

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

Коренев Д.А.

Преподаватель

Беляев С.А.

Санкт-Петербург

2023

Цель работы.

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 и убыток, которые он получил по каждой акции.

Основные требования:

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 учетом 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 определенную дату соответствующим образом изменяется баланс средств брокера и через некоторое время получается правильная прибыль/убыток по данной акции.

Выполнение работы.

Vue:

Для навигации между страницами был создан Router, в котором для каждой страницы были написаны путь, имя и компонент, который отображает страницу. Для подключения Router в компонент App была добавлена строка `<RouterView/>`.

Компонент Login создан для возможности входа в аккаунт брокера. Разметка страницы написана внутри `<template>` и состоит из поля ввода имени, кнопки входа, списка возможных имен. Для корректной работы компонента внутри тега `<script>` был написан код. BrokersSocket делает emit “findAll” для получения всех логинов брокеров, метод filterLogins фильтрует логины, чтобы

предоставить пользователю выбор подходящих логинов. По нажатию кнопки входа, с использованием `router`, пользователь входит в аккаунт брокера с вписанным логином.

Компонент `Account` отображает информацию об аккаунте пользователя, его акциях, текущую дату, акции на рынке. Информация о брокере: логин, стоимость приобретенных акций на текущий момент (обновляется со сменой дня торгов), баланс, приобретенные акции. По каждой приобретенной акции отображается следующая информация: название, количество, цена на рынке на текущий момент, цена за которую была приобретена акция (если акции приобретались не одновременно, то среднюю цену покупки), прибыль. Список всех акций отображается в виде таблицы: название, цена на текущий момент, кнопка просмотра графика акции, кнопка приобретения акции. При нажатии на кнопку продажи акции или приобретения всплывает соответствующее диалоговое окно с полем ввода количества покупки/продажи акций, покупка/продажа происходит моментально по цене на текущий момент. Если брокер является администратором, то ему отображается кнопка перехода на панель администратора.

Для отображения панели администратора был создан компонент `AdminPanel`. Он состоит из таблицы: логин брокера, баланс, список акций, прибыль. В каждой строке таблица содержит информацию: логин брокера, его баланс, в столбце списка акций присутствует информация о каждой приобретенной акции: количество, название, прибыль, в последнем столбце общая прибыль со всех акций, которыми владеет брокер.

Для работы с данными были созданы файлы состояний: `storeBroker`, `storeBrokers`, `storeDate`, `storeStocks`, рассмотрим их.

`storeBroker` хранит в себе информацию о конкретном брокере. С помощью этого состояния можно изменять значение текущего брокера (например баланс, список активов), получать необходимую информацию. Для получения информации были написаны следующие геттеры:

- `getBrokerId()` возвращает `id` брокера

- `getBrokerLogin()` возвращает логин брокера
- `getBrokerActives()` возвращает активы брокера
- `getBrokerBalance()` возвращает баланс брокера

`storeBrokers` похож на `storeBroker`, но он хранит информацию о каждом брокере, имеет аналогичные геттеры, отличные лишь тем, на вход они принимают `id` брокера, чью информацию необходимо получить.

`storeDate` используется для обновления и отображения текущей даты. Для обновления даты созданы `action` и `mutatins` `updateDate(date)`. Для получения текущей даты – геттер `getCurrentDate()`, для получения текущей даты в форматированном виде – `getCurrentDateStock()`.

`storeStocks` используется для хранения всех акций на текущий момент (состояние `stocks`), а так же историю акций уже прошедших дней (состояние `stocksHistory`). Для обновления текущих цен акций созданы `action` и `mutatins` `updateStockPrice`, принимают аргумент – акцию – в которой хранится информация об акции. `addStockHistoryPrice` так же принимает данные акции, но еще и дата, и вместо замены текущей акции на новую, добавляет в `map` (ключ – дата) новое значение акции. Для получения данных написаны следующие геттеры:

- `getStockPrice` принимает `id` и возвращает цену акции на текущий момент
- `getStockName` принимает `id` и возвращает название акции
- `getBrokerStocksPrice` принимает список активов брокера, на основе их и текущих цен подсчитывает сумму цен акций активов брокера
- `getStockProfit` принимает `id` акции, цену за которую ее купили и количество. Зная текущую цену акции с соответствующим `id` и цену покупки высчитывает прибыль по акции, общая прибыль формируется из умножения прибыли по одной акции на их количество
- `getStocksPrice` возвращает массив цен акций
- `getStockHistoryPrice` принимает `id` акции и дату, возвращает цены по акции до даты, которая поступила в качестве аргумента включительно.

Для получения данных с сервера используются сокеты. Это позволяет не писать запросы а лишь обрабатывать данные которые автоматически принимаются на стороне веб приложения.

