**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра МО ЭВМ**

**ОТЧЕТ**

**по лабораторной работе №6**

**по дисциплине «Web-технологии»**

**Тема: Модуль приложения «Покупка и продажа акций»**

| Студент гр. 1303 |  | Чубан Д.В. |
| --- | --- | --- |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2023

## Цель работы.

Целью работы является изучение возможностей применения фреймворка Vue (https://v3.ru.vuejs.org/ru/) для разработки интерфейсов пользователя web-приложений и организации E2E тестирования клиентской части приложения.

## Задание.

Необходимо создать web-приложение, обеспечивающее работу брокера, у него есть запас денежных средств, он имеет возможность купить или продать акции (любое доступное количество), а также контролировать изменение котировок акций. В приложении должен отображаться баланс (запас денежных средств плюс стоимость акций), а также прибыль или убыток, которые он получил по каждой акции. Основные требования следующие:

1. Приложение получает исходные данные из модуля администрирования приложения «Биржа акций» в виде настроек в формате JSON-файла и в виде данных от web-сокета по изменению стоимости акций во времени.

2. В качестве сервера используется NestJS.

3. Участники торгов подключаются к приложению «Покупка и продажа акций».

4. Предусмотрена HTML-страница администратора, на которой отображается перечень участников. Для каждого участника отображается его баланс, количество акций каждого типа у каждого участника и его прибыль или убыток по каждой акции в текущий момент времени.

5. Предусмотрена HTML-страница входа в приложение, где каждый участник указывает (или выбирает из допустимых) свое имя.

6. Предусмотрена HTML-страница, на которой участнику отображаются:

– текущая имитируемая дата;

– текущая стоимость каждой из акций, выставленных на торги;

– общее количество доступных средств;

– количество, стоимость и прибыль/убыток по каждой купленной акции. На этой же странице у брокера есть возможность:

– открыть диалоговое окно просмотра графика изменения цены каждой акции (с момента начала торгов до текущего момента) с учётом сообщений об изменении стоимости акций;

– купить/продать интересующее его количество акций. Комментарии:

– брокер не может купить акции, если денег не хватает;

– купля/продажа происходит «мгновенно».

7. Разработаны автоматизированные тесты для проверки корректности работы клиентской части web-приложения с использованием headless-браузера или фреймворка Selenium. Как минимум необходимо проверить, что при покупке/продаже N акций в определённую дату соответствующим образом изменяется баланс средств брокера и через некоторое время получается правильная прибыль/убыток по данной акции.

**Основные теоретические положения.**

Vue – фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создавался пригодным для постепенного внедрения. Его ядро решает задачи уровня представления, упрощая интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue подходит и для разработки сложных одностраничных приложений.

Selenium – фреймворк тестирования web-приложений, позволяет программировать автоматизированные тесты клиентской части web-приложений, а также записывать и воспроизводить действия пользователей.

Headless-браузер – браузер без интерфейса пользователя, предназначенный для тестирования web-приложений.

## Выполнение работы.

Для всех страниц модуля пользователя были разработаны макеты интерфейса с использованием Figma. В приложении А к отчету они представлены.

Создание каркаса web-приложения с использованием Vue.

*Файл Router/index.js*

Создание и экспорт объекта маршрутизатора (router) Vue, используя Vue Router.

* createRouter и createWebHashHistory: импортируются функции createRouter и createWebHashHistory из библиотеки Vue Router. createWebHashHistory создает экземпляр истории маршрута с использованием хэша в URL.
* routes: определяется массив маршрутов. Каждый маршрут представлен объектом с параметрами, такими как path (путь URL), name (имя маршрута), и component (компонент Vue, который будет отображен при переходе по этому маршруту).
* router: создается экземпляр маршрутизатора с использованием createRouter, передавая ему объект истории и массив маршрутов.
* export default router: экспортируется созданный маршрутизатор для использования в основном приложении Vue.

*Файл Store.js*

Создание и экспорт объекта хранилища (store) Vuex, который используется для управления состоянием приложения во Vue.js.

