

Αναφορά PROJECT HY-359

csd4534 Μιχαήλ Σαμψών

csd4111 Κωνσταντίνος Κρούτης

ADMINISTRATOR

- i. Προσβαση στον Admin με τη προσθηκη στο url Admin.html (πχ <http://localhost:8080/Project/admin.html>) .Εισαγωγή με username και password που δινονται από την εκφωνηση. Εγινε μεσω javascript και ελεγχο αυτων.
- ii. Προβολη χρηστών (Librarians , Students) μεσω html request από servlet(Admin.java) οπου ζηταμε(GET) πινακα από Students και πινακα από Librarians μεσα από getStudentTable getLibrarianTable οπου επρεπε να προσθεσω στο est τη databaseToStudents arraylist που επιστρεφει ολους τους μαθητες σε json array αντιστοιχα και για το librarians. Μετα συνενωνω τα δυο jsonarrays και τα επιστρεφω (ΕΔΩ ΥΠΗΡΧΕ ΠΡΟΒΛΗΜΑ καθώς η javascript στην εντολη xhr.onload =function() δεν δουλευε παρολο που όλα τα υπολοιπα λειτουργουσαν φυσιολογικα).
- iii. Τα στατιστικα δεν υλοποιηθηκαν.

LIBRARIAN

- i. Αυτομαστη συνδεση με τα σωστα στοιχεια από την αρχικη σελιδα .
- ii. Εγγραφη στο συστημα μεσω του κουμπιου Register και αυτοματη σωστη αλλαγη και διαχωρισμος από student
- iii. Με τη συνδεση χρηστη:
- iv. Αλλαγη στοιχειων εκτος του username και προβολη των στοιχειων από κατω
- v. Εισαγωγή βιβλιου πληρως λειτουργικη
- vi. Προβολη και αλλαγη στοιχειων βιβλιων
- vii. Προβολη αιτηματων για δανεισμο και επιτρεπει τον δανεισμο δηλαδη status σε borrowed

STUDENT

- i. Αυτομαστη συνδεση με τα σωστα στοιχεια από την αρχικη σελιδα .
- ii. Συνδεση ,καταχωρηση οπως και ο βιβλιοθηκαριος
- iii. Ευρεση Βιβλιων με επιλογες (ενώ η SQL που τροποποιηθηκε μοιαζει σωστη η βαση στελνει μηνυμα λαθους) όμως δουλευει με όλα τα κενα στοιχεια δινοντας όλα τα βιβλια
- iv. Αιτημα για δανεισμο που επιτρεπεται μονο αν το βιβλιο είναι διαθεσιμο

SIMPLE USER

- i. Συνδεση χωρις login.
- ii. Ανοιγει τη σελιδα προβολης βιβλιων με φιλτρα αναζυτησης (όπως Students iii.)

Τα αρχεία:

- i. **HTML**
 - a. **Admin** admin login page
 - b. **borrowBook** students page for borrowing
 - c. **borrowState** librarians page for granting borrowing requests
 - d. **index** starting main frame
 - e. **Librarian** librarians button options
 - f. **loginStuLib** old login page
 - g. **nini** admins option frame to view and delete users and open stats
 - h. **registerBook** librarians registration for books
 - i. **registerStuLib** registration for new user
 - j. **reviewBook** students review page not finished
 - k. **simpleUser** not used
 - l. **Student** students option page frame
 - m. **updateBook** librarians frame to view and update books
 - n. **updateLibInfo** update librarians information
 - o. **updateStInfo** update students info
 - p. **viewBooks** simple books view
- ii. **javascript**
 - a. **Admin**
 - i. **getUser** checks login username password
 - ii. **nextWindow** open nini.html on correct login
 - iii. **viewDelete** to view and or delete users
 - b. **borrowBook**
 - i. **getBooks** get books from borrowed table that are in booksinlibraries table with servlet
 - ii. **\$function** autoloading on refresh boxes on page load
 - iii. **reqBorrow** request book and add to borrowing table ,requested via servlet
 - c. **borrowState**
 - i. **getRequests** see all the borrowing requests
 - ii. **grantRequests** change borrowing state to borrowed

- d. **loginStuLib**
 - i. **getUser** gets user info and checks if he is a student
 - ii. **goForward** if he is student or librarian opens correct html frame page
 - iii. **Register** open html page
 - iv. **Update** open html page
 - v. **BorrowBook** open html page
 - vi. **ReviewBook** open html page
 - vii. **UpdateInfo** open html page
 - viii. **Logout** empty local mem
 - ix. **\$function** autoload on refresh boxes on page load
 - x. **UpdateStudent** update students info
 - xi. **getBooks** getBooks
- e. **registerBook**
 - i. **getBook** get books
- f. **registerBookLooks**
 - i. **check** for valid registration
 - ii. **isbnFunc** shows correct isbn
 - iii. **urlCheck** check url starts with https:
 - iv. **photoCheck** same as above
- g. **registerStuLib**
 - i. **getUser** send user to the database with servlet and display error message or success
- h. **registerStuLibLook**
 - i. **askisi2**
- i. **updateBook**
 - i. **getBooks** get list of boks
 - ii. **UpdateBookPages** update booksinformation
 - iii. **deleteBook** remove book from database with isbn
 - iv. **Logout** local mem forgets filters
 - v. **\$function** autoload on refresh boxes on page load
- iii. **CSS**
 - a. **loginStuLib** used in librarians update book
 - b. **regBook** register book style
 - c. **regStuLib ask2 css**
 - d. **updateBook** same as loginStuLib
 - e. **updateStuLib** mostly used over every page
 - f. **welcomeStyle** index.html style
- iv. **Servlets**
 - a. **Admin**
 - i. **Get** returns json arraylistlibrarians and students to display them
 - ii. **Post empty** would delete users
 - b. **GetBooks**
 - i. **Get** returns jsonarraylist of books
 - ii. **Post empty**
 - c. **LocalStorageExample**
 - i. **A local save spot**

- d. **Register**
 - i. **Connection** checks unique elements
 - ii. **existSt** send which unique field is written two times for student
 - iii. **existLb** send which unique field is written two times for librarian
 - iv. **Post** send all the elements to database to create new user
- e. **RegisterBook**
 - i. **Connection** check if unique elem is in database
 - ii. **Exist** sees if title or isbn exist
 - iii. **mistaken** return which unique element already exists in database
 - iv. **Post** send book to database
- f. **BorrowBooks**
 - i. **Get** from isbn and lib id if book is available and set local storage if it is
 - ii. **Post** get local storage to reuest book if it is available
- g. **borrowState**
 - i. **Get** get all borrowing requests for this library
 - ii. **Post** accept borrow request from specific borrow id
- h. **deeteBook**
 - i. **Post** deleteBook with isbn
- i. **loadBooks**
 - i. **Get** retuen jsonarraylist of books from database
- j. **updateBook**
 - i. **Post** update database in books
- k. **updateStudent**
 - i. **Post** update database in Students