

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. 1303

Коренев Д.А.

Преподаватель

Беляев С.А.

Санкт-Петербург

2023

Цель работы.

Изучение возможностей применения компилятора Babel, библиотеки jQuery, 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 администрирование 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), изменять 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.

Логика хранения данных:

В директории data были созданы javascript классы и другие директории для хранения и управления данными на сервере. Javascript классы:

- FriendsStorage.js – управляет списком друзей пользователей, имеет доступ к json файлам, хранящие списки друзей
- PostsStorage.js – управляет постами пользователей, может добавлять посты в json хранилище, удалять, возвращать необходимые
- UsersStorage.js – управляет пользователями, предоставляет инструменты для проверки корректности пароля пользователя, изменения данных пользователя.

Логика сервера:

Логика сервера настроена в файле index.js. При входе в аккаунт программа проверяет корректность данных и при предоставлении доступа, генерирует и сохраняет токен, чтобы пользователь мог пользоваться сервисом от своего имени. При остальных запросах, используя токен, сохраненный в cookie, делается проверка доступа пользователя: если доступ предоставлен, в систему вносятся изменения, требующие пользователем-администратором, в

противном случае в доступе отказывают. Для изменения данных используются экземпляры выше описанных классов (FriendsStorage.js, PostsStorage.js, UsersStorage.js).

Внешний вид и отображение:

Для создания разметки страницы использовался pug вместе с jQuery UI , для задания стилей - препроцессор SASS, для работы с данными страницы – jQuery.

На странице входа в систему доступно два поля ввода данных, при корректно введенных данных пользователь перенаправляется на страницу администратора (если были введены данные администратора) или отображается окно с информацией в отказе доступа (если у пользователя недостаточно прав), если данные введены некорректны – поля подсвечиваются красным цветом. Скриншот страницы входа представлен на рисунке 1.

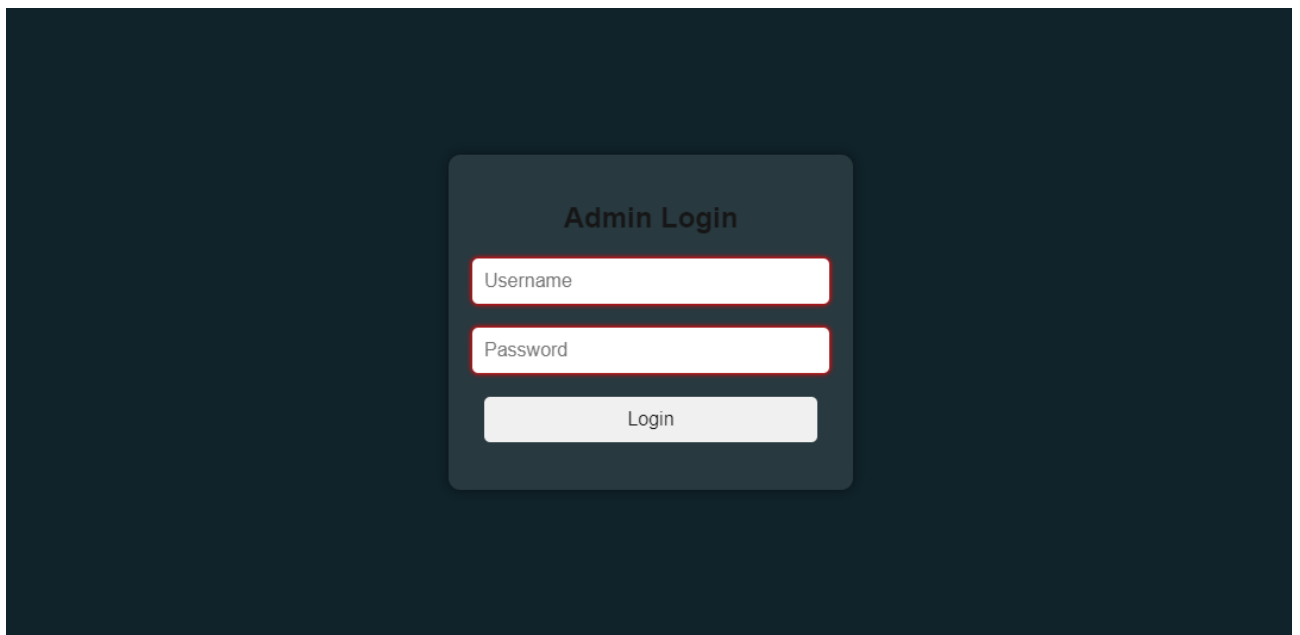


Рисунок 1 страница входа

На странице администратора виден список всех пользователей (в том числе и администратора). Скриншот страницы администратора представлен на рисунке 2.