Макет интерфейса:

Вход

Рисунок 1 макет входа

Акции на 23.12.2020

Название	Цена		
AAPL	132.16	График	Купить
AMD	93.08	График	Купить
AMZN	160.25	График	Купить

Информация о счете

Логин: логин
Текущая стоимость акций: 1000
Доступные средства: 1000

Admin Panel

Купленные акции

AAPL
Количество: 10
Цена продажи: 132.16
Цена покупки: 132.52
Прибыль: -3.6

Продать

Рисунок 2 макет аккаунта

Работа программы.

Вход в аккаунт:

ВХОД

Рисунок 3 вход в аккаунт

Страница аккаунта:

Акции на 21.10.2020

Название	Цена		
AAPL	116.67	График	Купить
AMD	81.72	График	Купить
AMZN	160.625	График	Купить
CSCO	39.31	График	Купить
MSFT	213.12	График	Купить
QCOM	128.04	График	Купить
SBUX	88.38	График	Купить
TSLA	140.9	График	Купить

Информация о счете

Логин: admin
 Текущая стоимость акций: 198.39
 Доступные средства: 23537.28

Admin Panel

Купленные акции

AAPL

Количество: 1
 Цена продажи: 116.67
 Цена покупки: 132.52
 Прибыль: -15.85

Продать

AMZN

Количество: 1
 Цена продажи: 160.625
 Цена покупки: 161.69
 Прибыль: -1.06

Продать

Рисунок 4 страница аккаунта

График цен:

Акции на 12.12.2020

Название	Цена		
AAPL	Не		
AMD	Не		
AMZN	Не		
CSCO	Не		
MSFT	Не		
QCOM	Не		
SBUX	Не		
TSLA	Нет торгов	График	Купить

Информация о счете

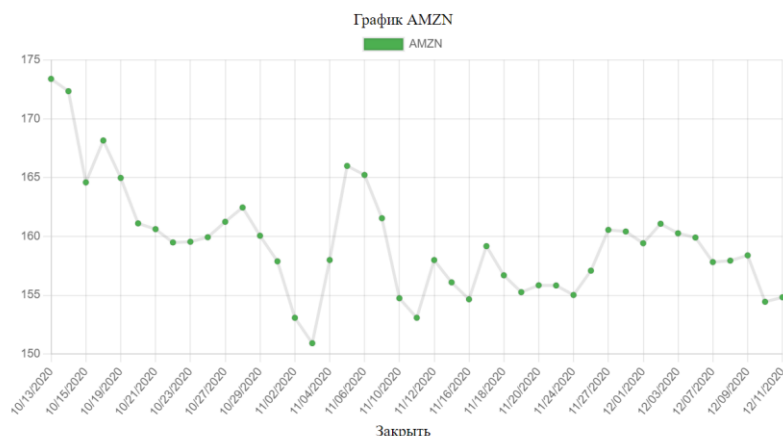


Рисунок 5 график цен

Диалоговое окно покупки:

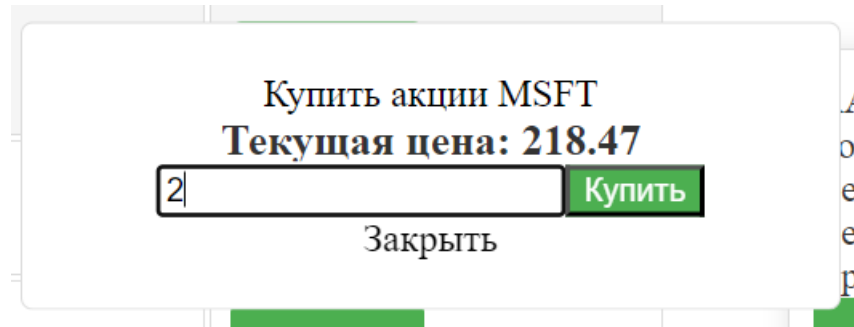


Рисунок 6 диалог покупки

Страница администратора:

Admin Panel

Участник	Баланс	Акции	Прибыль
admin	20853	1 - AAPL : 2 2 - AMD : 166 1 - AMZN : 5 3 - CSCO : 12 2 - QCOM : -13 3 - SBUX : 32 3 - TSLA : -60	146
dima	2	2 - CSCO : 5	5
new	262	1 - AAPL : 12 1 - AMZN : 11 1 - MSFT : 18 1 - QCOM : 2 1 - SBUX : 5 1 - TSLA : 31	82

К торгам

Рисунок 7 панель администратора

Вывод.

В ходе работы были изучены возможности применения фреймворка Vue для разработки интерфейсов пользователя web-приложений и организации E2E-тестирования клиентской части приложения.