

Лабораториска вежба 2

Потребно е да се изработи еден модул од апликација за изнајмување на книги од одредена библиотека, односно модулот кој би го користел корисник со *role librarian*, кој би имал неколку привилегии:

- Да додава нови книги кои може да се изнајмат
- Да брише книги кои повеќе не се во добра состојба и нема да се изнајмуваат
- Да изменува одреден запис за книга
- Да обележи одредена книга како изнајмена

Она што ќе ви биде потребно е **Spring Boot апликација**, односно API кое ќе го креирате според соодветните барања, како и React апликација која ќе биде задолжена за приказ на податоците и интеракција со корисниците. Во однос на базата на податоци, потребно е да користите **H2** база.

Во рамки на апликацијата се чуваат следните податоци за книгите: *id* (Long), *name* (String), *category* (enum), *author* (Author), *availableCopies* (Integer). Категоријата на книгата може да биде: NOVEL, THRILER, HISTORY, FANTASY, BIOGRAPHY, CLASSICS, DRAMA. За секој автор пак се чуваат податоците: *id* (Long), *name* (String), *surname* (String), *country* (Country). За секоја земја се чуваат податоците: *id* (Long), *name* (String), *continent* (String).

Користејќи слоевита архитектура треба да се дефинираат соодветни контролери, сервиси и репозиториуми.

React апликацијата треба да биде онаа која ќе ги прави соодветните барања кон вашето API. Внимавајте да избегнете CORS error.

Оваа апликација треба да има сличен изглед како апликацијата од аудиториски вежби, односно:

- На почетната страна се прикажани сите книги (може да се видат на патека / и /books)
 - За секоја книга овозможено е копче Edit, Delete, Mark As Taken
 - На клик на копчињата Edit/ Delete се прави повик кон API и соодветно да се измени/ избрише книгата
 - На клик на копчето Mark As Taken, потребно е да се направи повик кон API и соодветно да се намали бројот на availableCopies
 - Дополнително има едно копче Add a new book, со кое се овозможува додавање на нова книга
- На друга страна се прикажани сите категории (може да се видат на патека /categories)
- Постои header дел кој всушност е мени за навигација

За React апликацијата, освен рутирање, потребно е барем еднаш да искористите Hook, на пример useState. Дополнително, за прикажувањето на книгите треба да се имплементира пагинација, т.ш. по 5 книги ќе се прикажуваат на секоја страна.

Прикачување на лабораториската вежба 2

Двете апликации (Spring Boot и React) кои ќе ги креирате за изработка на оваа лабораториска вежба, потребно е да ги поставите на одделно repository на GitHub (или GitLab).

Дополнително, потребно е да ги поставите апликациите на [Heroku](#). Има повеќе начини како да го направите ова, во прилог ќе биде објаснет еден од начините.

Најпрво, креирајте account на Heroku. Потоа, откако сте ги поставиле двете апликации на GitHub, навигирајте до Dashboard. Тука изберете Create new app и креирајте најпрво една апликација (која ќе ви биде врзана со backend апликацијата). За да креирате апликација потребно ќе биде да внесете само име и регион (внесете Europe).

Доколку работите со Java 11, потребно е да додадете фајл со име system.properties во root фолдерот на апликацијата и во него да ставите само една линија код: `java.runtime.version=11`. Направете commit + push. Доколку користите друга верзија на Java, можете да го прескокнете овој чекор.

Понатаму, на Heroku, избирате Deploy, па избирате опција GitHub, по што потребно е да се поврзете со она repository на GitHub каде ви е поставена Spring Boot апликацијата. Изберете ја master гранката и селектирајте Enable Automatic Deploys и Deploy Branch.

По завршувањето на овој процес, на линкот кој ќе ви биде прикажан, ќе може да ја пристапите апликацијата хостирана на Heroku. Доколку имате било каков проблем, најдобро е да симнете Heroku CLI за да може соодветно да следите логови за вашата апликација. [На овој линк имате повеќе детали за овој дел.](#)

Следно, на истиот начин како што креиравме Heroku апликација за Spring Boot апликацијата, истото го правиме и за React апликацијата. Единствената разлика тука ќе биде тоа што ќе биде потребно во Settings, па Buildpacks, да додадете нов buildpack. Додадете го овој buildpack <https://github.com/mars/create-react-app-buildpack>. Потоа следуваат истите чекори за Deploy.

За крај, доколку е се во ред, потребно е да ги промените линковите во рамки на двете апликации, односно, секаде каде се користи повик до localhost сега потребно е да се стави линкот на хостираната апликација. Вакви промени ќе имате во @CrossOrigin на backend, како и во baseUrl кој се користи за axios повиците на frontend.

Како што кажавме, ова е само еден начин за да постават апликациите на Heroku, истото може да го направите и преку CLI итн.

Без разлика на кој начин ќе ги поставите апликациите, решението на оваа вежба треба да го поставите **САМО** како одговор на анкетата која е поставена на курсот.