**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Web-технологии»**

Тема: Модуль администрирования приложения «Социальная сеть»

|  |  |  |
| --- | --- | --- |
| Студент гр. 2300 |  | Богатов И.А. |
| Преподаватель |  | Беляев С.А. |

Санкт-Петербург

2025

## Цель работы

Изучение возможностей применения компилятора Babel, препроцессора LESS, препроцессора SASS/SCSS, инструмента выполнения повторяющихся задач GULP, освоение инструмента сборки Webpack, регистрация разработанных модулей, формирования навыков построение структурированных web-приложений, освоение особенностей стандартных библиотек.

## Задание

Необходимо создать web-приложение, обеспечивающее администрировние социальной сети: можно управлять участниками, их ролями, сообщениями. Основные требования следующие:

1. Перечень участников, их друзей, сообщений и т. п. хранится в JSON файлах на сервере.

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

3. Разработка ведется с использованием стандарта не ниже ECMAScript2015, используются модули ES6.

4. Стили описываются с использованием LESS или SASS, при этом используются ключевые методы LESS/SASS (переменные, вложенные блоки, миксины, операторы и т. п.).

5. Клиентская часть разрабатывается с использованием jQuery (работа с DOM, AJAX-запросы), используются компоненты jQuery UI или Bootstrap.

6. Предусмотрена HTML-страница для списка пользователей (ФИО, дата

рождения, email, фотография, роль, статус). Предусмотрена возможность редактировать данные пользователя, изменять роль (администратор, пользователь), из-

менять статус (неподтвержденный пользователь, активный, заблокированный).

7. Предусмотрены:

– HTML-страница для списка друзей пользователя;

– HTML-страница для списка новостей друзей пользователей.

8. Взаимодействие браузера с сервером осуществляется по протоколу

HTTPS, все изменения сохраняются в соответствующие json-файлы на сервере.

9. Сборка клиентской части (преобразования less или sass, pug или ejs, babel, минификация) осуществляется с использованием двух инструментов: GULP и Webpack. Это должны быть две отдельные сборки в разные папки.

10. Регистрация и удаление разработанных модулей в npm.

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

## Выполнение работы

1. Изначально было разработано API, доступное по /api префиксу, при помощи этих ручек производилась остальная работа.

2. Для валидации данных и работы с хранилищем использовались модули Joi и neDB

3. Для шаблонизации страничек также как и в прошлой работе использовался ejs

4. Был добавлен функционал просмотра всех новостей и новостей только тех пользователей, на которых подписан выбранный пользователь, однако добавление постов и механику подписки в UI требуется реализовать в 4 лабораторной работе. Для демонстрации отправлялись запросы на API через Postman для создания постов.

5. Сборка стилей и JS файлов реализована как при помощи Webpack, так и при помощи Gulp. Использовался формат SCSS для описания стилей. Многие компоненты брались из модуля Bootstrap.

