**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №5**

**по дисциплине «WEB-технологии»**

Тема: Модуль администрирования приложения

«Биржа акций»

|  |  |  |
| --- | --- | --- |
| Студент гр. 0304 |  | Максименко Е.М. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2022

## Цель работы.

Целью работы является изучение возможностей применения библиотеки React (https://reactjs.org/) для разработки интерфейсов пользователя web-приложений и использование фреймворка NestJS (https://nestjs.com/) для разработки серверных приложений.

## Задание.

Необходимо создать web-приложение, обеспечивающее настройку биржи брокера, в которой есть возможность задать перечень участников, перечень акций, правила изменения акций во времени. Основные требования следующие:

1. Информация о брокерах (участниках) и параметрах акций сохраняется в файле в формате JSON.

2. В качестве сервера используется NestJS с использованием языка TypeScript.

3. Предусмотрена HTML-страница с перечнем потенциальных брокеров. Брокеров можно добавлять и удалять, можно изменить начальный объем денежных средств.

4. Предусмотрена HTML-страница для перечня акций. Есть возможность просмотреть перечень доступных акций (обозначение, название компании) и исторические данные по изменению курса не менее чем за текущий и предыдущий год. Есть возможность выбрать какие акции будут участвовать в торгах. Минимально должны поддерживаться следующие компании (в скобках – обозначение): Apple, Inc. (AAPL), Starbucks, Inc. (SBUX), Microsoft, Inc. (MSFT), Cisco Systems, Inc. (CSCO), QUALCOMM Incorporated (QCOM), Amazon.com, Inc. (AMZN), Tesla, Inc. (TSLA), Advanced Micro Devices, Inc. (AMD).

Реальные исторические данные по изменению курса доступны по адресу: https://www.nasdaq.com/market-activity/quotes/historical.

Фрагмент данных для AAPL за три дня (переведён в формат json, оставлены только два столбца: дата и стоимость на время начала торгов):

[{«date»: «11/5/2021», «open»: «$151.89},

{«date»: «11/4/2021», «open»: «$151.58»},

{«date»: «11/3/2021», «open»: «$150.39»}]

5. Предусмотрена HTML-страница для настроек биржи (дата начала торгов, скорость смены дат в секундах при имитации торгов). На этой же странице должна быть кнопка «Начало торгов», которая запускает процесс имитации торгов и предоставление информации об изменении курсов акций всем брокерам по web-сокетам с учётом заданных настроек биржи, здесь же должна отображаться текущая имитируемая дата торгов и текущая стоимость каждой акции.

6. Все элементы в клиентском приложении реализованы с использованием компонентов React. Маршрутизация реализована с использованием «react-router-dom».

7. Для хранения общих данных используется Redux.

8. На сервере спроектированы компоненты и сервисы NestJS для имитации торгов и обработки запросов клиентского приложения.

9. Исторические данные по котировкам представляются как в виде таблиц, так и в виде графиков (например, с использованием Chart.js).

10. Приложение должно реализовывать responsive-интерфейс и корректно работать в том числе при просмотре с мобильного телефона.

11. Для всех страниц web-приложения разработан макет интерфейса с использованием Figma (https://www.figma.com/).

Преимуществом будет создание и использование аутентификации на основе passport.js (http://www.passportjs.org/).

Преимуществом будет использование Material UI React

(https://mui.com/ru/).

## Основные теоретические положения.

React – библиотека на JavaScript для построения интерфейса пользователя. React представляется удобным инструментом для создания масштабируемых web-приложений (в данном случае речь идет о клиентской части), особенно в тех ситуациях, когда приложение является одностраничным.

В основу React заложены принципы Redux, предлагающее предсказуе-

мый контейнер хранения состояния web-приложения.

Вся структура веб-страницы может быть представлена с помощью DOM. Для решения проблемы производительности предложена концепция виртуального DOM, который представляет собой облегченную версию DOM. React работает именно с виртуальным DOM. Реализован механизм, который периодически сравнивает виртуальный DOM с реальным и вычисляет минимальный набор манипуляций для приведения реального DOM к состоянию, которое хранится в виртуальном DOM.

