

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ  
Ακαδημαϊκό έτος 2022-2023

Ομάδα Project 29

Παρασκευή Γκούλτα | AM : 03120864

Κωνσταντίνος Σταθάκος| AM : 03120079

Σταύρος Μητρόπουλος| AM : 03120847

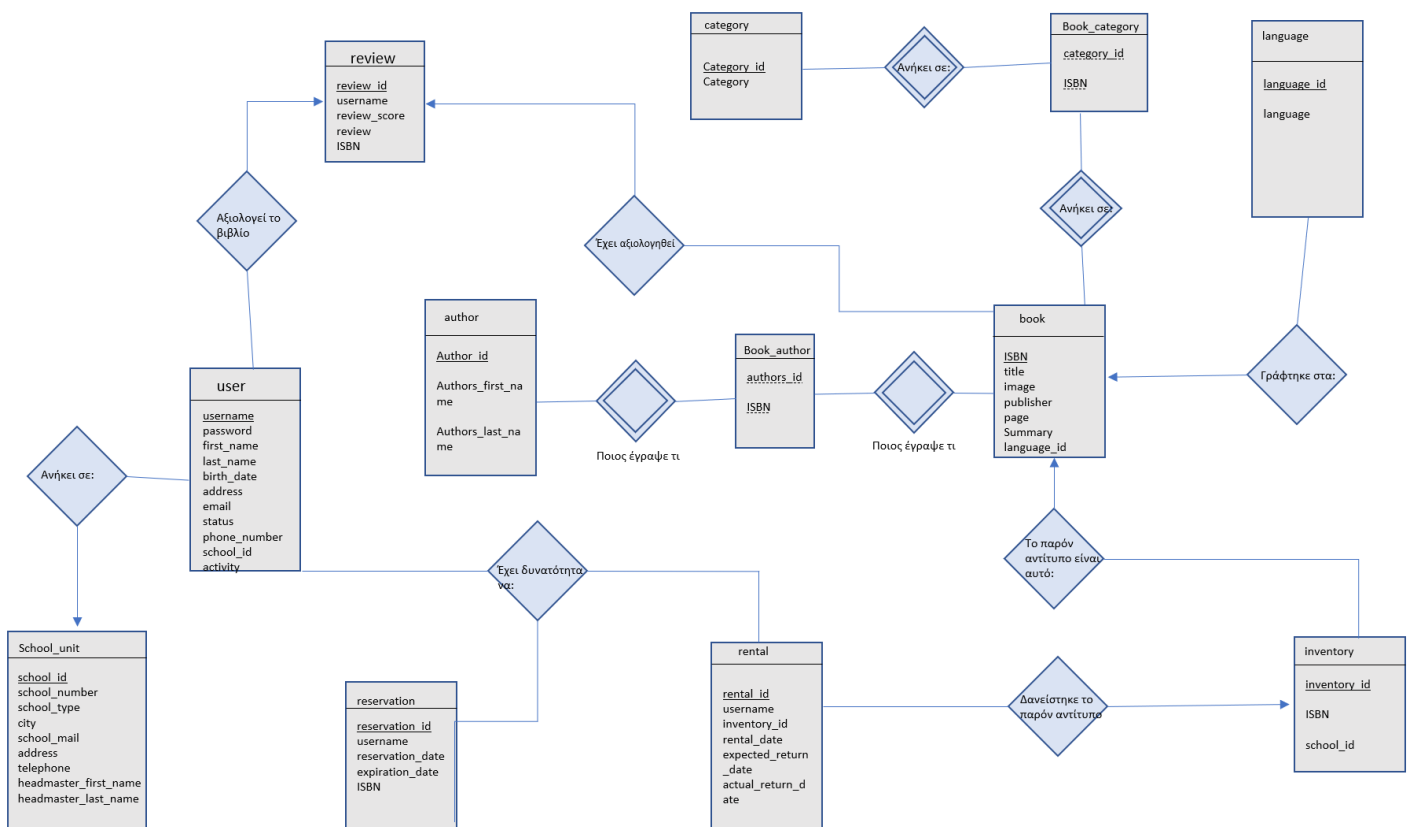
## Εισαγωγή:

Η παρούσα εργασία υλοποιεί ένα σύστημα το οποίο δίνει πρόσβαση σε κάθε δημόσιο σχολείο για την κάλυψη των πληροφοριακών αναγκών του σε θέματα λειτουργίας της σχολικής βιβλιοθήκης. Αποτελεί ένα σύστημα αποθήκευσης και διαχείρισης των πληροφοριών που απαιτούνται για την εύρυθμη λειτουργία βιβλιοθήκης στα δημόσια σχολεία.

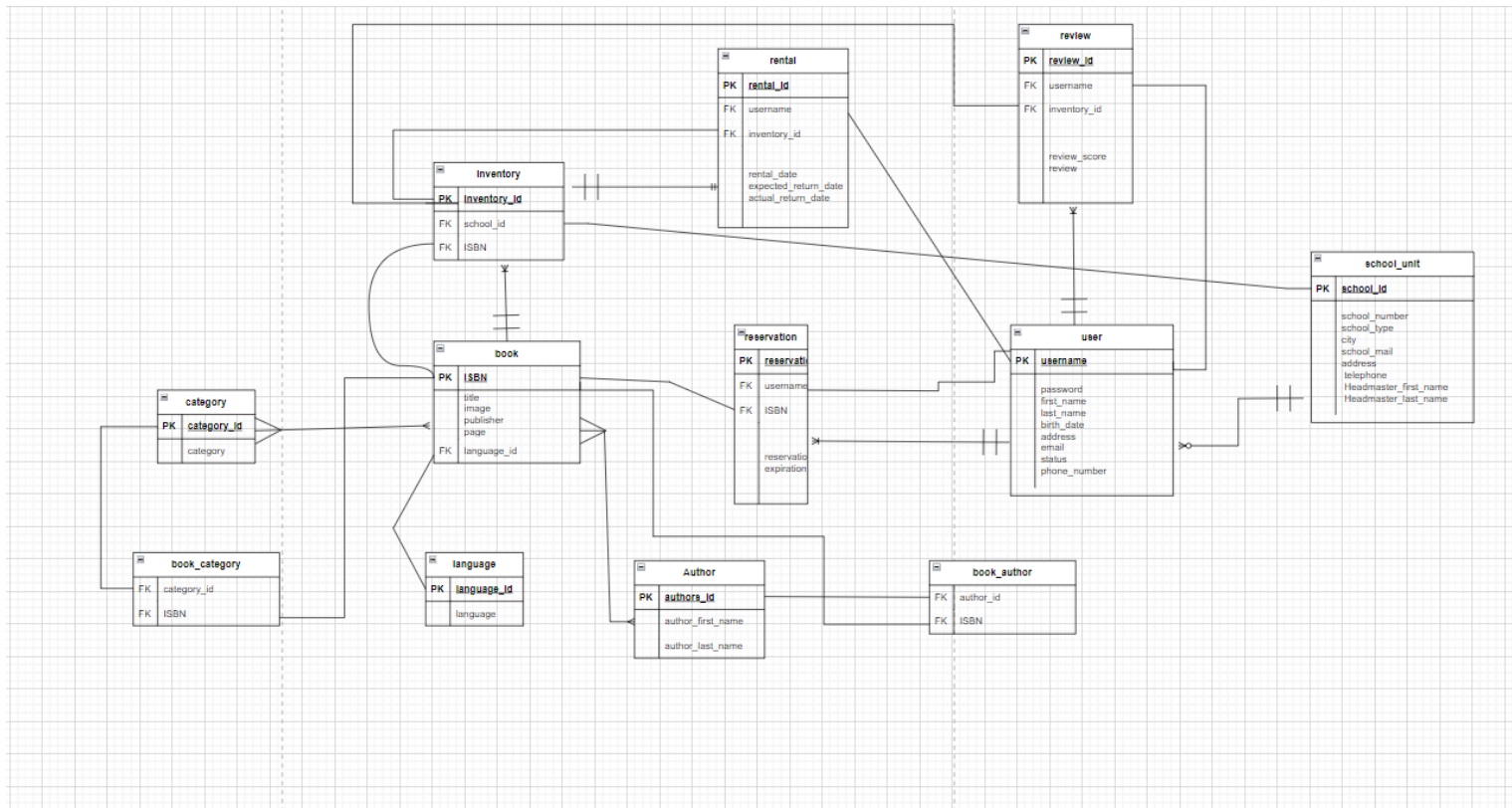
Για την υλοποίηση της, δουλέψαμε πάνω στην SQL η οποία είναι *standardized programming language* για σχεσιακές βάσεις δεδομένων καθώς επίσης δημιουργήσαμε γραφικό περιβάλλον ιστοσελίδας στην γλώσσα σεναρίου PHP το οποίο δίνει την δυνατότητα στον χρήστη να αλληλεπιδρά με την βιβλιοθήκη.