* createStore: импортируется функция createStore из библиотеки Vuex.
* state: определяется начальное состояние хранилища. В данном случае, состояние содержит два свойства: tradingList (список акций, участвующих в торгах) и broker.
* mutations: определяются мутации, которые представляют собой синхронные функции для изменения состояния.

В данном случае:

* setTradingList: устанавливает новый список акций, участвующих в торговли, в состояние.
* setBroker: устанавливает нового брокера в состояние.

*Файл main.js*

Создание и настройка приложения Vue.

*Файл Welcome.vue*

Компонент страницы входа, в нем описана форма выбора брокера из списка, а также дизайн страницы.

*Файл BrokerCard.vue*

Представляет собой карточку брокера, которая используется на странице брокеров. В данном компоненте реализована логика вычисления профита брокера, и получения стоимости конкретной акции.

*Файл StockCard.vue*

Представляет собой карточку акции, включающую в себя график стоимости акции от времени и информацию о цене акции.

*Файл StockChart.vue*

Представляет собой компонент с графиком стоимости акции, который затем используется в карточке акции.

*Файл StockSummary.vue*

Компонент для вывода на экран цены акции.

*Файл StockView.vue*

Страница торговли акциями, на ней расположено выпадающее окно с акциями, участвующими в торгах, в этом компоненте используется компонент StockCard, кроме того на странице расположены кнопки покупки и продажи акции, а так же поле ввода количества акций к покупке и продаже

*Файл BrokerView.vue*

Представляет собой страницу со всеми брокерами на бирже. В шаблоне компонента используется контейнер, в который с помощью цикла добавляются карточки брокеров, при обновлении стоимости акций, соответствующая информация обновляется и на странице.

Изображения с основными страницами приложения представлены в приложении A.

## Выводы.

В ходе лабораторной работы был разработан модуль приложения «Покупка и продажа акций».

Были созданы макеты и дизайн интерфейса web-приложения в *Figma.* Созданы страницы веб-приложения, включающие в себя следующие компоненты. Стартовая страница: вход в приложение пользователей для возможности покупки и продажи акций. Страница брокера: данные брокера (имя, баланс, прибыль и убытки по акции), возможность отслеживания текущей даты и стоимости соответствующих акций с возможностью их покупки, продажи и просмотра графика. Страница администратора: просмотр данных всех брокеров и информации о их прибыли и убыткам по купленным акциям. С помощью Selenium Организовано тестирование клиентской части web-приложения.

# Приложение А

Макеты интерфейса.

На рисунке 1 – макет страницы входа.

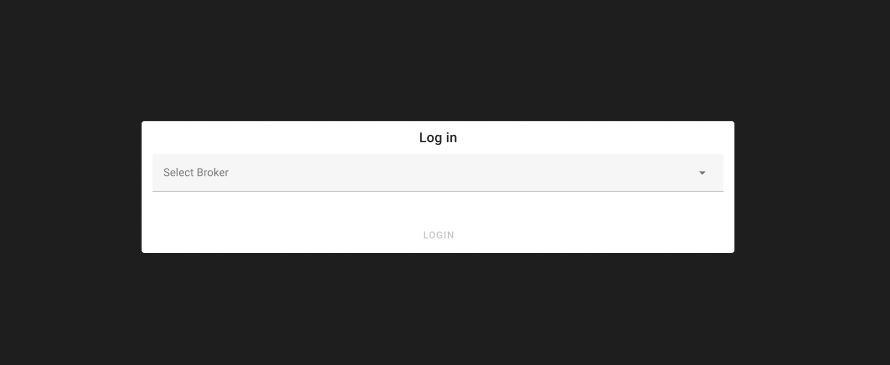


Рисунок 1 – макет страницы входа.

На рисунке 2 – макет страницы брокера.

