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. 1304 _____ Шаврин А.П.

Преподаватель _____ Беляев С.А.

Санкт-Петербург
2023

Цель работы.

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, добавить или удалить свою фотографию, управлять своими друзьями в социальной сети, добавить сообщение (новость) на свою страницу, 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.

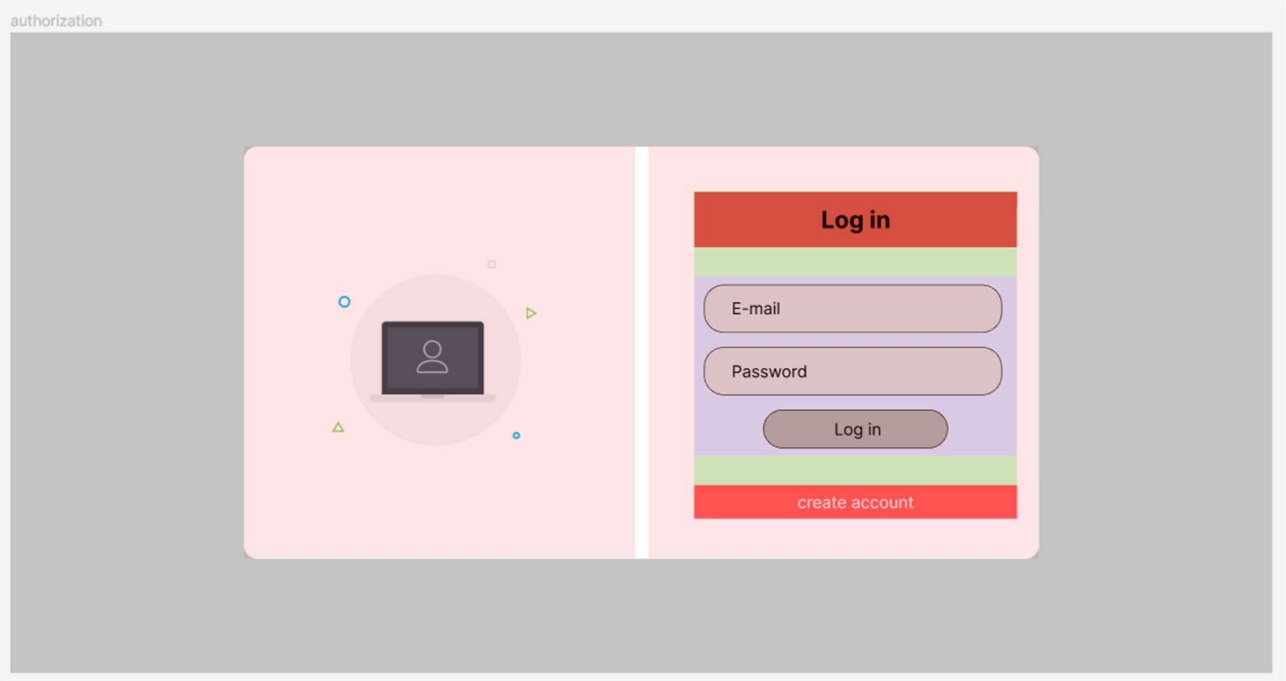
9. Для всех страниц web-приложения разработан макет интерфейса с использованием Figma (https://www.figma.com/).