### 1.1)

Το ER Diagram της βάσης δεδομένων της βιβλιοθήκης:



Το σχεσιακό διάγραμμα που προέκυψε από το παραπάνω:



### Σχολιασμός διαγραμμάτων:

Το ER Diagram φαίνονται όλες οι βασικές σχέσεις μεταξύ των οντοτήτων καθώς επίσης (μέσω της θεωρίας του προτεινόμενου βιβλίου του μαθήματος) οι σχέσεις one-one, one-many, many-many μεταξύ τους. Επιπροσθέτως, φαίνονται και οι λογικές σχέσεις μεταξύ τους, δηλαδή σε φυσική γλώσσα το γιατί συνδέονται στο διάγραμμα (στους ρόμβους του σχήματος). Η λογική πίσω από αυτό το ER είναι κάθε entity να έχει “επικοινωνία” με κάθε άλλο entity το οποίο μπορεί να επηρεαστεί από τις βασικές ιδιότητες του πρώτου, π.χ. βλέπουμε πως ο User ενώνεται με τα entities στα οποία μπορεί να “δράσει”.

Το Relational Diagram φαίνονται πιο αναλυτικά οι σχέσεις μεταξύ των οντοτήτων, τα primary και foreign keys. Επίσης υπάρχει ένωση μεταξύ των Foreign keys με τα attributes από τα οποία παίρνουν την ιδιότητα τους. Η λογική του, προφανώς, βασίστηκε πάνω σε εκείνη του ER Diagram.

Σε σχέση με τα ευρετήρια (indexes) που χρησιμοποιήθηκαν, επιλέξαμε να χρησιμοποιηθεί ένα index ανά ερώτημα ώστε να αποφύγουμε το over-indexing το οποίο φορτώνει την μνήμη και εν τέλει κάνει τη βάση πιο αργή από το να μην υπήρχαν indexes. Λειτουργήσαμε έχοντας στο νου μας ότι συμβαίνει αυτόματο Indexing των FK στη MySQL.

(<https://dev.mysql.com/doc/refman/8.0/en/constraint-foreign-key.html>).

Για κάθε ερώτημα (που κρίναμε ότι ένα index θα ήταν χρήσιμο) με την σειρά:

- `CREATE INDEX rental_rental_date_idx ON rental (rental_date);` για το 3.1.1. λόγω της διπλής κλήσης του `rental_date` στο `where` του αντίστοιχου query. Επίσης το `rental_date` χρησιμοποιείται και σε άλλα queries.
- Στο 3.1.2. επιλέξαμε το εξής: `CREATE INDEX idx_category ON category (category);` λόγω της διπλής του παρουσίας στο ερώτημα.
- Στο 3.1.3. επιλέξαμε το εξής `CREATE INDEX idx_user_status_username ON user (status, username);` με το ίδιο σκεπτικό, ότι δηλαδή αυτά είναι τα Attributes τα οποία χρησιμοποιούνται στο query περισσότερες φορές (δύο το καθένα).
- Στο 3.1.4 δεν έχουμε κάποια επανάληψη κάποιου attribute στο query οπότε επιλέγουμε το `CREATE INDEX idx_inv_id ON inventory (inventory_id);` διότι τα φυσικά αντίτυπα μιας βιβλιοθήκης είναι ο μεγαλύτερος αριθμός που μπορείς να βρεις σε ένα τέτοιο σύστημα.
- Στο 3.1.5 δεν θα προσθέσουμε index διότι έχουμε ήδη φτιάξει για τα `rental_date` & `username`.
- Στο 3.1.7 επιλέγουμε να κάνουμε indexing το `author_id` καθώς χρησιμοποιείται παραπάνω από μία φορές. `CREATE INDEX idx_author_id ON author (authors_id);`.
- Για το 3.2.2 επιλέγουμε το `CREATE INDEX idx_rental_expected_return_date ON rental (expected_return_date);` λόγω της συχνής χρήσης του attribute αυτού.
- Τέλος για το 3.3.1 δημιουργούμε ένα index `CREATE INDEX idx_author_name ON author (authors_first_name, authors_last_name);`

## 1.2)

Ακολουθούν τα DDL και DML scripts της βάσης. Ακολουθεί επεξήγηση των constraints και του Indexing της βάσης.

### DDL:

Το DDL βρίσκεται στο `create_schema.sql` στο git repo. Εμπεριέχει όλους τους πίνακες που δημιουργήθηκαν για την λειτουργία της βάσης. Ακολουθεί η επεξήγηση τους:

- Ο πίνακας **Author** κρατάει όλους του συγγραφείς των οποίων έργα κατέχει το σύνολο των βιβλιοθηκών του σχολικού δικτύου. Primary key αποτελεί το `author_id` ένας μοναδικός αριθμός που αντιστοιχεί σε κάθε συγγραφέα για πιο εύκολη αναζήτηση του ιδίου εντός της βάσης.
- Ο πίνακας **category** κρατάει όλες τις πιθανές κατηγορίες στις οποίες ένα βιβλίο θα μπορούσε να ανήκει. Primary key είναι το `category_id` το οποίο είναι

ένας μοναδικός αριθμός που αντιστοιχεί στην εκάστοτε κατηγορία ώστε να γίνεται πιο εύκολα η αναζήτηση των κατηγοριών εντός της βάσης. Ο πίνακας αυτός, επίσης έχει και ένα constraint το οποίο δεν επιτρέπει την ύπαρξη δύο ίδιων instances του entity Category.

