έχει την επιλογή να διαγράψει κάποιον χρήστη (Delete) και να εκτυπώσει την κάρτα δανεισμού του (See Card)
2. Πατώντας το κουμπί User Approvals, μπορεί να δει τα στοιχεία των χρηστών που έχουν κάνει αίτηση εγγραφής. Έχει την επιλογή είτε να αποδεχτεί είτε να απορρίψει κάποιον χρήστη.
  3. Πατώντας το κουμπί Add Book μπορεί να προσθέσει ένα νέο βιβλίο στο σχολείο του.  
Αν θέλει να προσθέσει αντίτυπα σε ένα βιβλίο που ήδη υπάρχει, πρέπει να εισάγει πάλι τα στοιχεία του βιβλίου, βάζοντας στο πεδίο copies πόσα ακόμα αντίτυπα επιθυμεί να προσθέσει.
  4. Πατώντας το κουμπί My Account, μπορεί να δει τα προσωπικά του δεδομένα και να αλλάξει τον κωδικό του.
  5. Πατώντας το κουμπί Borrowing Approvals / Return Registrations, μπορεί να δει όλες τις αιτήσεις δανεισμού που έχουν κάνει οι χρήστες του σχολείου του. Για κάθε δανεισμό μπορεί να κάνει approve, reject ή waiting.  
Approve στην περίπτωση που ο χρήστης πληροί τα κριτήρια για να δανειστεί κάποιο βιβλίο, υπάρχουν διαθέσιμα αντίτυπα του συγκεκριμένου βιβλίου και δεν έχει καθυστερήσει την επιστροφή κάποιου βιβλίου.  
Σε περίπτωση που ο χρήστης πληροί τις παραπάνω προϋποθέσεις αλλά δεν υπάρχουν διαθέσιμα αντίτυπα θα πρέπει να επιλέξει την επιλογή waiting.  
Διαφορετικά ο δανεισμός δεν μπορεί να πραγματοποιηθεί και πρέπει να απορρίψει την αίτηση πατώντας την επιλογή

«reject».

Στην ίδια σελίδα εμφανίζεται και η επιλογή «register return» ή «do not register return» σε περίπτωση που κάποιος χρήστης έχει κάνει αίτηση για επιστροφή κάποιου βιβλίου.

6. Πατώντας το κουμπί Reservation Approvals εμφανίζονται όλες οι ενεργές κρατήσεις και τα αιτήματα για κράτηση. Μόλις ένα αίτημα για κράτηση εγκριθεί, εξαφανίζονται οι επιλογές approve, reject, waiting. Ο χειριστής πρέπει να επιλέξει την επιλογή approve για ένα αίτημα εφόσον ο μαθητής πληροί τις προϋποθέσεις κρατήσεις και δεν έχει στην κατοχή του καθυστερημένο βιβλίο.
7. Πατώντας το κουμπί Review Approvals μπορεί να δει τα στοιχεία του μαθητή, την αξιολόγησή του (ελεύθερο κείμενο και Likert) και το ISBN του βιβλίου που έχει αξιολογήσει. Ο χειριστής μπορεί να εγκρίνει (Approve) ή να απορρίψει (Reject ) μια αξιολόγηση.

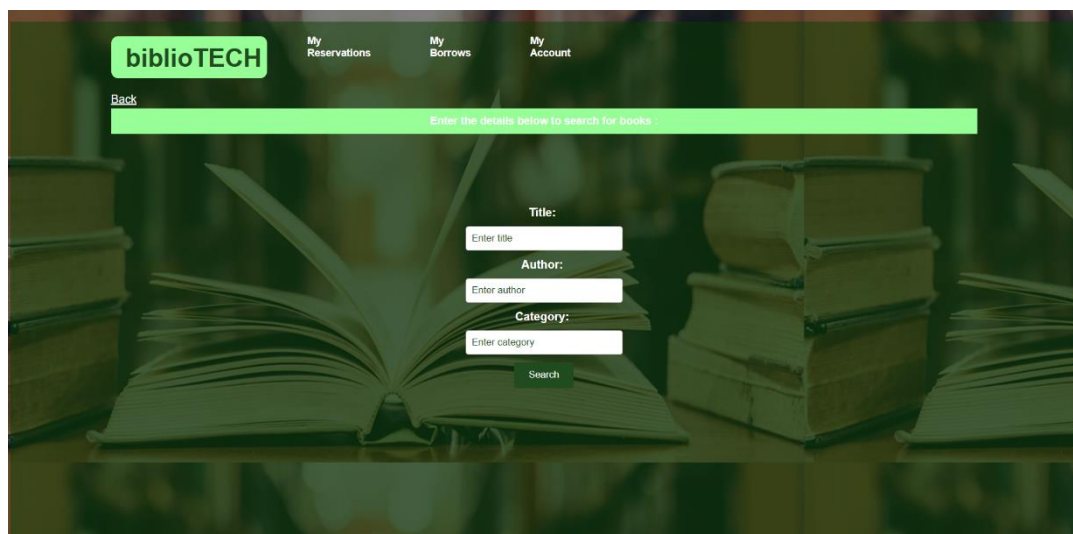
#### **Κάτω από το μενού υπάρχουν 4 φόρμες:**

