Вітаю!

Найкращою демонстрацією моїх навичок є робота, яку я виконав, тому я детальніше опишу реалізацію курсового проекту.

В курсовому проекті реалізований односторінковий веб-додаток інтернет-книгарні.

Додаток дозволяє виконувати пошук книг за частиною назви; фільтрувати добірку книг за чотирма ціновими діапазонами або певною ціною; сортувати книги у порядку зменшення та збільшення за артикулом або за ціною. На сторінці окремої книги можна за тематичним тегом перейти до списку книг з відповідною до тегу тематикою, таким чином у додатку також реалізовано сортування і за тематичними тегами. На сторінці окремої книги додатково є посилання в Google-пошуковик за ключовими словами назви книги. Як на сторінці окремої книги, так і в кошику виконується автоматичний підрахунок суми замовлення конкретної книги в залежності від кількості, у кошику крім цього виконується підрахунок загальної суми замовлення всіх книг та їх кількості. Для зручності у кошику додатково реалізований перехід із кошика на сторінку окремої книги.

Веб-додаток реалізований за стратегією «Mobile First» з адаптивно-респонсивною версткою з наступними точками перелому:

* значення за замовчуванням - для смартфонів з шириною екрану менше 320 px (в реальності - 200...319 px);
* 320 px - для смартфонів з шириною екрану 320...419 px;
* 420 px - для смартфонів та планшетів з шириною екрану 420...799 px;
* 800 px - для стаціонарних комп'ютерів та ноутбуків з шириною екрану 800...1023 px;
* 1024 px - для стаціонарних комп'ютерів та ноутбуків з шириною екрану 1024...1599 px або при використанні допоміжної точки перелому - 1024...1199 px;
* 1200 px - допоміжна точка перелому для настільних комп'ютерів та ноутбуків з шириною екрану 1200...1599 px;
* 1600 px - для настільних комп'ютерів, ноутбуків, телевізорів, проекторів з шириною екрану понад 1600 px.

Проект створений з використанням фреймворку React у поєднанні з бекендом Node.js на базі сервера на фреймворку Express, підключеного до бази даних MongoDB.

Проект веб-сервера знаходиться за адресою [github.com/Eduard-Konovka/feb101-course-task-api](https://github.com/Eduard-Konovka/feb101-course-task-api)

Веб-додаток використовує бекенд-сервер за адресою [feb101-course-task-api-eduard-konovka.onrender.com](https://feb101-course-task-api-eduard-konovka.onrender.com), відповідно, як базовий URL для доступу до API використовується змінна оточення REACT\_APP\_URL, яка відповідає цій адресі.

Нажаль, у зв'язку з тим, що для навчальних цілей використовується безкоштовний тариф бекенд-хосту [Render](render.com), ***перше*** завантаження списку книг виконується з затримкою, будь ласка,

**ДОЧЕКАЙТЕСЯ ПЕРШОГО ЗАВАНТАЖЕННЯ СПИСКУ КНИГ!**

Для взаємодії з бекендом використовується REST API з наступними ендпоїнтами:

* ендпоїнт `/api/books` використовується для отримання списку книг;
* ендпоїнт `/api/orders` використовується для надсилання замовлень.

Замовлення надходить до бази даних MongoDB. Замовлення я відстежую за допомогою додатку бази даних MongoDB Compass. Зазвичай у стандартних інтернет-магазинах налаштовують ще й поштовий клієнт, який автоматично надсилає власнику магазину замовлення на електронну пошту. Таке мені теж доводилося реалізовувати, проте це виходить за межі курсового проекту, тому я це поки що опустив.