

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. 1304

Шаврин А.П.

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 построения структурированных web-приложений, освоение особенностей стандартных библиотек.

Задание.

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/>).

Преимуществом будет, если предусмотрена работа:

- с фотографиями пользователя: список фотографий и возможность блокирования и активации фотографий администратором;
- новостями: возможность блокирования и активации новостей

Выполнение работы.

1. Созданы макеты страниц в Figma:



Рис 1. Страница всех пользователей и друзей пользователя

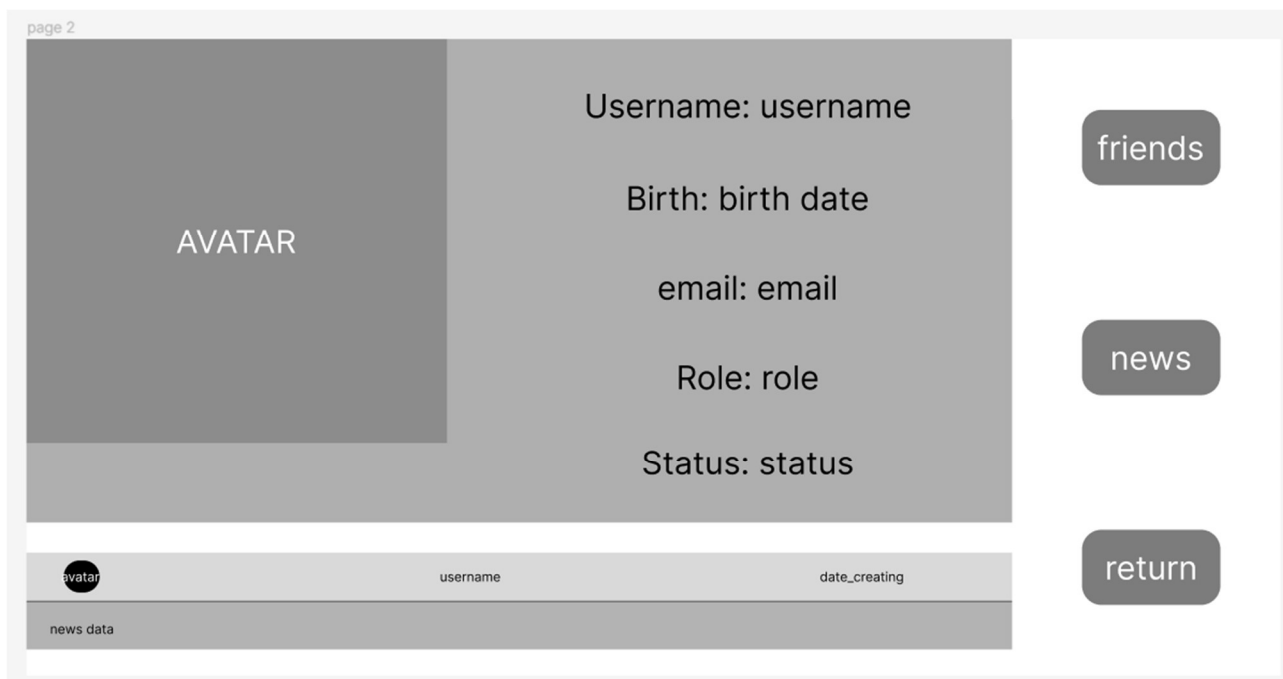


Рис 2. Страница пользователя.



Рис 3. Страница новостей.

2. Установлены все необходимые модули

```

"dependencies": {
  "@babel/preset-env": "^7.23.2",
  "babel": "^6.23.0",
  "babel-core": "^6.26.3",
  "babel-loader": "^9.1.3",
  "css-loader": "^6.8.1",
  "ejs": "^3.1.9",
  "express": "^4.18.2",
  "fs": "^0.0.1-security",
  "gulp": "^4.0.2",
  "gulp-babel": "^8.0.0",
  "gulp-clean-css": "^4.3.0",
  "gulp-less": "^5.0.0",
  "gulp-uglify": "^3.0.2",
  "https": "^1.0.0",
  "jquery": "^3.7.1",
  "jquery-ui": "^1.13.2",
  "less": "^4.2.0",
  "less-loader": "^11.1.3",
  "style-loader": "^3.3.3",
  "webpack": "^5.89.0",
  "webpack-cli": "^5.1.4"
},
"devDependencies": {
  "@babel/core": "^7.23.2",
  "@babel/polyfill": "^7.12.1",
  "@babel/preset-env": "^7.23.2"
}