NestJS – фреймворк для разработки серверных приложений на языках

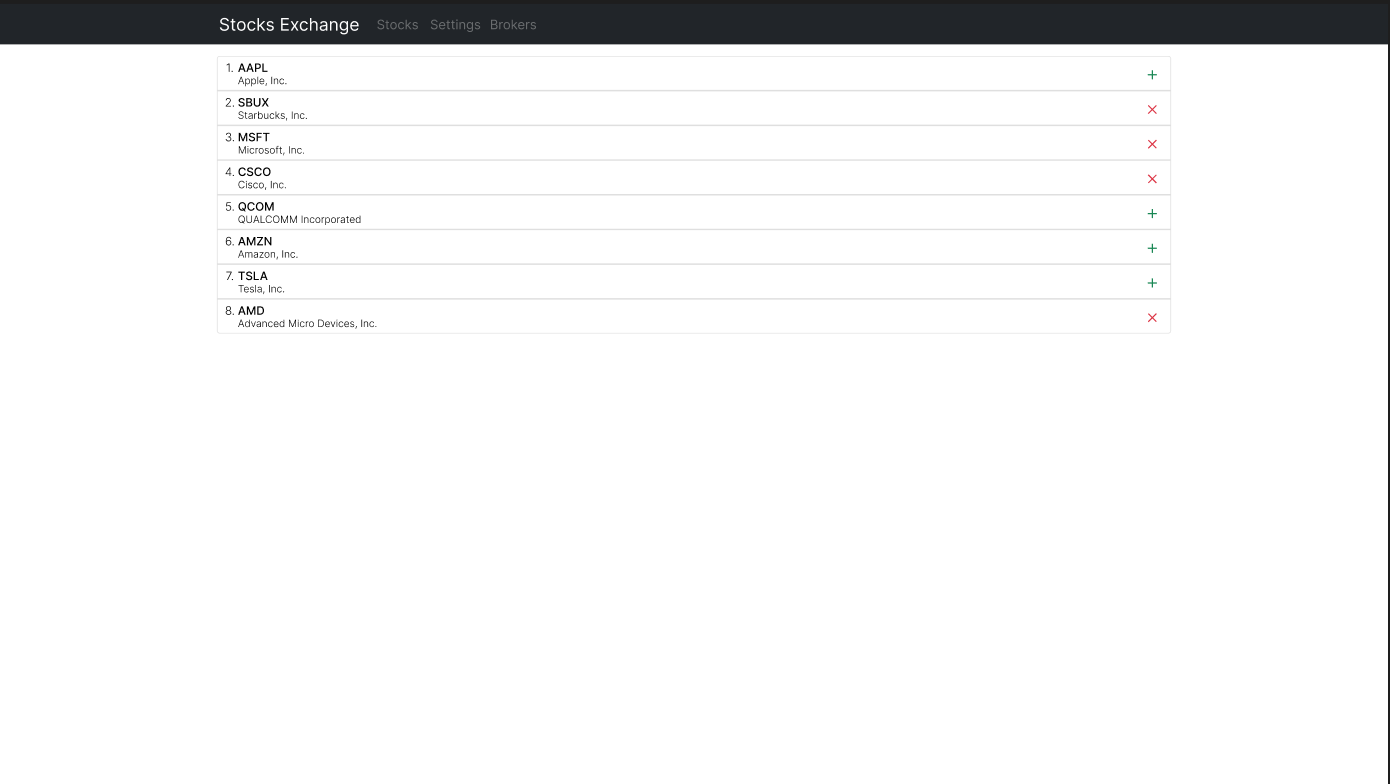
JavaScript и TypeScript. Фреймворк построен на основе компонентного подхода и предлагает стандартизованную структуру приложения по аналогии с Angular.

## Выполнение работы.

1. Верстка макетов страниц в *Figma*.

С помощью *Figma* были разработаны макеты страниц модуля администрирования.

Страница акций содержит в себе список всех акций, при нажатии на которые открывается окно с информацией о конкретной акции. Также на данной странице можно выбрать, будет ли участвовать акция в эмуляции торгов. Макет страницы акций см. на рис. 1.