1. Στην πρώτη φόρμα μπορεί να αναζητήσει τα βιβλία που υπάρχουν στο σχολείο του κατά τίτλο, συγγραφέα, κατηγορία και διαθέσιμα αντίτυπα.  
Αν δεν συμπληρώσει κανένα πεδίο της φόρμας τότε μπορεί να δει όλα τα βιβλία.  
Έχει επίσης την δυνατότητα να ανανεώσει τα στοιχεία κάποιο βιβλίου πατώντας πάνω στο Update Book.
2. Στην δεύτερη φόρμα, μπορεί να δει τον μέσο όρο αξιολογήσεων ανά δανειζόμενο και κατηγορία.  
Αν δεν συμπληρώσει κανένα πεδίο της φόρμας, τότε θα μπορεί να δει για όλους τους χρήστες του σχολείου του, τον μέσο όρο αξιολογήσεων ανά κατηγορία.
3. Στην τρίτη φόρμα, μπορεί να δει τους όλους δανεισμούς που έχει πραγματοποιήσει ένας χρήστης εισάγοντας το username του.  
Αν δεν συμπληρώσει το username, τότε μπορεί να δει τους δανεισμούς όλων των χρηστών.



4. Στην τέταρτη φόρμα, μπορεί να αναζητήσει κατά όνομα, επίθετο και ημέρες καθυστέρησης τους χρήστες που έχουν στην κατοχή τους ένα βιβλίο και έχουν καθυστερήσει την επιστροφή του.  
Αν δεν συμπληρώσει κανένα πεδίο της φόρμας, μπορεί να δει όλους όσους έχουν καθυστερήσει να επιστρέψουν κάποιο βιβλίο.

## Οδηγίες Χρήσης της Εφαρμογής για τους Καθηγητές



### Προβολή Στοιχείων

Πατώντας την επιλογή «My Account» ο καθηγητής μπορεί να δει τα προσωπικά του δεδομένα, δηλαδή το username, password, όνομα σχολείου στο οποίο ανήκει, ημερομηνία γέννησης και ονοματεπώνυμο. Επιπλέον, στη σελίδα αυτή έχει την επιλογή να αλλάξει τον κωδικό του και τα προσωπικά του δεδομένα.

### Αναζήτηση Βιβλίων

Κάθε καθηγητής μόλις συνδεθεί στον λογαριασμό του μπορεί να ψάξει τα βιβλία που υπάρχουν στο σχολείο του. Η αναζήτηση αυτή μπορεί να γίνει με βάση τον τίτλο του βιβλίου, τον συγγραφέα ή την κατηγορία του βιβλίου. Αν ωστόσο, δεν συμπληρώσει κανένα από τα πεδία αυτά και πατήσει το κουμπί «search» τότε του εμφανίζονται όλα τα βιβλία του σχολείου στο οποίο είναι εγγεγραμμένος.

### Δραστηριότητες Κράτησης/Δανεισμού/Αξιολόγησης Βιβλίων

Για κάθε βιβλίο έχει την επιλογή να κάνει αίτηση δανεισμού ή αίτηση κράτησης. Να σημειωθεί πως δεν επιτρέπεται ένας χρήστης να κάνει αίτηση για κράτηση για ένα βιβλίο που έχει ήδη δανειστεί ή για ένα βιβλίο που έχει ήδη κάνει αίτηση για δανεισμό. Αντίστοιχα, δεν επιτρέπεται να κάνει αίτηση για δανεισμό σε ένα βιβλίο που έχει ήδη κάνει αίτηση για κράτηση. Παράλληλα, έχει την επιλογή να γράψει την άποψη του για ένα βιβλίο με τη μορφή κειμένου ή να το αξιολογήσει σε κλίμακα Likert.

### **Δανεισμοί**

Αφού υποβάλλει την αίτηση του, θα πρέπει να περιμένει για την ανανέωση της κατάστασης του δανεισμού από τον χειριστή του σχολείου.