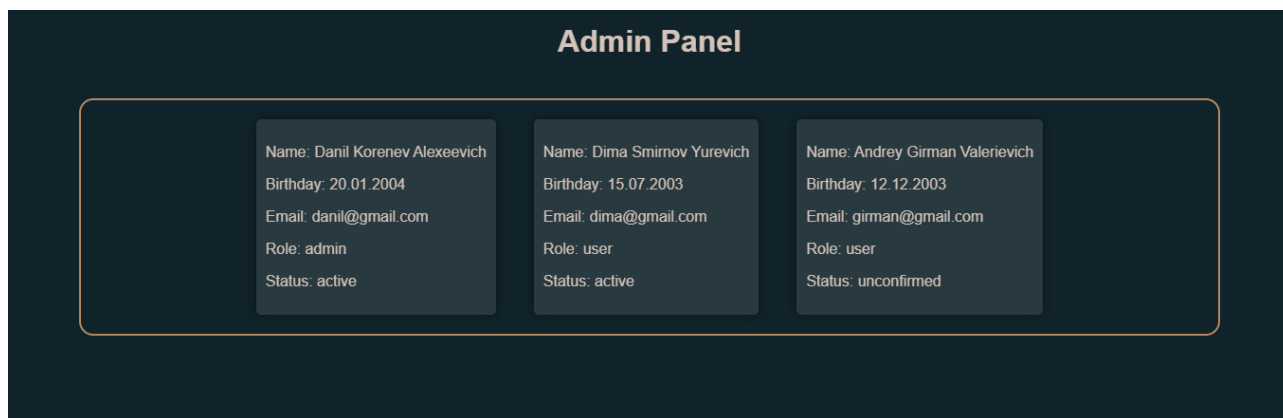


Рисунок 2 страница администрирования

При нажатии на пользователя администратор будет перенаправлен на страницу пользователя, на которой отображена краткая информация о пользователе, посты пользователя, и 5 кнопок: список друзей, список постов друзей пользователя (новости), изменение роли пользователя, изменение статуса пользователя, изменение данных пользователя. Страница пользователя представлена на рисунке 3.