Рисунок 1. Макет страницы акций

Страница просмотра акций содержит информацию об акции и о ее котировках в виде таблицы (см. рис. 2) и в виде графика (см. рис. 3).

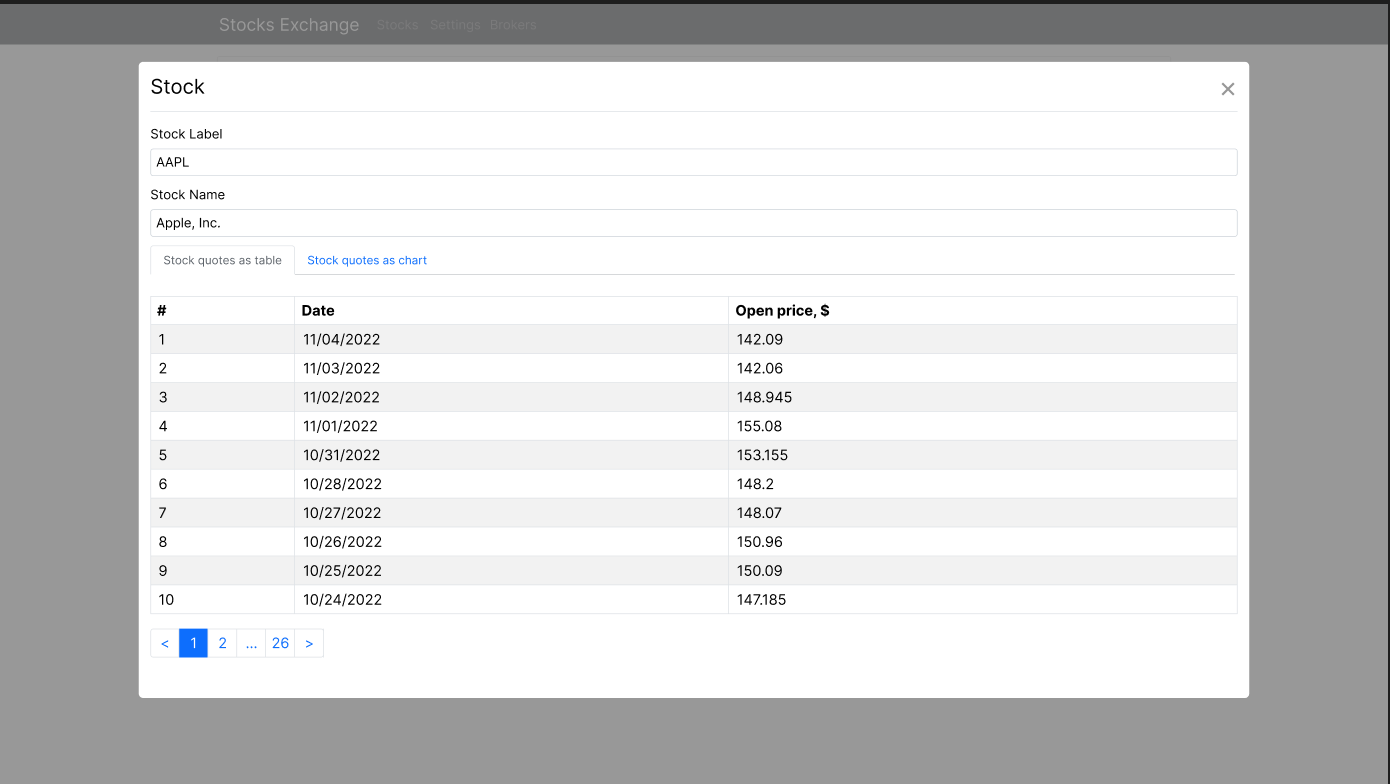
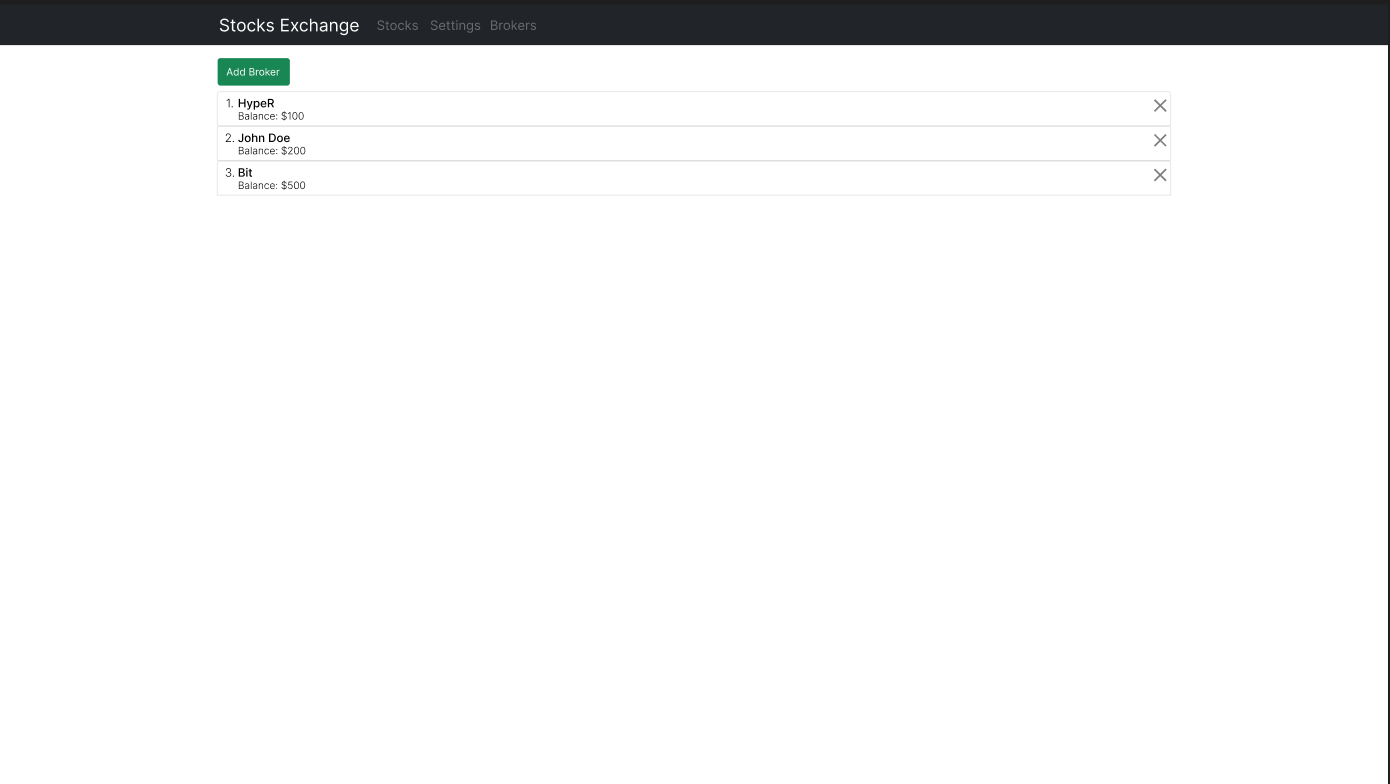
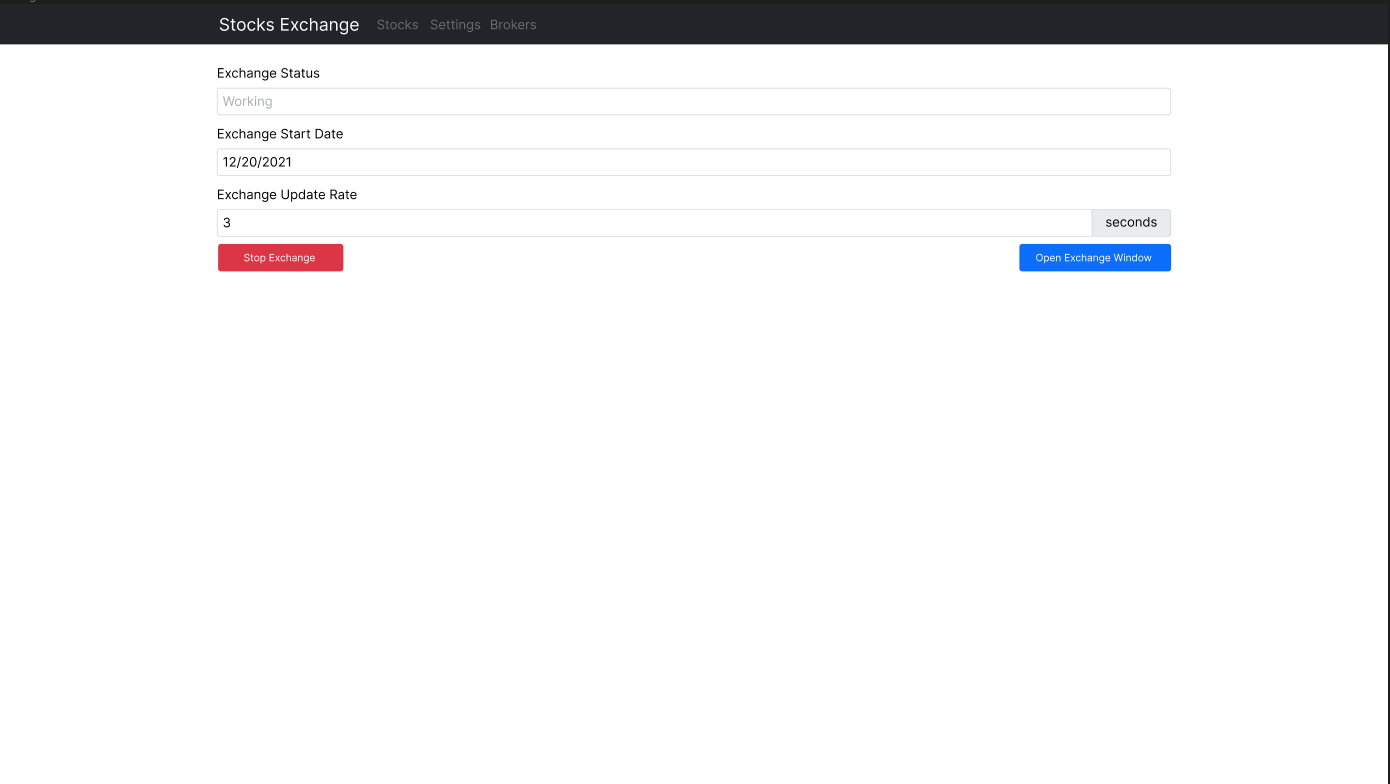
Рисунок 2. Страница просмотра акций с таблицей котировок