6. Основные странички можно увидеть на рис.1, рис.2, рис.3

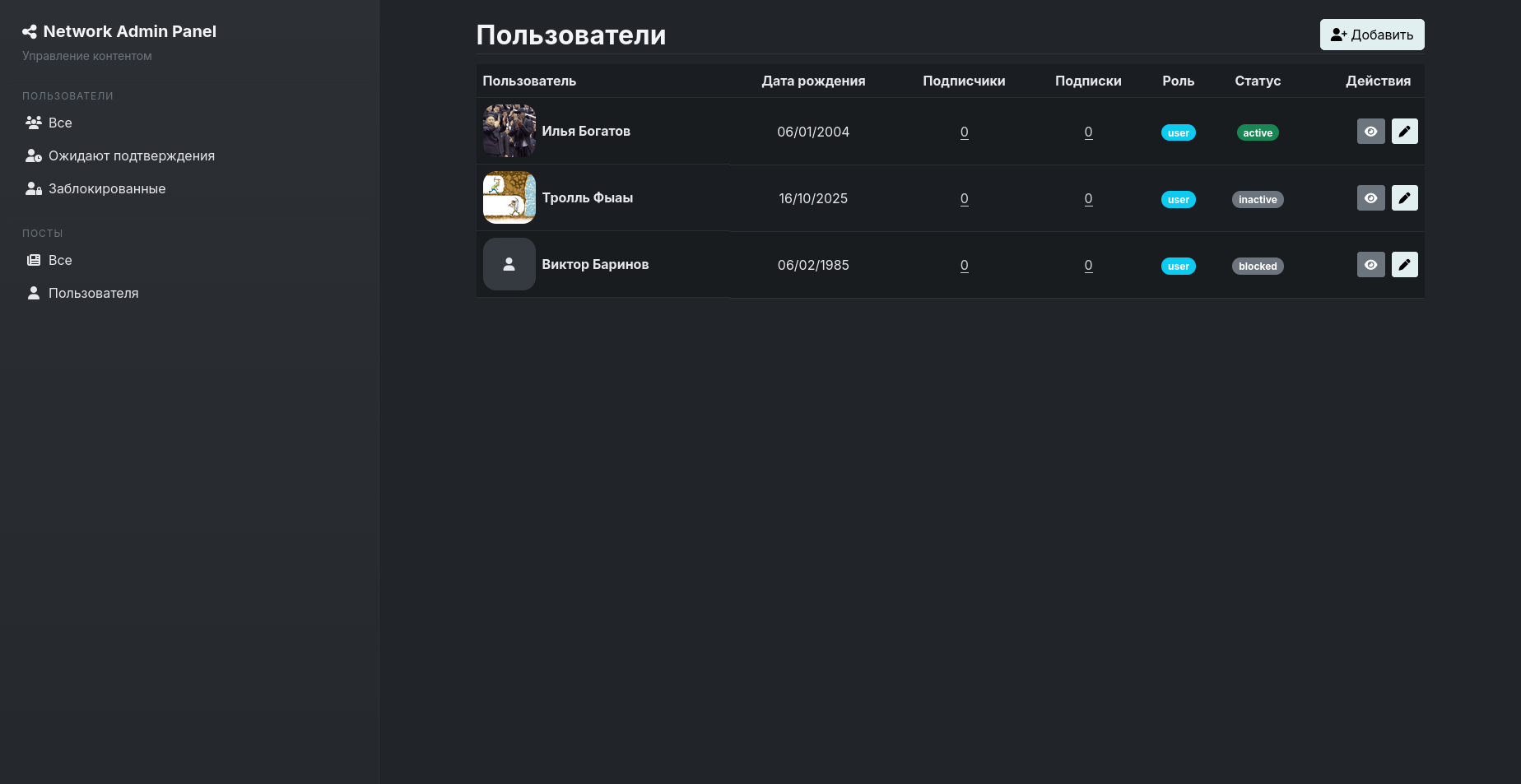


Рисунок 1 – Главная страница со всеми пользователями

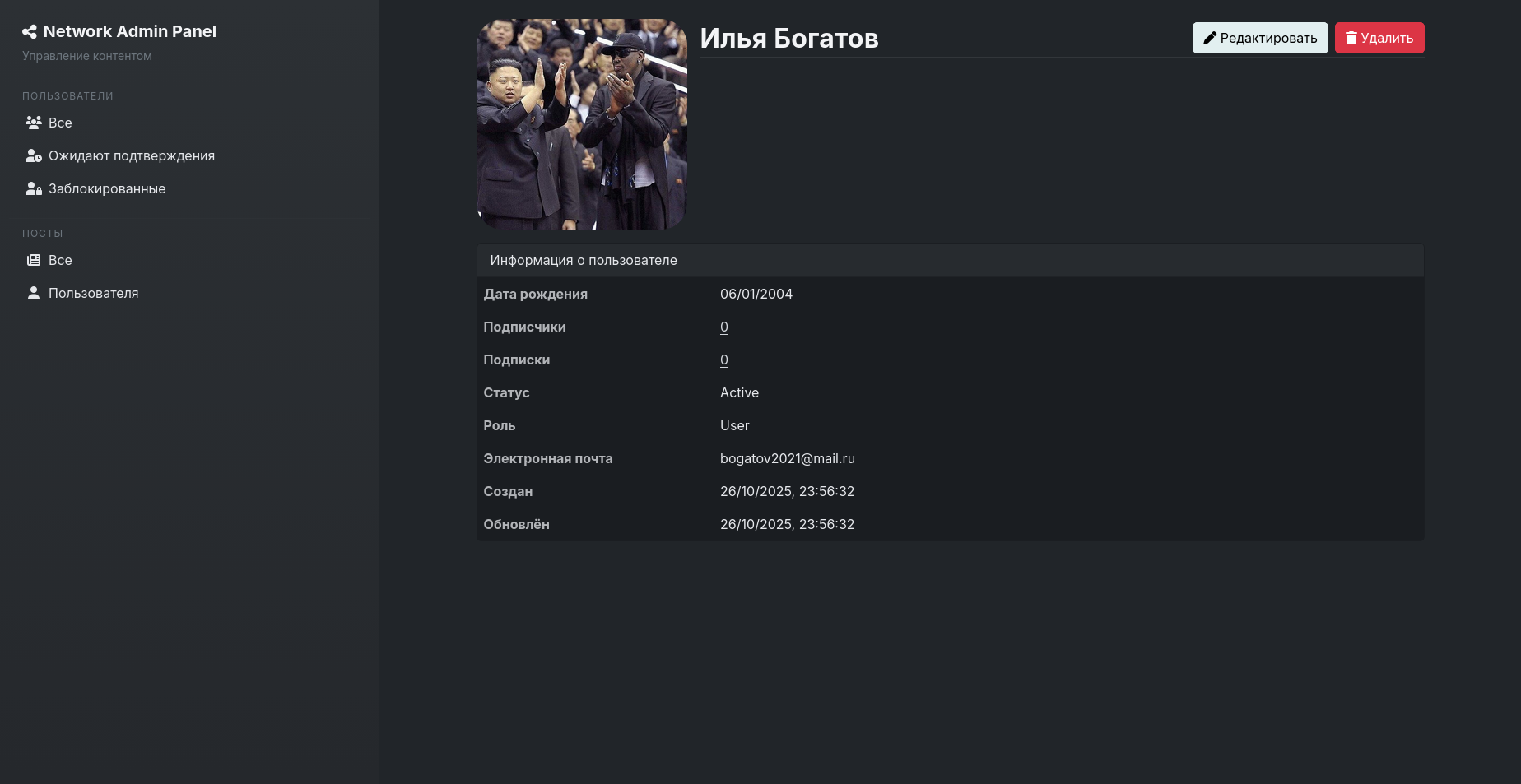


Рисунок 2 - Страницы пользователя

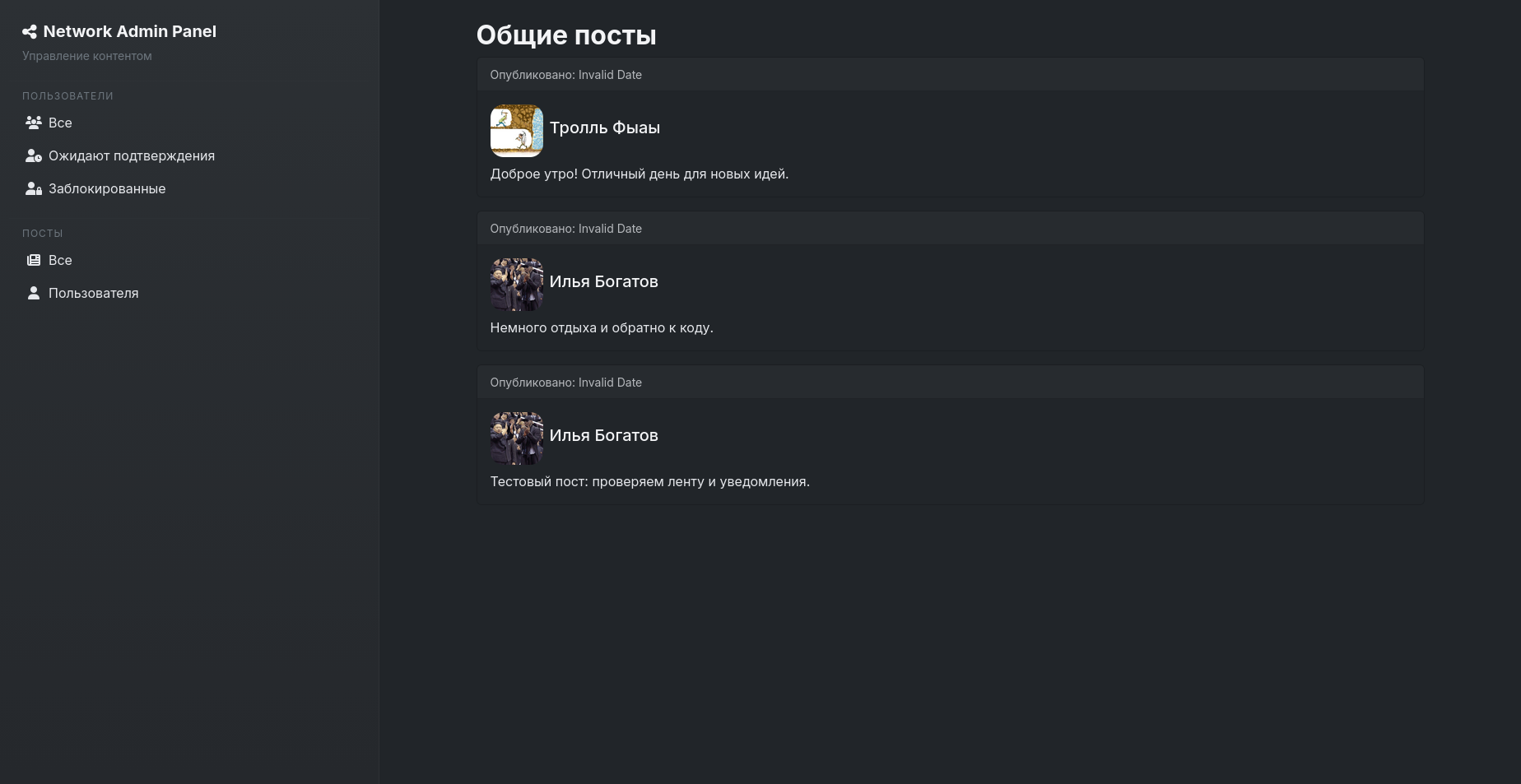


Рисунок 3 – Страница всех постов

## Выводы

Реализовано клиент-серверное приложения для работы с админ панелью социалной сети, отображающее динамические html страницы, которые генерируются на сервере при помощи языка для шаблонизации ejs.

Для насыщения пользовательского интерфейса использованы таблицы стилей CSS и иконки fontawesome. Причем CSS генерируется из сборки при помощи Webpack или Gulp из SCSS файла.