- Ο πίνακας **language** κρατάει όλες τις πιθανές γλώσσες στις οποίες ένα βιβλίο θα μπορούσε να έχει γραφτεί. Primary key είναι το language\_id το οποίο είναι ένας μοναδικός αριθμός που αντιστοιχεί στην εκάστοτε γλώσσα ώστε να γίνεται πιο εύκολα η αναζήτηση εντός της βάσης. Ο πίνακας αυτός, επίσης έχει και ένα constraint το οποίο δεν επιτρέπει την ύπαρξη δύο ίδιων instances του entity language.
- Ο πίνακας **book** κρατάει όλους τους τίτλους βιβλίων που υπάρχουν στις βιβλιοθήκες του σχολικού βιβλίου, μαζί με χρήσιμες πληροφορίες για αυτά. Primary key το ISBN, το αριθμητικό αναγνωριστικό κάθε βιβλίου παγκοσμίως. Βασική πληροφορία για κάθε βιβλίο προφανώς είναι και η γλώσσα στην οποία είναι γραμμένο, οπότε η οντότητα αυτή έχει foreign key στο language\_id του entity Language ώστε να μπορούμε εύκολα να διευκρινίζουμε την γλώσσα του βιβλίου μέσα από μία μεγάλη γκάμα γλωσσών οι οποίες αποτελούν και κριτήριο αναζήτησης σε κάποια από τα queries που ζητήθηκαν.
- Ο πίνακας **book\_author** αποτελεί ένα weak entity το οποίο δεν μπορεί να σταθεί μόνο του καθώς αποτελείται από δύο foreign keys και ουσιαστικά δεν περιγράφει κάποια φυσική έννοια. Είναι βοηθητικό στην αντιστοιχία των βιβλίων με τους συγγραφείς του. Αυτό είναι αναγκαίο λόγω της υψηλής πιθανότητας ένα βιβλίο να έχει παραπάνω από έναν συγγραφείς. Έχει λοιπόν 2 foreign keys στα Primary keys των οντοτήτων βιβλίο-συγγραφέας.
- Ομοίως το **book\_category** λειτουργεί με την ακριβώς ίδια λογική με το book\_author ως weak entity.
- Ο πίνακας **school\_unit** περιέχει πληροφορίες για κάθε σχολείο της επικράτειας το οποίο έχει βιβλιοθήκη και άρα έχει νόημα να υπάρχει στο σύστημα. Primary key μοναδικό id του.
- **Inventory** είναι ο πίνακας του κρατάει κάθε βιβλίο που υπάρχει στις βιβλιοθήκες αλλά εν αντιθέσει με την οντότητα book, πλέον μιλάμε για τη φυσική παρουσία του βιβλίου. Αυτό μας βοηθάει ώστε να μπορούμε να εξακριβώσουμε την παρουσία του βιβλίου σε κάποια βιβλιοθήκη ώστε να υπάρχει η δυνατότητα δανεισμού του. Έτσι σε κάθε μοναδικό βιβλίο δίνουμε μοναδικό id, ενώ μέσω foreign keys το αντιστοιχίζουμε σε ποιον τίτλο βιβλίου ανήκει το κάθε αντίτυπο και σε ποιο σχολείο της επικράτειας βρίσκεται. Σχετικά με τα constraints των Foreign Keys, ο πρώτος περιορισμός ονομάζεται "inventory\_book\_ISBN\_fk" και αναφέρεται στη στήλη "ISBN" του πίνακα "book". Αυτό σημαίνει ότι η στήλη "ISBN" στον πίνακα "inventory" πρέπει να περιέχει τιμές που υπάρχουν ήδη στη στήλη "ISBN" του πίνακα "book", δημιουργώντας μια σχέση μεταξύ των δύο πινάκων. Ο δεύτερος περιορισμός ονομάζεται "inventory\_school\_unit\_school\_id\_fk" και αναφέρεται στη στήλη "school\_id" του πίνακα "school\_unit". Αυτό σημαίνει ότι η στήλη "school\_id"

στον πίνακα "inventory" πρέπει να περιέχει τιμές που υπάρχουν ήδη στη στήλη "school\_id" του πίνακα "school\_unit", δημιουργώντας μια σχέση μεταξύ των δύο πινάκων.