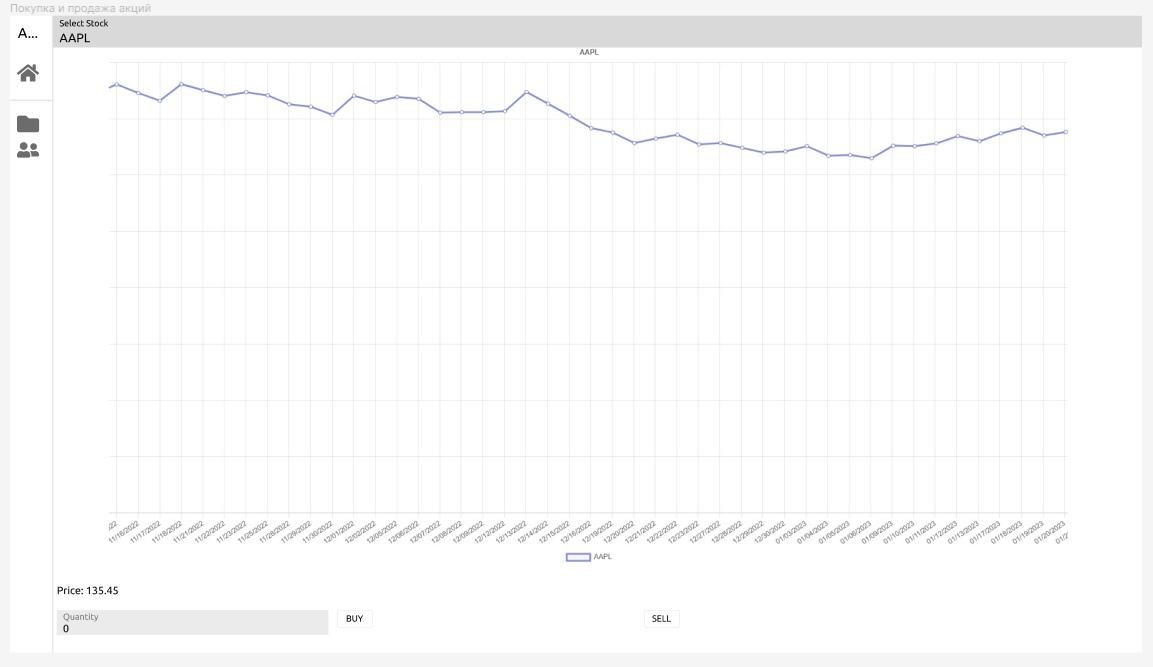


Рисунок 2 – макет страницы брокера.

На рисунке 3 – макет страницы всех брокеров.



Рисунок 3 – макет страницы всех брокеров.

Модуль приложения «Покупка и продажа акций».

На рисунке 4 – стартовая страница.



Рисунок 4 – стартовая страница.

На рисунке 5 – страница брокера.



Рисунок 5 – страница брокера.

На рисунке 6 – страница со списком всех брокеров.

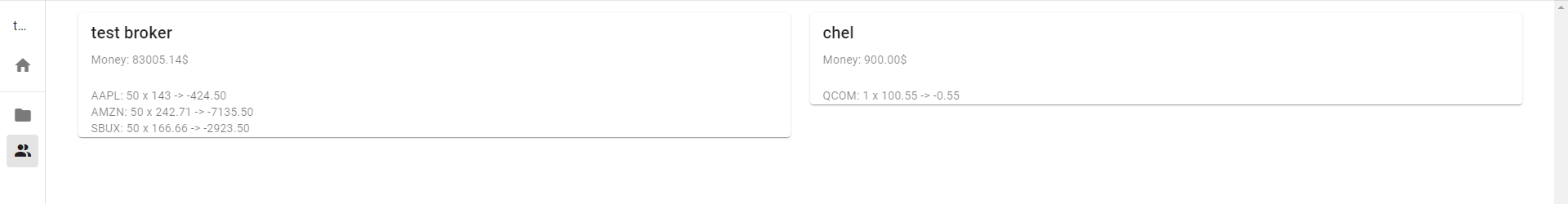


Рисунок 6 – страница со списком всех брокеров.