```

Рис 4. Установленные модули

3. Настроена работа сервера на https протокол, используя модуль https

```

//Настройка https сервера
const https_server = https.createServer({
  cert: fs.readFileSync('public/ssl_certs/open_key.csr', 'utf8'),
  key: fs.readFileSync('public/ssl_certs/private_key.key', 'utf8')
}, app)

```

Рис 5. Htps на сервере

4. В router описаны все необходимые запросы

```

9 > router.get("/", (req, res) => { ...
11 })
12
13 > router.get("/admin_module/", (req, res) => { ...
15 })
16
17 > router.get("/admin_module/users", (req, res) => { ...
19 })
20
21 > router.get("/admin_module/users/:id([0-9]{1,})", (req, res) => { ...
35 })
36
37 > router.post("/admin_module/users/:id([0-9]{1,})", body_parser.json(), (req, res) => { ...
54 })
55
56 > router.get("/admin_module/users/:id([0-9]{1,})/friends", (req, res) => { ...
56 })
57
58 > router.get("/admin_module/users/:id([0-9]{1,})/news", (req, res) => { ...
85 })

```

Рис 6. Router.js

5. Для страниц пользователя и друзей написан скрипт на jquery для обработки нажатия на карту пользователя и перехода на его страницу

```
1 < $(() => {  
2 <     $(".user_container").on("click", function () {  
3 <         $.get(`/admin_module/users/${this.id}`)  
4 <         window.location = `/admin_module/users/${this.id}`  
5 <     })  
6 < })
```

Рис 7. JQuery страниц пользователя и друзей

6. Для страницы пользователя написан скрипт обработки кнопок перехода на страницу новостей и друзей

```
$(".new_page_button").on("click", function () {  
    $.get(`/admin_module/users/${this.value}/${this.id}`)  
    window.location = `/admin_module/users/${this.value}/${this.id}`  
})
```

Рис 8. JQuery переход на новую страницу новостей и пользователя

7. Для страницы пользователя написан скрипт обработки кнопок перехода на страницу всех пользователей

```
$(".return_button").on("click", function () {  
    window.location = "/admin_module/users"  
})
```

Рис 9. JQuery переход на страницу всех пользователей

8. Для страницы пользователя написан скрипт обработки кнопок изменения данных пользователя (роль, статус). Используется добавление событий к динамически созданным объектам.

```
> $(document).on("click", ".change_data_button", function () { ...  
    })  
> $(document).on("click", ".submit_button", function () { ...  
    })
```

Рис 10. JQuery изменение данных

9. Настроены gulp и webpack конфигурационные файлы
10. Для каждой страницы написаны less файлы стилей
11. Разработанные модули загружены в prn



Рис 11. Загруженный модуль

12.Итоговый внешний вид приложения:

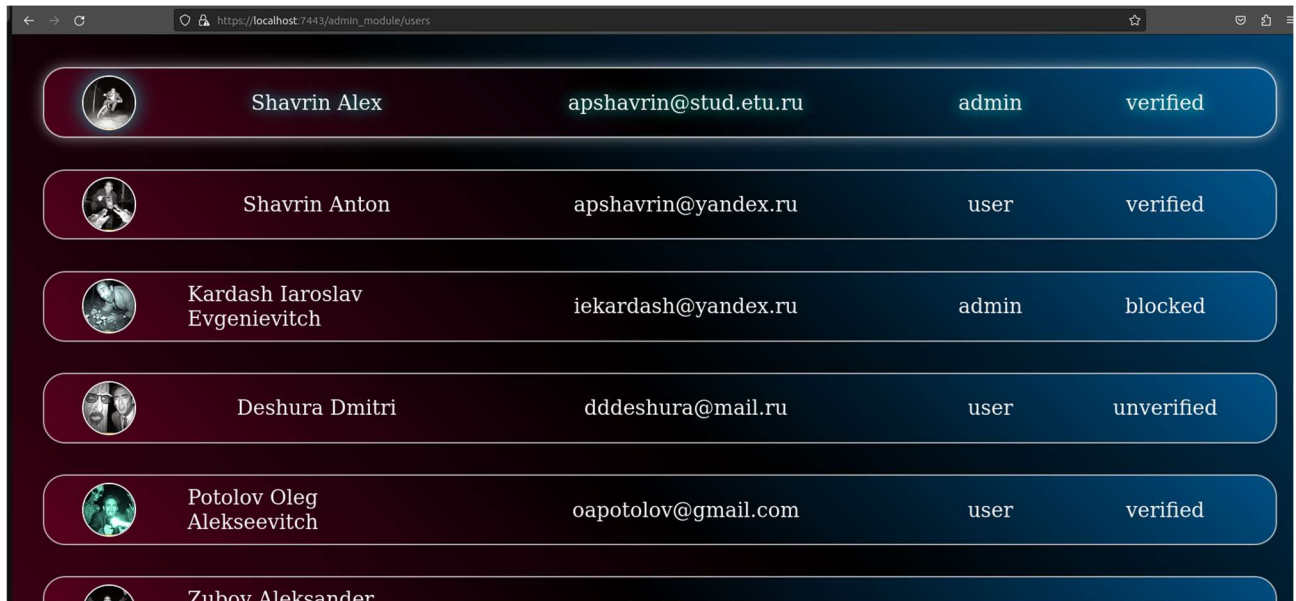


Рис 12. Главная страница пользователей или друзей

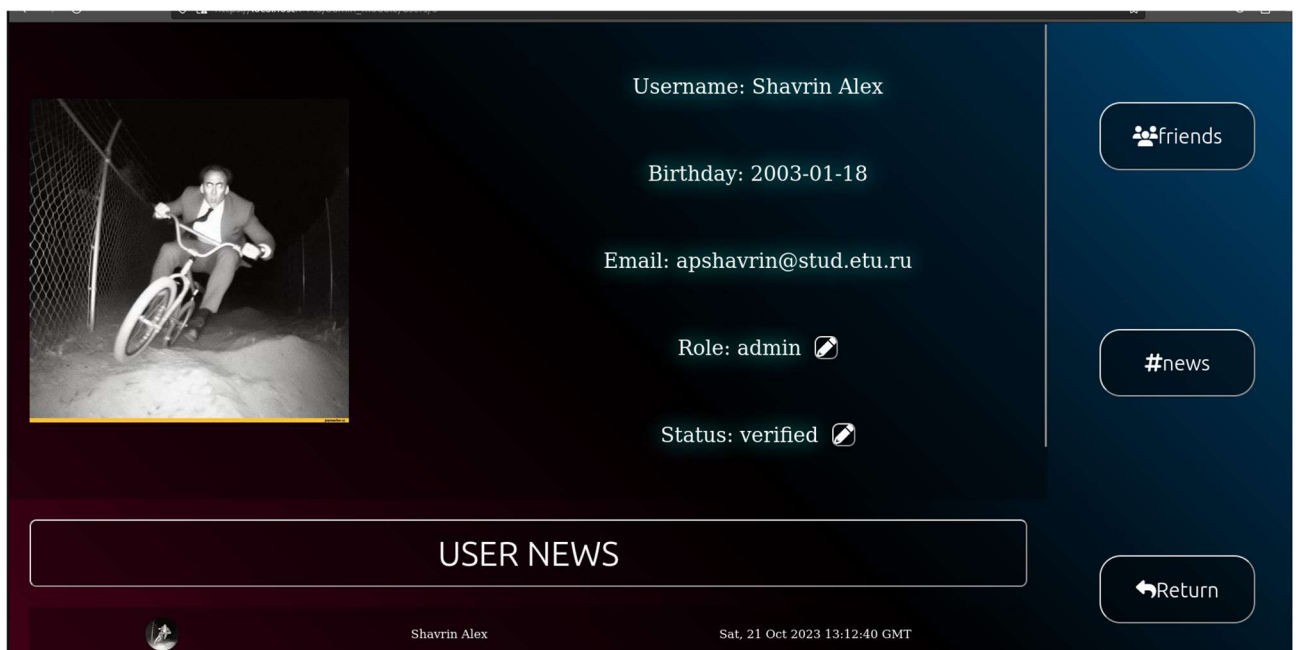


Рис 13. Страница пользователя.

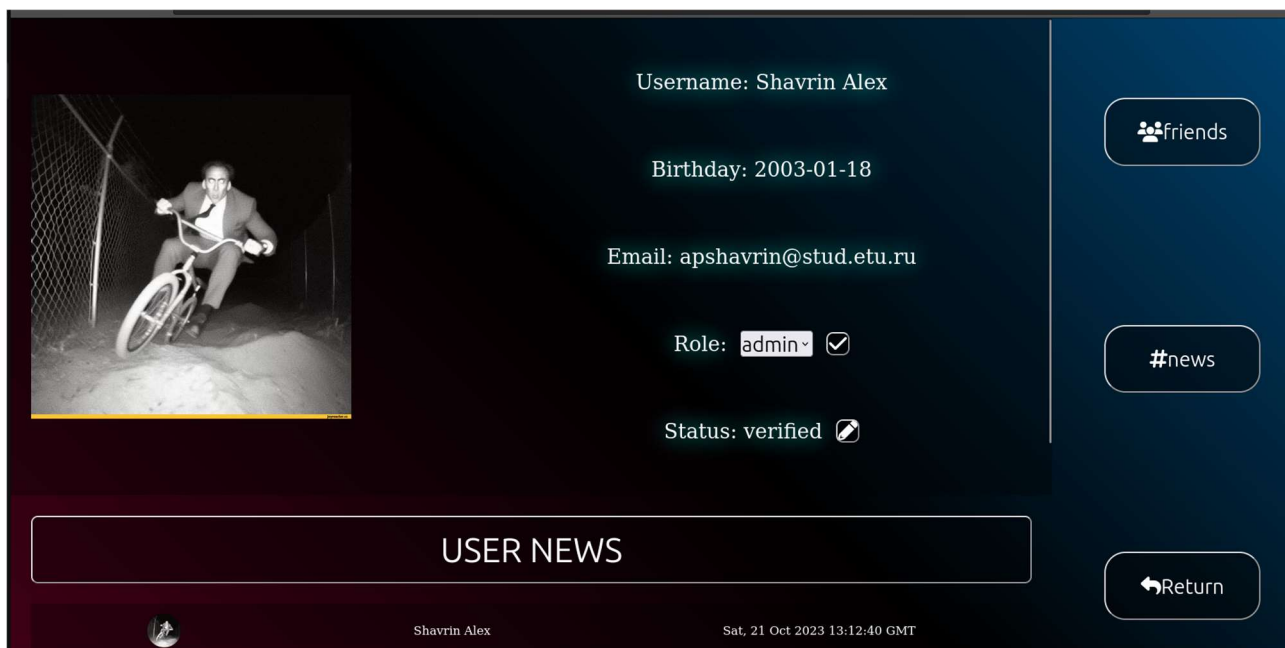


Рис 14. Страница пользователя при изменении данных

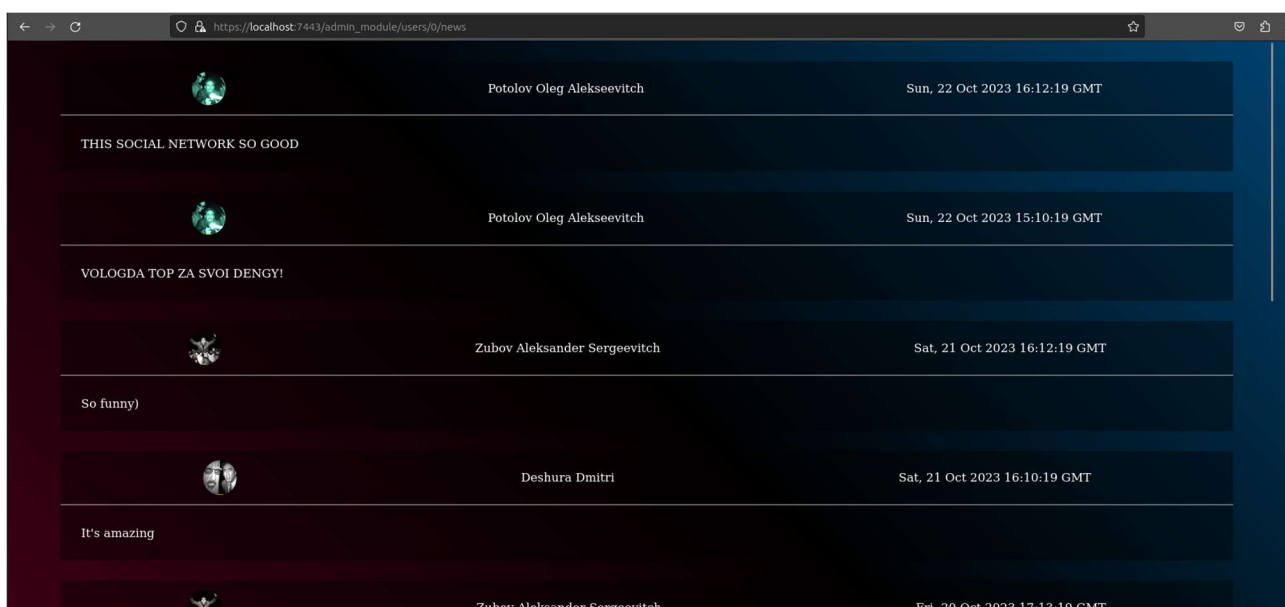


Рис 15. Страница новостей

Выводы.

Изучены возможности применения компилятора Babel, библиотеки jQuery, препроцессора LESS, инструмента выполнения повторяющихся задач GULP, освоение инструмента сборки Webpack, регистрация разработанных модулей, формирования навыков построение структурированных web-приложений, освоены особенностей стандартных библиотек.

