**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №4**

**по дисциплине «WEB-технологии»**

Тема: Модуль пользователя приложения

«Социальная сеть»

|  |  |  |
| --- | --- | --- |
| Студент гр. 0304 |  | Асташёнок М.С. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2022

## Цель работы.

Целью работы является изучение основ языка TypeScript и особенностей применения фреймворка Angular для разработки web-приложений, ведения журналов ошибок, реализации взаимодействия приложений с использованием web-сокетов, организации модульного тестирования web-приложений с использованием Jest.

## Задание.

Необходимо создать web-приложение, обеспечивающее использование пользователем социальной сети. Пользователь может зарегистрироваться в социальной сети. Может добавить или удалить свою фотографию, может управлять своими друзьями в социальной сети, может добавить сообщение (новость) на свою страницу, может просматривать список новостей своих друзей.

Основные требования:

1. Приложение получает исходные данные из модуля администрирования приложения «Социальная сеть» в виде JSON-файла и работает одновременно с модулем администрирования приложения «Социальная сеть».

2. В качестве сервера используется Node.JS с модулем express.

3. Предусмотрены:

– HTML-страница для регистрации пользователя;

– HTML-страница для просмотра ленты новостей (пользователя и его друзей);

– HTML-страница для добавления сообщения (новости).

4. Если пользователь является администратором, то у него есть возможность перехода в модуль администрирования приложения «Социальная сеть».

5. Переписка и страница новостей обновляются сразу после появления сообщений и новостей от пользователей без необходимости обновлять страницу целиком.

6. Разработаны тесты для серверной части web-приложения с использованием Jest.

7. Все элементы управления реализованы с использованием компонентов Angular. Взаимодействие между компонентами реализовано с использованием сервисов Angular.

8. Для реализации эффектов на HTML-страницах используются директивы Angular.

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

Преимуществом будет использование звукового сопровождения событий: получений сообщений, появление новостей.

Преимуществом будет использование компонентов Angular Material (https://material.angular.io/).

## Основные теоретические положения.

Angular – фреймворк для разработки клиентских частей web-приложений, основанный на языке TypeScript. Фреймворк построен на использовании компонентного подхода, где каждый компонент может отображаться пользователю в соответствии с его индивидуальным шаблоном.

Фреймворк в том числе предлагает инструменты для управления из командной строки и организации тестирования.

CORS (Cross-Origin Resource Sharing) – это система, позволяющая отвечать на запросы из другого домена, отличного от домена происхождения запрашиваемого ресурса. Пример запроса CORS: приложение запущено на http://localhost:8080, а запросы отправляются на другой порт – http://localhost:3000.

Журналы ошибок позволяют контролировать появление ошибок как на этапе разработки, так и при работе пользователей. В качестве журналов ошибок предлагается использовать Rollbar (https://rollbar.com/) или Sentry (https://sentry.io/).

WebSocket – протокол связи, который может передавать и принимать одновременно сообщения поверх TCP-соединения, предназначен для обмена сообщениями между браузером и web-сервером, но может быть использован для любого клиентского или серверного приложения. Для создания web-сокетов предлагается использовать модуль Socket.IO (https://socket.io/).

Jest (https://jestjs.io/ru/) – это фреймворк для написания тестов серверной части web-приложений.

## Выполнение работы.

**1. Верстка макетов страниц в *Figma*.**

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

Страница новостей содержит список новостей от друзей пользователя и самого пользователя, отсортированный по новизне. Можно перейти на автора поста и просмотреть его профиль.

Страница друзей содержит список друзей пользователя. Для друзей есть возможность удаления из друзей. Также присутствует строка для добавления друга по идентификатору

Страница профиля содержит информацию об аккаунте, такую как имя, email, дату рождения, роль, статус и идентификатор.

Страница авторизации содержит форму авторизации, в которой пользователь вводит имя и дату рождения (только при регистрации), email и пароль.