- Η οντότητα **user** όπως είναι αντιληπτό εκφράζει τους χρήστες της εφαρμογής. Primary key αποτελεί το μοναδικό username κάθε χρήστη. Αξία έχουν δύο enum data types που υπάρχουν στα Attributes status και activity. Το status διευκρινίζει την ιδιότητα του κάθε χρήστη σε σχέση με σχολική εκπαίδευση (μαθητής, δάσκαλος κλπ) αλλά και την ιδιότητα ως προς το ίδιο το σύστημα της βιβλιοθήκης καθώς για την εύρυθμη λειτουργία της έχουν διαμοιραστεί ρόλοι (admin, operator etc) οι οποίοι έχουν και συγκεκριμένες αρμοδιότητες και δικαιώματα εντός εφαρμογής, έτσι αυτό το attribute βοηθάει στο εύκολο διαχωρισμό τους.
- **Rental** είναι ο πίνακας που κρατάει αρχείο όλων των δανεισμών που έχουν γίνει. Ο διαχωρισμός των ενεργών δανεισμών με εκείνους των οποίων η διαδικασία επιστροφής ολοκληρώθηκε γίνεται μέσω του attribute actual\_return\_date όπου για που δεν έχουν επιστραφεί παίρνει τιμή NULL. Οι περιορισμοί foreign key διασφαλίζουν ότι τα δεδομένα που εισάγονται στον πίνακα "rental" είναι συνεπή με τα δεδομένα στους σχετικούς πίνακες ("inventory" και "user"), συμβάλλοντας στην αποφυγή σφαλμάτων στη βάση δεδομένων.
- **Reservation** είναι ο πίνακας που αποθηκεύει τις κρατήσεις που έχουν γίνει. Όταν μια κράτηση δεν έχει πια νόημα (δηλαδή όταν πραγματοποιηθεί ο δανεισμός ή περάσει το χρονικό όριο της κράτησης) τότε το instance του entity διαγράφεται.
- **Review** είναι πίνακας ο οποίος κρατάει τα reviews των χρηστών για κάθε βιβλίο μέσω του Foreign key του ISBN.
- Υπάρχουν και σταθερά views στο DDL.
  - Το operator\_user\_info δίνει την δυνατότητα σε έναν operator να βλέπει τα στοιχεία κάθε χρήστη, αποκρύπτοντας εκείνα τα οποία δεν επιτρέπεται να δει (π.χ. κωδικό πρόσβασης).
  - Το rentals\_per\_school χρησιμοποιείται στο 3.1.1 για πιο ευκολη παρουσίαση των βιβλίων.
  - Το authors name δημιουργήθηκε για να μπορούμε πιο εύκολα (πολύ μικρότερο query στην php) να δείχνουμε σε λίστα τους συγγραφείς των οποίων υπάρχουν βιβλία στο κάθε σχολείο.
  - Το copier\_per\_school μας δείχνει τα αντίτυπα ανά σχολείο για το ερώτημα 3.2.1.
  - Το not\_rented μας βοηθάει στο query 3.1.4 ώστε εύκολα να μπορούμε να εντοπίζουμε βιβλία που δεν έχουν τύχει δανεισμού.

Σε σχέση με τα constraints, επιλέξαμε τις λογικές επιλογές σε κάθε table π.χ. κάθε username να είναι μοναδικό ή κάθε id. Για τα foreign keys μπήκαν constraints για εξασφάλιση της συνέπειας τους με τα parent keys τους.

## DML:

Στο git repo υπάρχει το αρχείο insert\_data.sql με όλα τα δεδομένα που εισήχθησαν στην βάση για δοκιμές στην λειτουργία της.

## Προϋποθέσεις για χρήση της εφαρμογής:

Για να λειτουργήσει η εφαρμογή μας υπάρχουν δύο διαθέσιμοι τρόποι, με εγκατάσταση του XAMPP και εκκίνηση των Apache και MySQL μέσω του Control Panel (όπως προτείνεται από τις διαφάνειες του εργαστηρίου), είτε έχοντας εγκατεστημένη την MySQL, το Apache και την PHP στον υπολογιστή.

Για την εκτέλεση της εφαρμογής χρειάζεται να γίνει clone το repository τοπικά. Αν ο υπολογιστής χρησιμοποιεί xampp αρκεί η εγκατάστασή του στον φάκελο htdocs που βρίσκεται μέσα στον φάκελο xampp όπου αυτός έχει εγκατασταθεί τοπικά. Το link του repository:

[https://github.com/kstathakos1/DB\\_Lab.git](https://github.com/kstathakos1/DB_Lab.git)

Αφού γίνει η εγκατάσταση πρέπει να εκτελεστούν τα αρχεία create\_schema.sql και insert\_data.sql

Κάνοντας copy-paste στον browser (προτείνεται το Google Chrome) το εξής link "[http://localhost/DB\\_lab/school\\_library](http://localhost/DB_lab/school_library)" η εφαρμογή θα είναι σε θέση να λειτουργήσει κανονικά ακολουθώντας τις οδηγίες στο user manual.