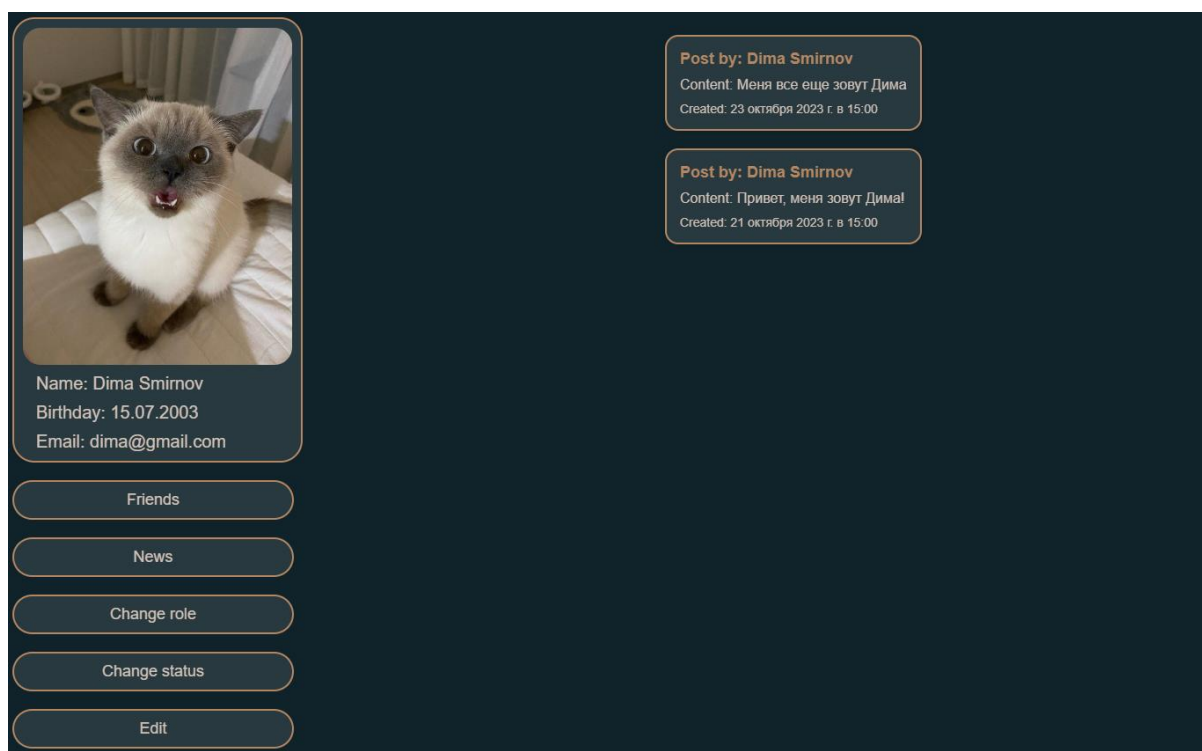


Рисунок 3 страница пользователя

На рисунке 4 представлена страница новостей пользователя danil@gmail.com, т.е. посты друзей пользователя.

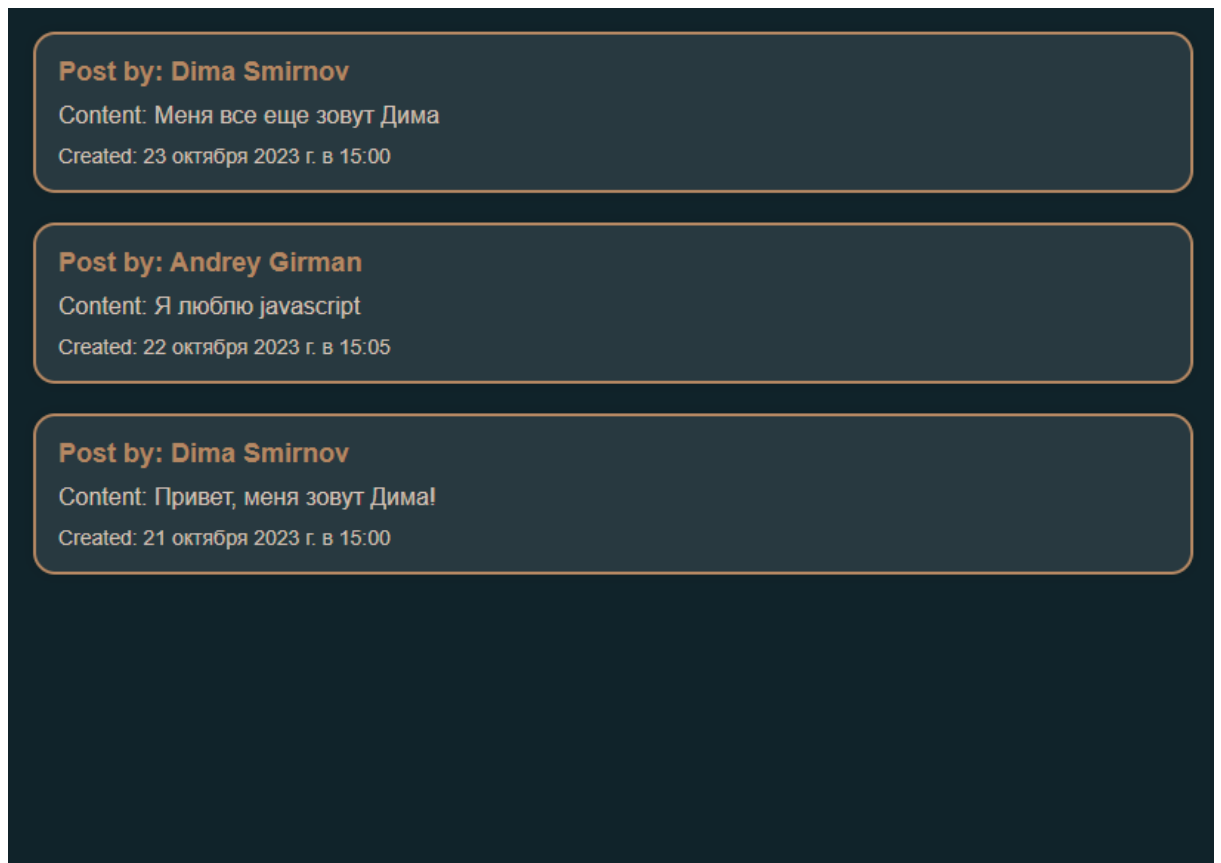


Рисунок 4 Страница новостей
На рисунке 5 представлено окно изменения роли пользователя.