Για να δει την κατάσταση των δανεισμών ή των κρατήσεων του, χρειάζεται να πάει στην σελίδα «My Borrowows». Εκεί βλέπει όλους τους δανεισμούς του και την κατάσταση τους ('borrowed', 'waiting', 'returned' ή 'delayed').

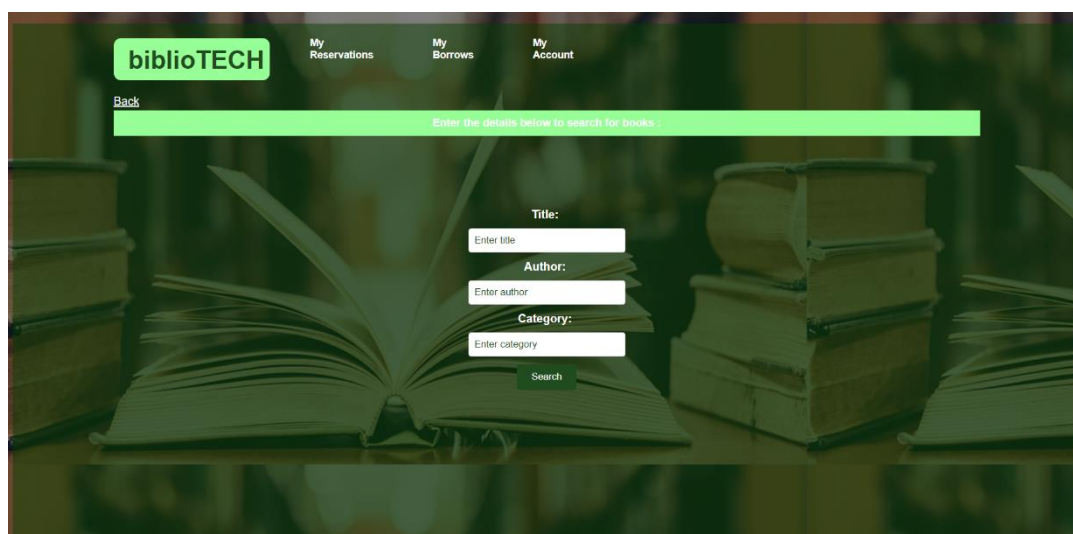
### **Επιστροφή Βιβλίου**

Σε περίπτωση που θέλει να επιστρέψει ένα βιβλίο που έχει ήδη δανειστεί, χρειάζεται να πάει στη σελίδα «My Borrowows» και να επιλέξει το κουμπί «Return». Αν το κουμπί δεν εμφανίζεται σημαίνει ότι είτε το βιβλίο έχει ήδη επιστραφεί και άρα έχει αλλάξει η κατάστασή του, είτε ότι έχει ήδη επιλέξει την επιλογή αυτή και πρέπει να περιμένει την καταχώρηση της επιστροφής από τον χειριστή. Η κατάσταση του δανεισμού αλλάζει σε «returned» ή «delayed» μόλις ο χειριστής καταχωρήσει την επιστροφή.

### **Κρατήσεις**

Για να δει ο καθηγητής τα αιτήματα κράτησης ή τις ενεργές κρατήσεις του θα πρέπει να επιλέξει το «My Reservations». Εκεί φαίνονται τα παραπάνω και έχει την δυνατότητα να διαγράψει κάποια κράτηση επιλέγοντας το κουμπί «Delete». Να σημειωθεί ότι οι κρατήσεις έχουν διάρκεια 7 ημέρες και μετά το πέρας των ημερών αυτών, διαγράφονται αυτόματα.

## Οδηγίες Χρήσης της Εφαρμογής για τους Μαθητές



### Προβολή Στοιχείων

Πατώντας την επιλογή «My Account» ο καθηγητής μπορεί να δει τα προσωπικά του δεδομένα, δηλαδή το username, password, όνομα σχολείου στο οποίο ανήκει, ημερομηνία γέννησης και ονοματεπώνυμο. Επιπλέον, στη σελίδα αυτή έχει την επιλογή να αλλάξει τον κωδικό του.

### Αναζήτηση Βιβλίων

Κάθε μαθητής μόλις συνδεθεί στον λογαριασμό του μπορεί να ψάξει τα βιβλία που υπάρχουν στο σχολείο του. Η αναζήτηση αυτή μπορεί να γίνει με βάση τον τίτλο του βιβλίου, τον συγγραφέα ή την κατηγορία του βιβλίου. Αν ωστόσο, δεν συμπληρώσει κανένα από τα πεδία αυτά και πατήσει το κουμπί «search» τότε του εμφανίζονται όλα τα βιβλία του σχολείου στο οποίο είναι εγγεγραμμένος.

### Δραστηριότητες Κράτησης/Δανεισμού/Αξιολόγησης Βιβλίων

Για κάθε βιβλίο έχει την επιλογή να κάνει αίτηση δανεισμού ή αίτηση κράτησης. Να σημειωθεί πως δεν επιτρέπεται ένας χρήστης να κάνει αίτηση για κράτηση για ένα βιβλίο που έχει ήδη δανειστεί ή για ένα βιβλίο που έχει ήδη κάνει αίτηση για δανεισμό. Αντίστοιχα, δεν επιτρέπεται να κάνει αίτηση για δανεισμό σε ένα βιβλίο που έχει ήδη κάνει αίτηση για κράτηση. Παράλληλα, έχει την επιλογή να γράψει την άποψη του για ένα βιβλίο με τη μορφή κειμένου ή να το αξιολογήσει σε κλίμακα Likert. Η αξιολόγηση του θα πρέπει να εγκριθεί από τον χειριστή.

### **Δανεισμοί**

Αφού υποβάλλει την αίτηση του, θα πρέπει να περιμένει για την ανανέωση της κατάστασης του δανεισμού από τον χειριστή του σχολείου.

Για να δει την κατάσταση των δανεισμών ή των κρατήσεων του, χρειάζεται να πάει στην σελίδα «My Borrowers». Εκεί βλέπει όλους τους δανεισμούς του και την κατάσταση τους ('borrowed', 'waiting', 'returned' ή 'delayed').

### **Επιστροφή Βιβλίου**

Σε περίπτωση που θέλει να επιστρέψει ένα βιβλίο που έχει ήδη δανειστεί, χρειάζεται να πάει στη σελίδα «My Borrowers» και να επιλέξει το κουμπί «Return». Αν το κουμπί δεν εμφανίζεται σημαίνει ότι είτε το βιβλίο έχει ήδη επιστραφεί και άρα έχει αλλάξει η κατάστασή του, είτε ότι έχει ήδη επιλέξει την επιλογή αυτή και πρέπει να περιμένει την καταχώρηση της επιστροφής από τον χειριστή. Η κατάσταση του δανεισμού αλλάζει σε «returned» ή «delayed» μόλις ο χειριστής καταχωρήσει την επιστροφή.

### **Κρατήσεις**

Για να δει ο καθηγητής τα αιτήματα κράτησης ή τις ενεργές κρατήσεις του θα πρέπει να επιλέξει το «My Reservations». Εκεί φαίνονται τα παραπάνω και έχει την δυνατότητα να διαγράψει κάποια κράτηση επιλέγοντας το κουμπί «Delete». Να σημειωθεί ότι οι κρατήσεις έχουν διάρκεια 7 ημέρες και μετά το πέρας των ημερών αυτών, διαγράφονται αυτόματα.

#### 4. Οδηγίες Εγκατάστασης της εφαρμογής μας

Υλοποιήσαμε την εφαρμογή μας χρησιμοποιώντας Python και VSCODE.

Κατεβάζουμε το xampp και κάνουμε start το Apache και το Mysql.

Αρχικά κατεβάζουμε την έκδοση 3.11.3 της Python.

Για την ανάπτυξη της εφαρμογής μας χρησιμοποιούμε **το Flask της Python** το οποίο είναι ένα web framework που διευκολύνει την ανάπτυξη εφαρμογών. Το Flask βασίζεται στο εργαλείο Werkzeug και στο Jinja2 template engine. Μέσω του Jinja2 μπορούμε να περάσουμε μεταβλητές της Python σε templates της HTML και μέσω του Werkzeug, υλοποιείται η σύνδεση μεταξύ του web server και της εφαρμογής μας.

Για την εγκατάσταση του Flask ακολουθήσαμε τα εξής βήματα:

- Στο command prompt γράφουμε τις εξής εντολές:
  - 1.python -m venv myenv
  - 2.myenv\Scripts\activate
  - 3.pip install Flask;
- Αφου ολοκληρωθεί η εγκατάσταση ανοίγουμε το VSCODE και εκτελούμε τις παραπάνω εντολές.  
Σε περίπτωση που δεν είναι δυνατή η εγκατάσταση λόγω ελλিপών δικαιωμάτων ( πρέπει να έχουμε δικαιώματα administrator) εκτελούμε τα εξής:
  - 1.Get-ExecutionPolicy (για να δω τι είμαι)
  - 2.Αν είμαι Restricted ή AllSigned γράφω στο terminal  
  