Για ανέβασμα φωτογραφιών χρειάζεται η εγκατάσταση του cloundinary.

Το user manual υπάρχει σε ξεχωριστό αρχείο PDF στο repository. Ακολουθεί και παρακάτω:

### User Manual

Αφού εγκατασταθεί σωστά η εφαρμογή και η βάση δεδομένων, για να συνδεθεί καινούριος χρήστης, πρέπει να κάνει εγγραφή πατώντας τον σύνδεσμο: [New user? Please Register](#). Εκεί συμπληρώνει ο χρήστης τα προσωπικά του στοιχεία καθώς και την ιδιότητά του. (μαθητής ή καθηγητής) Αφού εγγραφεί μπορεί να κάνει login χρησιμοποιώντας το username και το password.

Οδηγίες πλοήγησης για μαθητές και καθηγητές:

Η εφαρμογή διαθέτει μια μπάρα πλοήγησης με τις εξής δυνατότητες:

- **Εικονίδιο βιβλιοθήκης:** μπορεί κανείς να αναζητήσει κάποιο βιβλίο που διαθέτει η σχολική του μονάδα βάσει τίτλου, συγγραφέα και κατηγορίας.
- **Books:** λίστα όλων των βιβλίων που διαθέτει η σχολική του μονάδα
- **Authors:** Όλοι οι συγγραφείς των οποίων τα βιβλία βρίσκονται στη σχολική μονάδα
- **Publisher:** Όλες οι εκδόσεις οι οποίες έχουν τίτλους βιβλίων στην σχολική βιβλιοθήκη
- **Category:** Όλες οι κατηγορίες βιβλίων

- **Language:** Όλες οι γλώσσες που είναι γραμμένα τα βιβλία
- **User dropdown menu:** μέσω του μενού μπορεί ο χρήστης να μεταφερθεί στην σελίδα με τα προσωπικά του στοιχεία (profile) και να αποσυνδεθεί (logout)

Τα παραπάνω παρουσιάζονται για το εκάστοτε σχολείο στο οποίο υπάγεται ο χρήστης.

\*οι καθηγητές μπορούν να αλλάξουν τα προσωπικά τους στοιχεία πατώντας πάνω στο κουμπί “CHANGE PROFILE INFORMATION” και μετά μεταφέρονται σε καινούρια φόρμα στην οποία μπορούν να αλλάξουν τις πληροφορίες τους.

Οδηγίες χρήσης για τους Χειριστές (operator):

Η αρχική σελίδα του operator είναι ίδια με αυτή των υπόλοιπων χρηστών και έχει τις ίδιες δυνατότητες με αυτούς. Ωστόσο, στο dropdown menu βρίσκεται άλλη μια δυνατότητα. Συγκεκριμένα υπάρχει η επιλογή **tools** η οποία θα οδηγήσει τον χειριστή σε μια νέα σελίδα με καινούρια μπάρα πλοήγησης (μπάρα διαχείρισης). Από εκεί ο operator μπορεί να οδηγηθεί μέσω αυτής στις εξής σελίδες:

- **Add Book:** στην σελίδα αυτή ο χειριστής μπορεί να προσθέσει ένα βιβλίο συμπληρώνοντας τα πεδία και έπειτα να πατήσει: “ADD BOOK”
- **Rentals of School:** μπορεί να δει όλους τους δανεισμούς που έχουν γίνει στο σχολείο του πατώντας: options->All rentals ενώ μπορεί να δει συγκεκριμένα τους δανεισμούς που έχουν καθυστερήσει πατώντας: options->Delayed Books that havent been returned. Υπάρχει επίσης μια μπάρα αναζήτησης στις οποία μπορεί να αναζητήσει κανείς ένα βιβλίο που έχει δανειστεί με βάση το όνομα του βιβλίου. Για να προσθέσει ο χειριστής ένα βιβλίο πρέπει να πατήσει: options->All rentals->+ADD RENTAL κι έπειτα να συμπληρώσει το username του χρήστη που θέλει να δανειστεί το βιβλίο καθώς και το ISBN του βιβλίου. Έπειτα θα εμφανιστεί μήνυμα που θα αναγράφει αν ο έγινε ή όχι ο δανεισμός ανάλογα με τους περιορισμούς της βιβλιοθήκης.
- **Reservations:** μπορεί να δει όλες τις κρατήσεις που έχουν γίνει στο σχολείο.
- **Users:** Βλέπει όλους τους χρήστες του σχολείου και μπορεί να τους κάνει ενεργούς, ανενεργούς ή να τους διαγράψει εντελώς.
- **Average Review:** μπορεί να αναζητήσει τον μέσο όρο αξιολογήσεων ανά δανειζόμενο και κατηγορία συμπληρώνοντας τα πεδία που εμφανίζονται στην οθόνη και έπειτα πατώντας search.

Για να φύγει ο χειριστής από την μπάρα χειριστή αρκεί να πατήσει το εικονίδιο με την βιβλιοθήκη πάνω αριστερά όπου και θα οδηγηθεί στην αρχική σελίδα.

Οδηγίες χρήσης για τον Διαχειριστή (Admin):

Όπως κι οι χειριστές έτσι κι ο διαχειριστής έχει την ίδια αρχική σελίδα με τους απλούς χρήστες. Ωστόσο στο dropdown menu πάνω δεξιά βρίσκονται οι επιλογές tools και admin tools.

- Πατώντας tools:  
Ο διαχειριστής οδηγείται σε μια νέα σελίδα (rentals of school) και σε μια νέα μπάρα (μπάρα εξερεύνησης διαχειριστή). Σε αυτή μπορεί να δει συγκεκριμένα:
  - **Rentals of School:** Φόρμα αναζήτησης που οδηγεί στην παρουσίαση λίστας με συνολικό αριθμό δανεισμών ανά σχολείο (Κριτήρια αναζήτησης: έτος, ημερολογιακός μήνας)



- **Book Category History:** Στην μπάρα αναζήτησης μπορεί να πληκτρολογήσει ο διαχειριστής μια κατηγορία βιβλίων και γι αυτήν να δει τους συγγραφείς που έχουν γράψει βιβλία της συγκεκριμένης κατηγορίας ενώ πατώντας το κουμπί “FIND TEACHERS” να βρει τους καθηγητές που έχουν δανειστεί βιβλία απ’ τη συγκεκριμένη κατηγορία
- **Young teachers with most rents:** Σε αυτή τη σελίδα εμφανίζονται οι νέοι εκπαιδευτικοί (ηλικία < 40 ετών) που έχουν δανειστεί τα περισσότερα βιβλία και των αριθμό των βιβλίων που έχουν δανειστεί.
- **Obscure Authors:** Εμφανίζονται οι συγγραφείς των οποίων κανένα βιβλίο δεν έχει τύχει δανεισμού
- **Operators with same number:** Εμφανίζονται οι χειριστές που έχουν δανειστεί τον ίδιο αριθμό βιβλίων σε διάστημα ενός έτους με περισσότερους από 20 δανεισμούς.
- **Top 3 Category Pairs:** Εμφανίζονται τα 3 κορυφαία ζεύγη βιβλίων που έχουν τύχει δανεισμού.
- **Runner-up Authors:** Λίστα με όλους τους συγγραφείς που έχουν γράψει τουλάχιστον 5 βιβλία λιγότερα από τον συγγραφέα με τα περισσότερα βιβλία.
- Πατώντας admin tools:
  - **Users:** Σελίδα στην οποία ο διαχειριστής μπορεί να δεχτεί ή να απορρίψει αιτήσεις χειριστών.
  - **Add School:** προσθέτει σχολικές μονάδες.