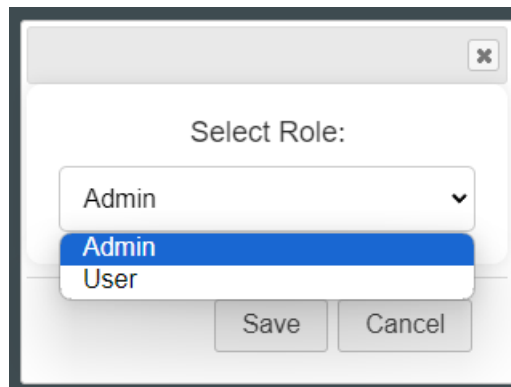


Рисунок 5 окно изменения роли
На рисунке 6 представлено окно изменения статуса пользователя.

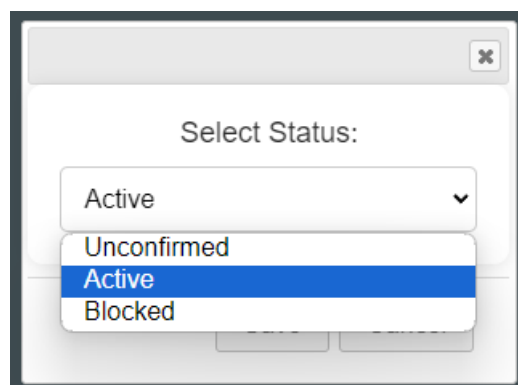


Рисунок 6 окно изменения статуса

На рисунке 7 представлено окно изменения данных пользователя.

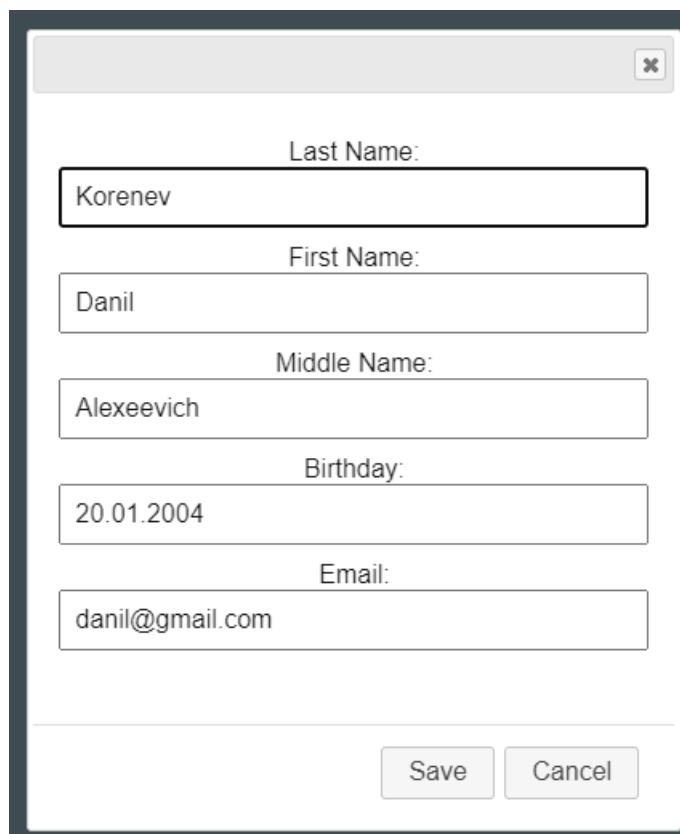


Diagram illustrating the user profile editing window (Figure 7). The window contains the following fields and values:

Field	Value
Last Name:	Korenev
First Name:	Danil
Middle Name:	Alexeevich
Birthday:	20.01.2004
Email:	danil@gmail.com

Buttons: Save, Cancel

Рисунок 7 окно редактирования пользователя

Компиляция и сборка:

Созданы файлы *webpackHTML.js*, *webpackJS.js*, *webpackSASS.js* для сборки Webpack и *gulpfile.js* для сборки Gulp. Каждая из сборок минимизирует js файлы, компилирует sass файлы в css файлы, преобразует pug файлы в html разметку.

Макет интерфейса:

При создании макета использовалась Figma. На рисунке 8 представлена страница входа.

