**Софийски университет „Св. Климент Охридски“**

Факултет по математика и информатика

Специалност: “Информационни системи ”

Курс: 3, Група: 1

Дисциплина:

*Разработка на клиент-сървър (fullstack) приложения с Node.js + Express.js + React.js, летен семестър 2019/2020”*

**Курсов Проект**

Book Library

**Изготвил:**

Йордан Богданов Толев (ФН: 71775)

**Преподавател:**

Траян Илиев

София

Летен семестър 2019/2020

1. **Backend API**

Backend-а е реализиран с MVC патърн, като има стремеж към възможно най-високо ниво на абстракция. За това всяка бизнес логика е опитано да бъде изведена в отделен сървис, същото важи и за всякакви мидълуери, които са били използвани.

Има 4те основни ендпоинта, който са комбинирани в един основен – main router

1. **Админ АПИ**

|  |
| --- |
| **adminRouter.post('/createUser', authAdmin.isAdmin, adminCrudController.createUser) – създава юзър** |
|  |

|  |
| --- |
| **adminRouter.post('/deleteUser', authAdmin.isAdmin, adminCrudController.deleteUser); - изтрива юзър** |
|  |

|  |
| --- |
| **adminRouter.post('/changeName', authAdmin.isAdmin, adminCrudController.changeName); - променя името на юзър** |
|  |

|  |
| --- |
| **adminRouter.post('/changePassword', authAdmin.isAdmin, adminCrudController.changePassword); - смяна на паролата** |
|  |

|  |
| --- |
| **adminRouter.post('/changeRole', authAdmin.isAdmin, adminCrudController.changeRole); - смяна на ролята на даден юзър** |
|  |

|  |
| --- |
| **adminRouter.post('/addAuthorToBook', authAdmin.isAdmin, adminCrudController.addAuthorToBook); - добавя автор на книга** |
|  |

|  |
| --- |
| **adminRouter.post('/addBookToAuthor', authAdmin.isAdmin, adminCrudController.addBookToAuthor); - добавя книга към автор** |
|  |

|  |
| --- |
| **adminRouter.post('/removeComment/:bookId', authAdmin.isAdmin, adminCrudController.removeComment); - премахва коментар от книга** |
|  |

|  |
| --- |
| **adminRouter.get('/getAllComments/:bookId', auth.isAuthenticated, adminCrudController.getAllComments); - взима всички коментари за всяка книга** |
|  |

**adminRouter.post('/editBook/:bookId', authAdmin.isAdmin, adminCrudController.editBook); = едитва информациата на дадена книга**

1. **Юзър АПИ**

|  |
| --- |
| **router.post('/register', userController.register);** |
| **-регистрация** |
|  |

|  |
| --- |
| **router.post('/login' ,userController.login);** |
| **-логин** |
|  |

|  |
| --- |
| **router.post('/edit', auth.isAuthenticated , userController.edit);** |
| **-едитване на профил** |
|  |

|  |
| --- |
| **router.post('/logout' , auth.isAuthenticated, userController.logout);** |
| **логаутване** |
|  |

|  |
| --- |
| **router.get('/userProfile' , auth.isAuthenticated, userController.userProfile);- преглед на профила** |
|  |

|  |
| --- |
| **router.post('/addCommentToBook/:bookId' , auth.isAuthenticated, userController.addCommentToBook); - добавяне на коментар към книга** |
|  |

|  |
| --- |
| **router.post('/addRating/:bookId' , auth.isAuthenticated, userController.addRating); - добавя рейтинг към книга** |
|  |

|  |
| --- |
| **router.get('/addToWishList/:bookId' , auth.isAuthenticated, userController.addBookToWishList); - добавя книга към исканите на юзър** |
|  |

|  |
| --- |
| **router.get('/removeFromWishList/:bookId' , auth.isAuthenticated, userController.removeBookToWishList); - премахване от искани книги** |
|  |

**router.get('/wishList' , auth.isAuthenticated, userController.getWishList);**

* **Показва всички книги в исканите**

1. **Book API**

|  |
| --- |
| **booksRouter.post('/comment', auth.isAuthenticated ,commentscontoller.comment);** |
|  |

|  |
| --- |
| **booksRouter.post('/addBook' , authAdmin.isAdmin ,booksController.createBook); - добавя се книга** |
|  |

|  |
| --- |
| **booksRouter.get('/books' , booksController.getBooks); -показват се всички книги** |
|  |

|  |
| --- |
| **booksRouter.get('/books/:id' , booksController.getBookById);** |
| * **Показва се книга по ИД** |
|  |

|  |
| --- |
| **booksRouter.get('/booksByName/:name' , auth.isAuthenticated, booksController.getBookByName);**   * **Книгата се изкарва по име** |
|  |

|  |
| --- |
| **booksRouter.get('/bookAuthor/:id' , auth.isAuthenticated, booksController.getBookAuthor); - показва се автора на книга** |
|  |

|  |
| --- |
| **booksRouter.get('/getAvarage/:bookId' , auth.isAuthenticated, booksController.getAvgRating); - взима се средния рейтинг спрямо всички гласували** |
|  |

|  |
| --- |
| **booksRouter.post('/deleteBook' , authAdmin.isAdmin ,booksController.deleteBook);**  **-трие се книга** |
|  |

|  |
| --- |
| **booksRouter.get('/mostRecent' , auth.isAuthenticated, booksController.mostRecent); - показва най-скорошно добавените книги** |
|  |

|  |
| --- |
| **booksRouter.get('/userDeleteComment/:bookId' , auth.isAuthenticated, booksController.userDeleteComment);**  **Човек може да си изтрие коментара на книга** |
|  |

**booksRouter.post('/easyCommentDelete' , auth.isAuthenticated ,booksController.userDeleteCommentEasy);**

По лесен начин за същата имплементация

1. **Аuthor API**

|  |
| --- |
| **authorRouter.post('/deleteAuthor' , authAdmin.isAdmin, authorController.deleteAuthor);**   * **Изтрива се автор** |
|  |

|  |
| --- |
| **authorRouter.post('/addAuthor' , authAdmin.isAdmin , authorController.createAuthor);**   * **Добавя се автор** |
|  |

|  |
| --- |
| **authorRouter.get('/authors' , auth.isAuthenticated ,authorController.getAuthors);**  **-лист от всички автори** |
|  |

|  |
| --- |
| **authorRouter.get('/authors/:id' ,auth.isAuthenticated ,authorController.getAuthorbyId);**  **-взима се само един автор** |
|  |

|  |
| --- |
| **authorRouter.get('/authorsbyName/:name' ,auth.isAuthenticated, authorController.getAuthorbyName);**   * **Взима се автор по име** |
|  |

**authorRouter.get('/authorsBooks/:id' ,auth.isAuthenticated ,authorController.getAuthorBooks);**

* **Визмат се всички книги на един автор**

1. **What technologies have been user**

* **For the frontend – material ui, bootstrap for styling**
* **React, React router dom – for components and for routing**
* **Redux, react-redux – for having a single sorce of truth**
* **MongoDB – for a database**
* **Mongoose – for easier handling the MongoDB**
* **Passport – for authentication**
* **Redis – as a backend store**
* **Bcrypt – for hashing passwords**