**2. Разработка интерфейса модуля.**

С помощью фреймворка Angular версии 14 был разработан интерфейс пользовательского модуля. Были разработаны компоненты страниц логина и регистрации, авторизации, новостей, друзей, профиля, а также компонент отдельного поста (новости).

Для получения данных от сервера используются сервисы: сервис авторизации для авторизации пользователя и сервис данных для получения различных данных с сервера (новости, пользователи, друзья и т. д.).

Для использования приложения пользователь должен войти либо зарегистрироваться. После авторизации в локальном хранилище страницы сохраняется идентификатор пользователя, который используется для персонализации приложения.

Внешний вид интерфейса см. на рис. 1-4.

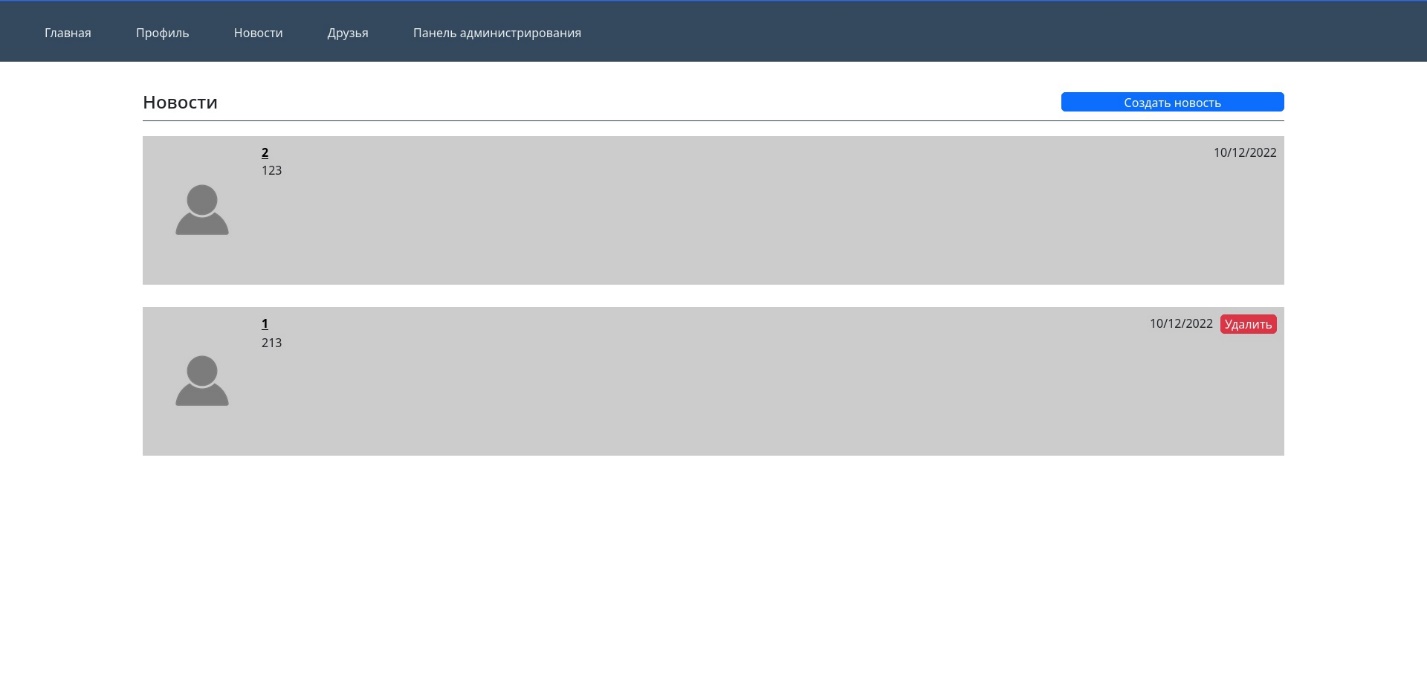


Рисунок 1. Внешний вид страницы новостей

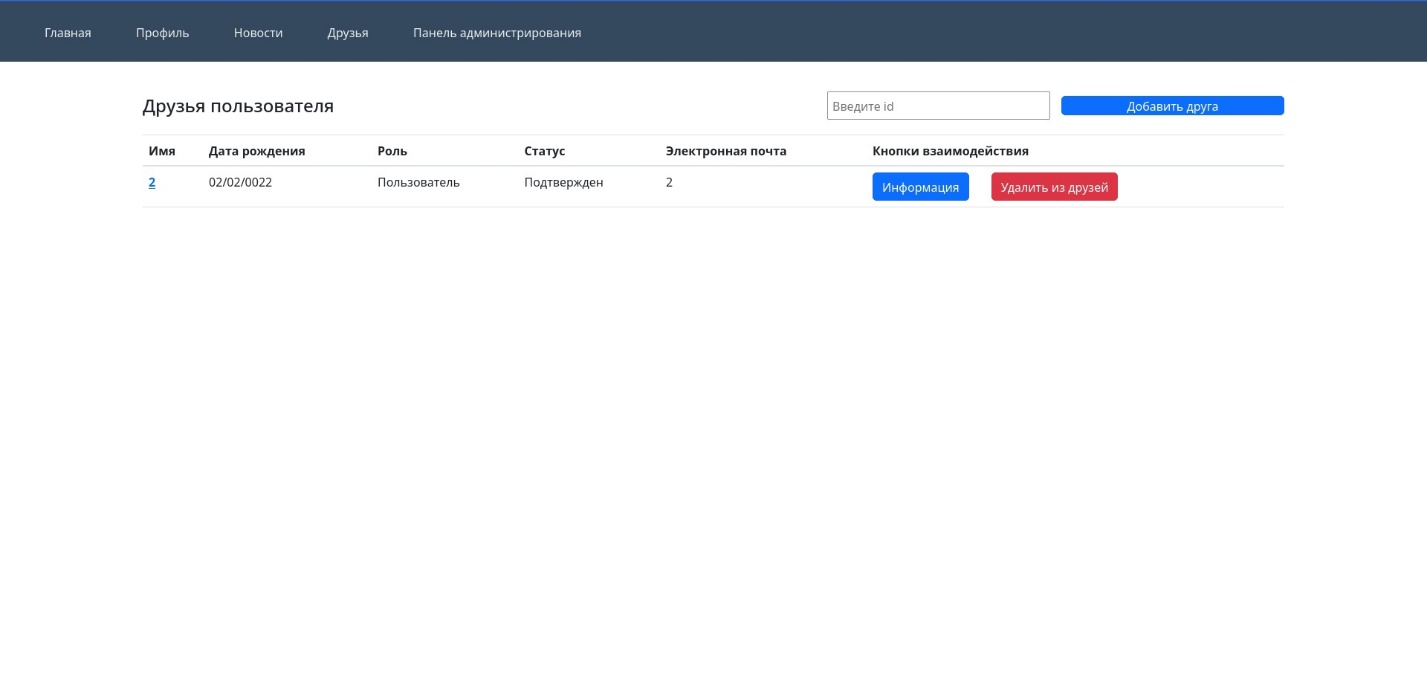


Рисунок 2. Внешний вид страницы друзей

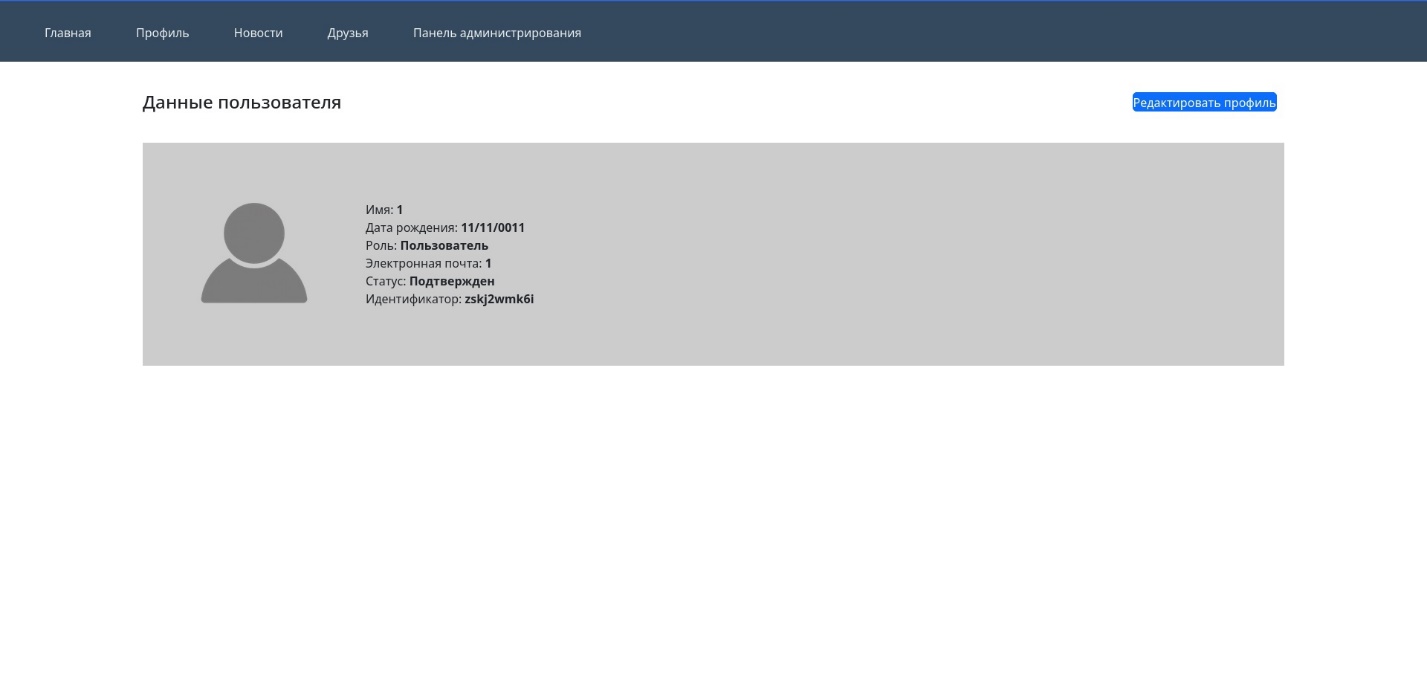


Рисунок 3. Внешний вид страницы профиля

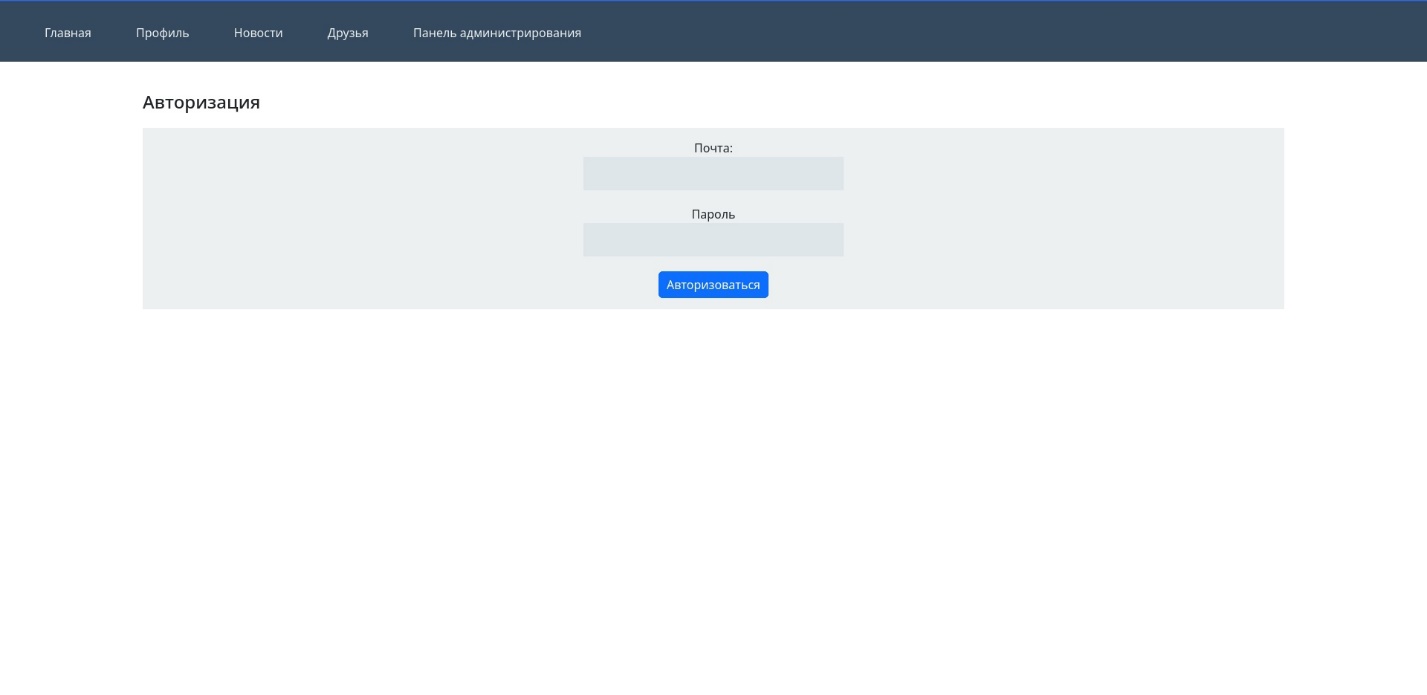


Рисунок 4. Внешний вид страницы авторизации (логина)

**3. Разработка серверной части.**

Серверная часть была разработана с помощью Node.js с модулем express, а также библиотеки Socket.io. На сервере настроен CORS для возможности взаимодействия сервера клиента (Angular) и данного сервера.

При авторизации клиент посылает данные на сервер, после чего сервер проверяет данные с помощью базы пользователей, которую он получает с сервера панели администратора. Если данные верны, то сервер отсылает ответ, содержащий информацию о пользователе (кроме пароля).

Получение и обновление данных происходит путем запроса к сокет-серверу. На клиенте создается сокет, который подключается к сокет-серверу, и все взаимодействие происходит путем общения с помощью сокетов. Сокет-сервер обрабатывает входящие соединения и сохраняет новое подключение в список подключенных клиентов. При подключении к сокет-серверу и при авторизации клиент высылает идентификатор пользователя, и, при последующих запросах к серверу, используется пользователь, которому принадлежит данный идентификатор. При обновлении данных (добавлении поста, изменении статуса дружбы) новые данные отсылаются всем подключенным клиентам.

С помощью сокетов клиент получает список новостей для пользователя, список своих друзей и данные профиля какого-либо пользователя. Также клиент имеет возможность добавления нового поста, добавления или удаления друга, все данные действия осуществляются путем запросов к сокет-серверу, причем при изменении данных обновления приходят всем пользователем сети без необходимости перезагрузки страницы.

**4. Написание тестов серверной части.**

Для серверной части были написаны модульные тесты с использованием Jest. Тестировалась API регистрации и входа в аккаунт. Проверялось, что с валидными данными пользователь успешно может войти в аккаунт. Также проверялось, что с уже занятым email’ом пользователь не сможет создать аккаунт. Приложение проходит тесты успешно.

**Выводы.**

В ходе работы было разработано приложение «Социальная сеть» с помощью Angular и Node.js. При написании клиентской части использовались возможности Angular, такие как компоненты, модулю, сервисы и т. д. Клиент-серверное взаимодействие осуществляется с использованием веб-сокетов, что позволяет обновлять данные без перезагрузки страницы.