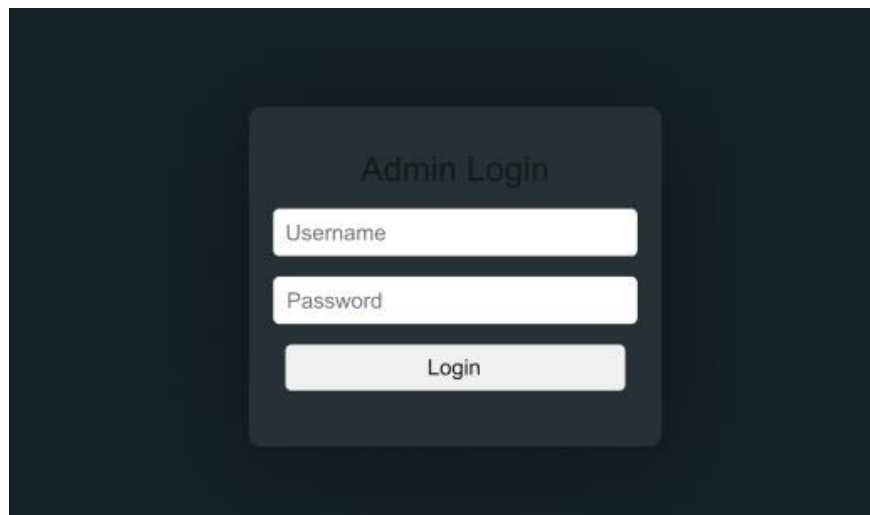


Рисунок 8 Макет страницы входа
 На рисунке 9 представлена страница панели администратора.



Рисунок 9 макет страницы панели админстратора
 На рисунке 10 представлена страница пользователя.

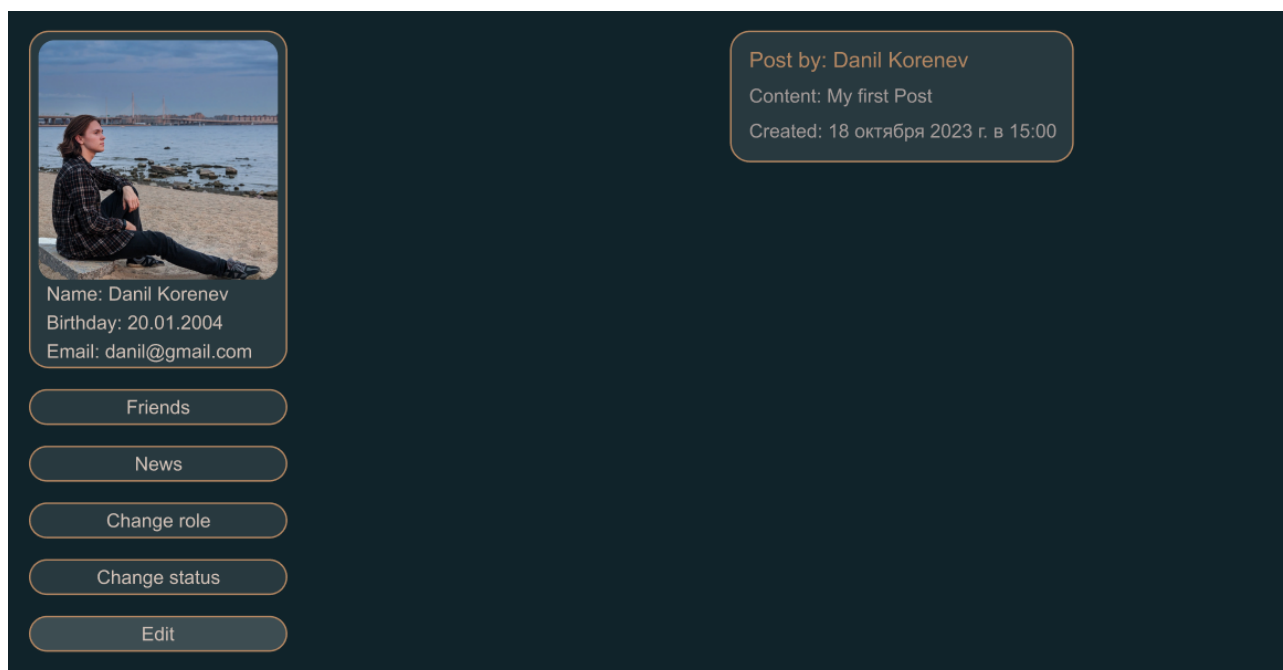


Рисунок 10 Макет страницы пользователя

Вывод.

Изучены возможности применения компилятора Babel, библиотеки jQuery, препроцессора SASS/SCSS, освоены инструменты выполнения повторяющихся задач GULP, освоены инструменты сборки Webpack, регистрация разработанных модулей.