Set-ExecutionPolicy -Scope CurrentUser -  
ExecutionPolicy RemoteSigned
  - 3.Τώρα μπορώ να γράψω  
.\myenv\Scripts\Activate  
και μετά  
pip install Flask
- Τέλος κατεβάζουμε και τις υπόλοιπες βιβλιοθήκες που χρειαζόμαστε, γράφοντας στο terminal τις εξής εντολές:

- 1.pip install flask-mysqldb
- 2.pip install -U Flask-WTF

Η βιβλιοθήκη flask-mysqldb είναι απαραίτητη για την σύνδεση της εφαρμογής μας με την βάση μας.

Η σύνδεση αυτή πραγματοποιείται στο αρχείο `__init__.py`.

Αφού ολοκληρωθεί η σύνδεση με επιτυχία, μπορούμε να υλοποιήσουμε query ως εξής:

```
cur = db.connection.cursor()    ;έναρξη σύνδεσης
cur.execute(query,values)       ;εκτέλεση query
db.connection.commit()          ;ανενέωση της βάσης ανάλογα με το
query
cur.close()                     ;τερματισμός της σύνδεσης
```

Η δομή της εργασίας μας

Όλο το project βρίσκεται σε έναν φάκελο, τον οποίο εμείς καλούμε `library_web`. Μέσα σε αυτόν τον φάκελο έχουμε έναν φάκελο που λέγεται `dbdemo` καθώς και τρία ξεχωριστά αρχεία: το `run.py` , το `DDL.sql` και το `DML.txt`.

Ο φάκελος `dbdemo` περιέχει έναν φάκελο `static`, έναν φάκελο `templates` και ένα αρχείο `__init__.py`.

- Όπως αναφέρθηκε και παραπάνω, στο αρχείο `__init__.py` υλοποιείται η δημιουργία της εφαρμογής και η σύνδεση με την βάση.
- Ο φάκελος `static` περιέχει ένα αρχείο `style.css` στο οποίο βρίσκεται όλη η μορφοποίηση των `html` αρχείων μας, καθώς επίσης και μια φωτογραφία `library.png` η οποία είναι το `background` της εφαρμογής μας.
- Ο φάκελος `templates` περιέχει όλα τα `html templates` μας, δηλαδή όλες τις σελίδες της εφαρμογής μας.

Το `run.py`, περιέχει όλα τα `routes` της εφαρμογής μας.

Το `DML.txt` περιέχει όλα τα δεδομένα της βάσης μας σε μορφή κώδικα `sql` ( `INSERT INTO table_name(attribute1, attribute2, attribute3) VALUES(value1, value2, value3)` ). Τα `values` συμβαδίζουν με τον τύπο των `attributes`.

Το DDL.sql περιέχει σε sql τον κώδικα δημιουργίας της βάσης μας, δηλαδή όλους του πίνακες, τα views, τα triggers και τα indexes που έχουμε χρησιμοποιήσει.

Για να δημιουργηθεί η βάση μας και να μπορεί να λειτουργήσει σωστά η εφαρμογή, αρκεί να τρέξουμε το DDL.sql. Για να γίνει αυτό, ακολουθούμε τα εξής βήματα:

1.Ανοίγουμε το command prompt και γράφουμε:

cd path (το path στο οποίο βρίσκεται το DDL.sql)

2."C:\xampp\mysql\bin\mysql" -u root -p vash < DDL.sql

(vash: είναι το όνομα της βάσης μας)

Αφού έχουμε πάρει όλους τους κώδικες και έχουμε δημιουργήσει την βάση, τρέχουμε το αρχείο run.py στο vascoder. Μόλις βεβαιωθούμε ότι τρέχει ( δεν έχει βγάλει μηνύματα λάθους) , ακολουθούμε τον σύνδεσμο

<http://localhost:3000>

Εναλλακτικά, μπορούμε να τρέξουμε την εφαρμογή και από το command prompt γράφοντας τις εξής εντολές:

α)cd C:\Users\mpik\Desktop\library\_web (το path στο οποίο βρίσκεται το run.py)

β)python run.py

Σχηματικά, η δομή της εργασίας μας είναι η ακόλουθη:

Library\_web

- Dbdemo
  - Static(folder)
  - Templates(folder)
  - \_\_init\_\_.py
- run.py
- DDL.sql
- DML.txt

## 5. Σύνδεσμος git repo της εφαρμογής μας

<https://github.com/DimitraGkini123/DATABASES-ASSIGNMENT-OMADA-2.git>