Отчет. Задание № 3. GraphQL.

Левченко Антон, 43506/3

1. **Разработать приложение по продаже товаров (любые на свое усмотрение), позволяющее пользователю производить покупки в разных валютах (в идеале в любой).**
   1. **Информация о товарах и продажах хранится в базе данных приложения**
   2. **Информация о валютах получается через API (можно взять API ЦБ РФ, *подробнее здесь <https://www.cbr-xml-daily.ru/>*)**

Приложение написано с использованием технологий angular 7 для клиентской части и nodejs express для сервера.

Репозиторий: <https://github.com/antonylevchenko/ui_task_3_GraphQL>

Для запуска приложения необходимо запустить все сервисы:

**node server/businessLogic/rate-service.ts**

**node server/businessLogic/product-service.ts**

**node server.ts**

Для запуска клиентской части необходимо установить зависимости командой **npm run install** и собрать приложение командой **npm run build**. После этого оно будет доступно по адресу localhost:5000 (при запущеном server.ts).

Для разработки с использованием angular.cli и подхватом изменений на лету можно воспользоваться командой npm run start, приложение будет доступно по адресу localhost:4200

Имеется два сервиса:

**rate-service** (server/businessLogic/rate-service.ts) отвечает за получение информации о курсах валют через API ЦБ РФ и перевод цен из рублей в другие валюты. Запускается как отдельное приложение.

**product-service** (server/businessLogic/product-service.ts) отвечает за хранение и выдаёт информацию о товарах. Данные хранятся в базе mongoDb. Запускается как отдельное приложение.

1. **Реализовать прослойку между клиентом и сервером на основе GQL, которая будет управлять запросами для получения товаров и валют.**

**server.ts** содержит логику по обработке очередей graphQl на сервере. Запускается как отдельное приложение, связывается с другими сервисами по мере необходимости.

1. **Ели пользователь запрашивает товар в рублях, все данные берутся только из БД приложения, если пользователь запрашивает данные в иной валюте, данные о валюте берутся из кэша (если есть) или из API.**

За эту часть логики отвечает запрос graphQL price.

Немного скриншотов:





