**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. 1381 |  | Возмитель В. Е. |
| Преподаватель |  | Беляев С. А. |

Санкт-Петербург

2023

**Цель.**

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.**

Для достижения поставленной цели требуется решить следующие задачи:

– разработка интерфейса web-приложения;

– создание web-сервера на основе express, настройка маршрутов, подготовка и обработка REST-запросов с учетом CORS (серверная часть);

– создание каркаса web-приложения с использованием Angular;

– определение перечня компонентов и сервисов web-приложения;

– создание шаблонов компонентов;

– регистрация и подключение в web-приложение журнала ошибок.

– создание web-сокета для отправки сообщений всем клиентам;

– создание и запуск Jest тестов для приложения.

**Основные теоретические сведения.**

Angular – фреймворк для разработки клиентских частей web-приложений, основанных 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.**

Для выполнения поставленных требований было разработано web-приложение с использованием Node.JS и фреймворка Express в качестве сервера. Приложение получает исходные данные из модуля администрирования приложения «Социальная сеть» в виде JSON-файла и работает одновременно с этим модулем.

В составе приложения были предусмотрены следующие функциональные страницы:

HTML-страница для регистрации пользователя, которая позволяет пользователю зарегистрироваться в приложении.

HTML-страница для просмотра ленты новостей, где пользователь может просмотреть новости своих друзей, а также собственные новости.

HTML-страница для добавления новости, которая позволяет пользователю добавить свою новость.

При наличии прав администратора у пользователя есть возможность перейти в модуль администрирования приложения «Социальная сеть».

Для обновления переписки и страницы новостей без необходимости обновлять страницу целиком, была реализована механика обновления данных сразу после появления новых сообщений и новостей от пользователей.

Также были разработаны тесты для серверной части приложения с использованием Jest, для обеспечения надежности и корректности работы приложения.

Все элементы управления в приложении реализованы с использованием компонентов Angular, а взаимодействие между компонентами осуществляется с использованием сервисов Angular. Для реализации эффектов на HTML-страницах были использованы директивы Angular.

Макет интерфейса для всех страниц приложения был разработан с использованием Figma, что позволило предварительно визуализировать и протестировать интерфейс перед его реализацией. Макеты приведены в приложении А.

**Вывод.**

В результате выполнения работы было достигнуто основное изучение языка TypeScript и его применение в разработке web-приложений с использованием фреймворка Angular. Были изучены особенности работы с фреймворком Angular, включая взаимодействие с сервером через web-сокеты и организацию модульного тестирования с использованием Jest.

В ходе разработки было создано функциональное web-приложение, которое позволяет пользователям регистрироваться, просматривать новости своих друзей, добавлять свои новости и взаимодействовать с модулем администрирования. Были реализованы механизмы обновления данных без необходимости обновления страницы целиком, что повышает удобство использования приложения.

Также были разработаны тесты для серверной части приложения с использованием Jest, что обеспечивает надежность и корректность работы приложения.

Вся пользовательская инфраструктура приложения была построена с использованием компонентов Angular, а взаимодействие между компонентами осуществляется через сервисы Angular. Для создания эффектов на HTML-страницах были использованы директивы Angular.

Макет интерфейса для всех страниц приложения был разработан с использованием Figma, что позволило предварительно визуализировать и протестировать интерфейс перед его реализацией. Все поставленные цели работы были успешно достигнуты, что сделает данное приложение полезным и эффективным для пользователя.

## Приложения А. *Figma* макеты и ссылка.

Ссылка на Figma проект:

<https://www.figma.com/file/wblHfls7YujTsYznbJA2Yf/lab4?type=design&node-id=0%3A1&mode=design&t=x35oEGi2TCI5tJGt-1>

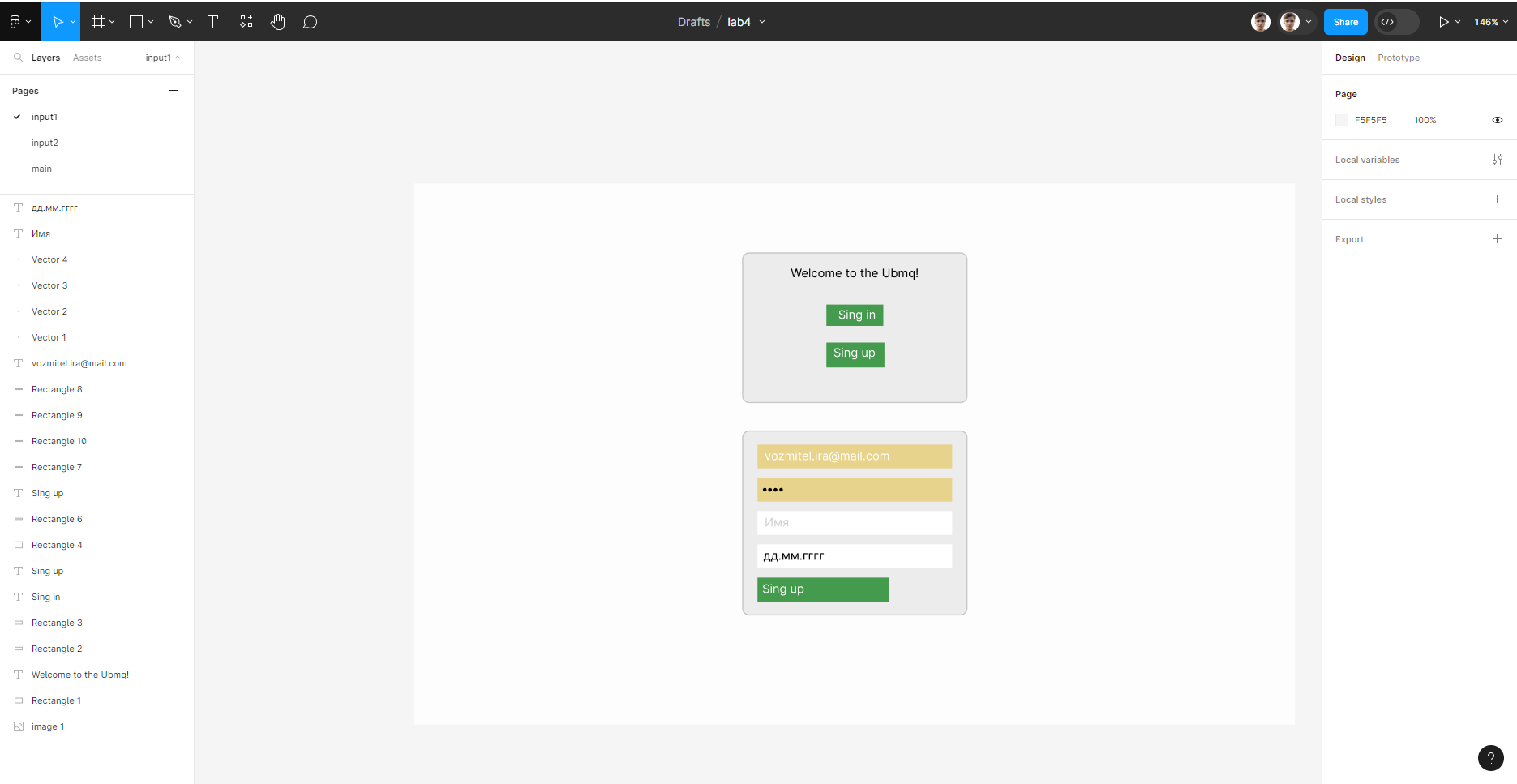


Рис №1 – Макет начальной страницы.

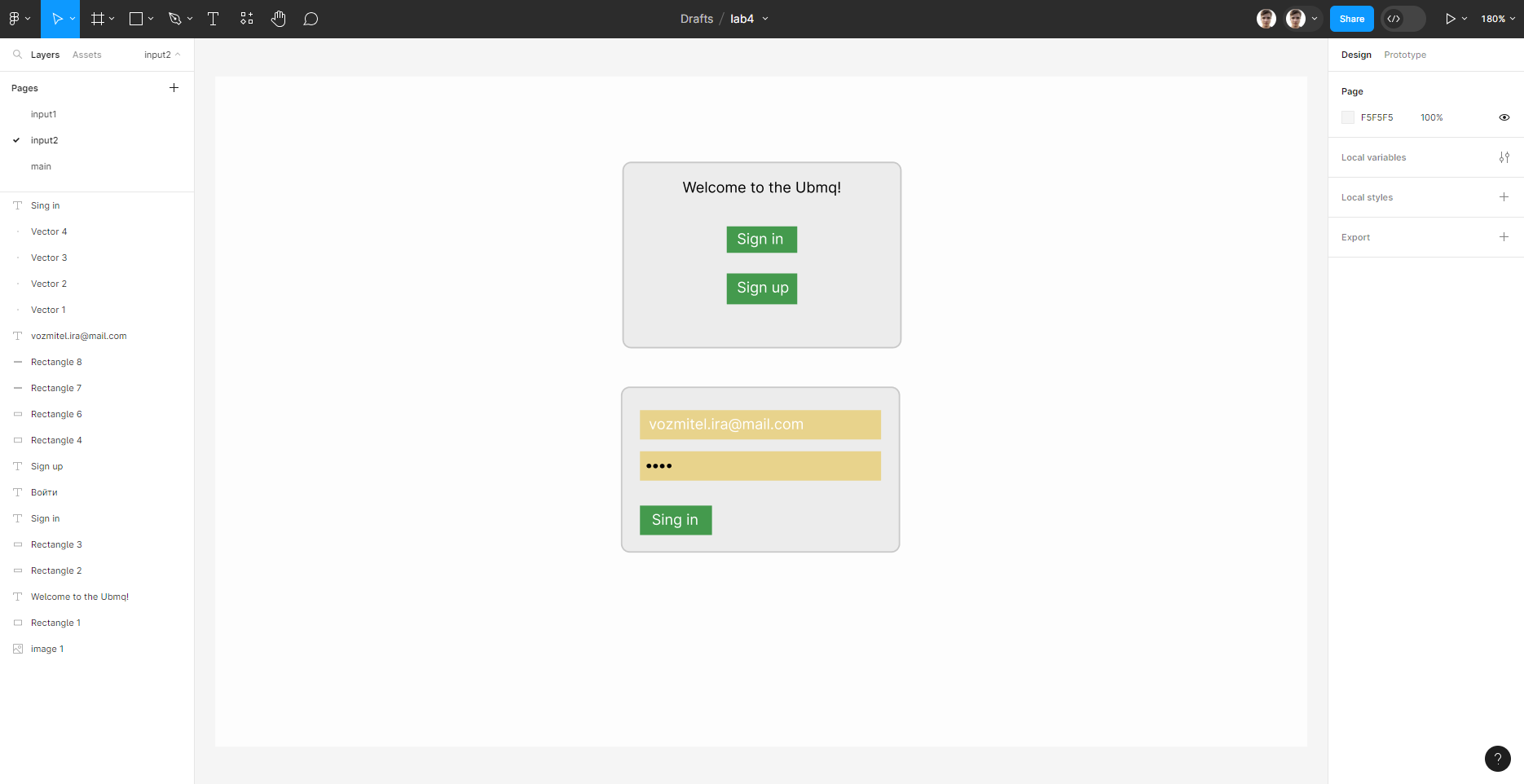


Рис №2 – 2 макет начальной страницы.

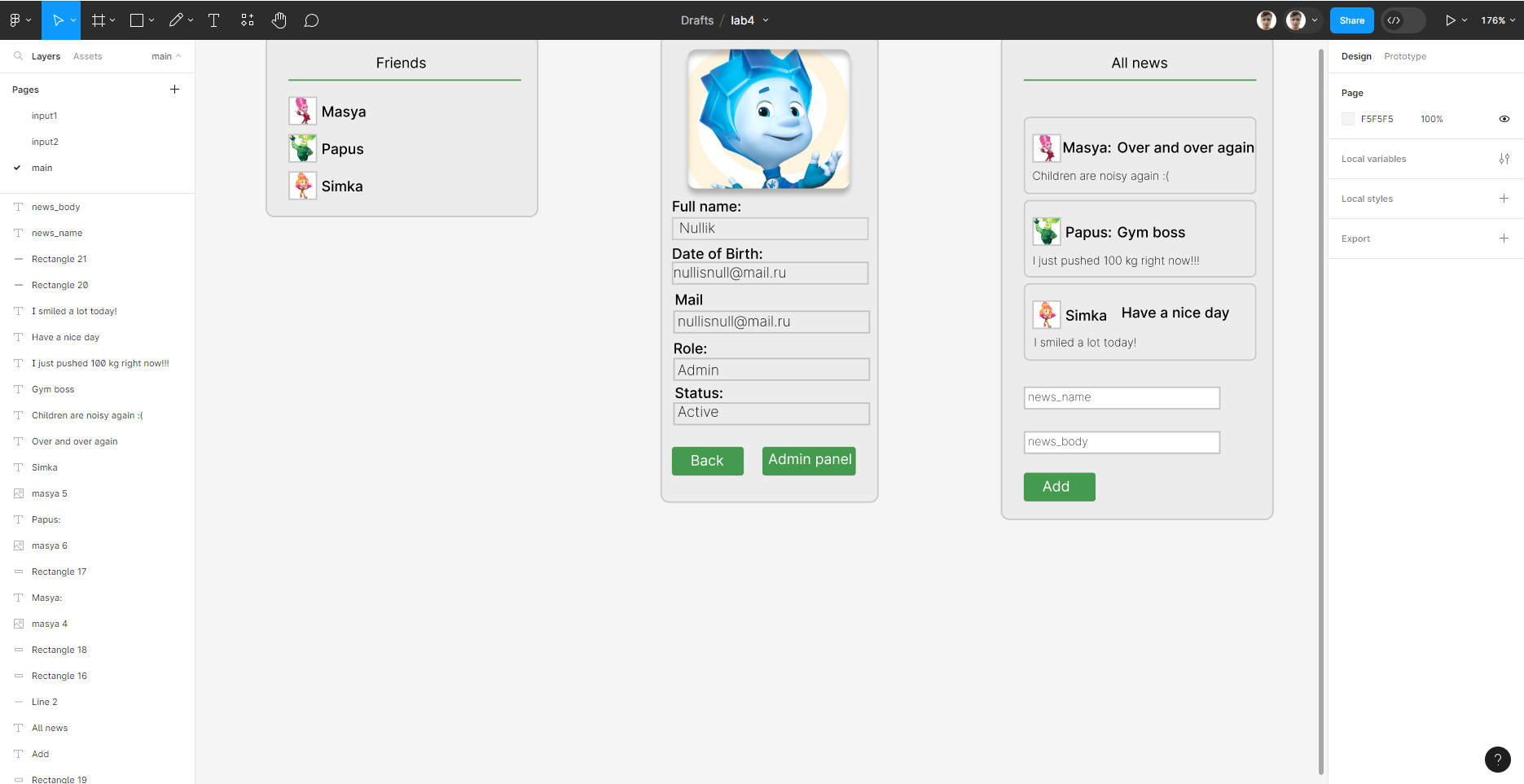


Рис №3 – макет основной страницы.