Рисунок 3. Страница просмотра акций с графиком котировок

Страница брокеров содержит список брокеров, нажав на которых можно отредактировать данные о выбранном брокере. Также есть возможность создать нового брокера. Страницу брокеров см. на рис. 4.

Рисунок 4. Страница брокеров

Страница настроек биржи содержит информацию о том, работает ли эмуляция в данный момент и настройки эмуляции, такие как дата начала торгов и время обновления дат в секундах. Также на странице настроек есть кнопка начала/остановки торгов и кнопка открытия окна с информацией о котировках на данный день торгов, если эмуляция работает. Страницу настроек см. на рис. 5.

Рисунок 5. Страница настроек биржи

2. Создание серверной части приложения.

С помощью фреймворка Nest.js с использованием языка TypeScript была создана серверная часть приложения. Приложение поделено на компоненты, такие как компонент авторизации (на данный момент не задействован), брокеров, котировок, акций, пользователей и эмуляции. Связь с клиентом осуществляется посредством веб-сокетов. Данные о бирже, акциях и их котировках, пользователях и брокерах хранятся в JSON файлах.

3. Создание клиентской части приложения.

Клиентская часть приложения была создана с использованием библиотеки React. Были реализованы все страницы из п.1. Для верстки был использован компонент react-bootstrap, что позволило сверстать адаптивные страницы. Для хранения состояния клиентской части приложения использовалось Redux-хранилище. График котировок акций выводился с использованием библиотеки Chart.js.

**Выводы.**

В ходе данной работы был разработан модуль администрирования приложения «Биржа акций». Для серверной части модуля был использован фреймворк Nest.js, а также Socket.io. Для клиентской части модуля была использована библиотека React.