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

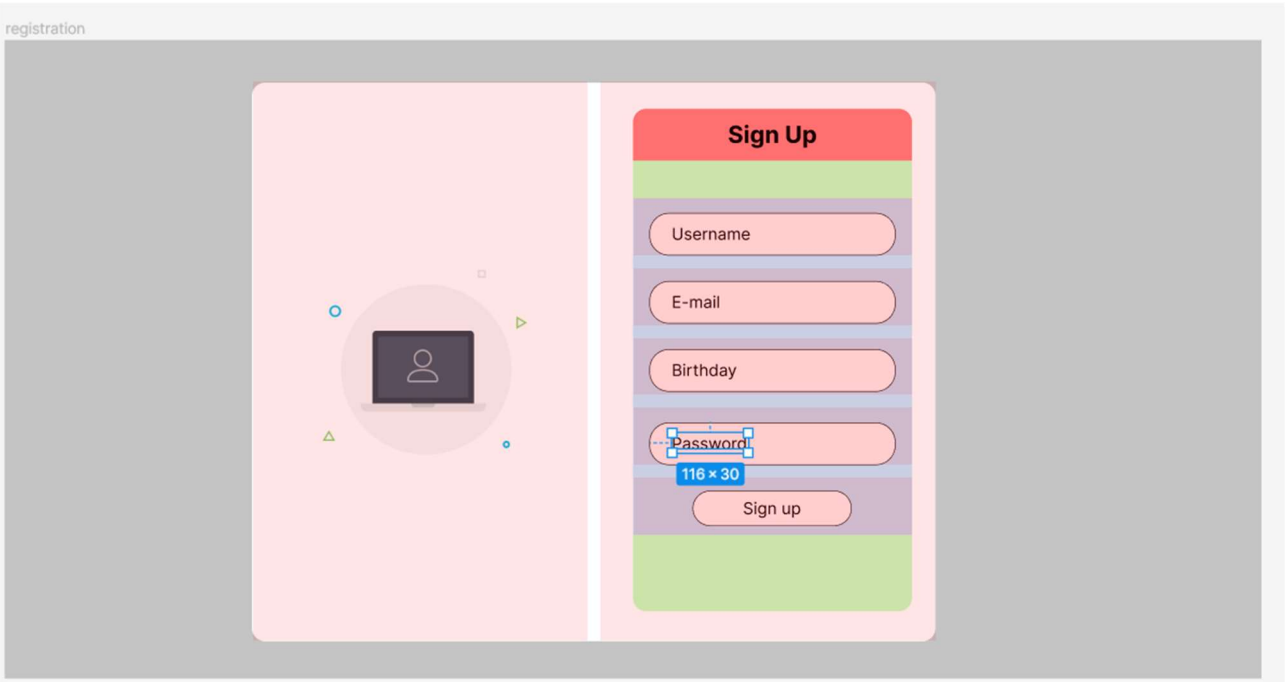
Преимуществом будет использование компонентов Angular Material (https://material.angular.io/).

Выполнение работы.

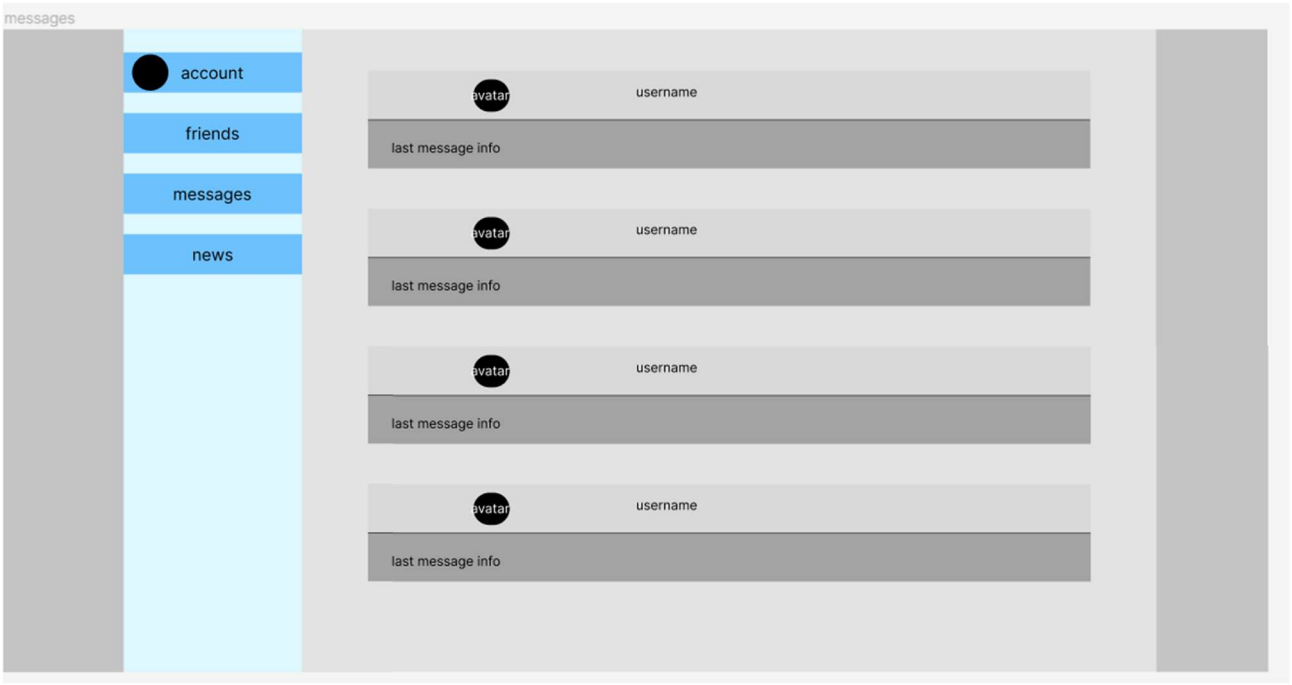
1. Созданы макеты страниц в Figma:

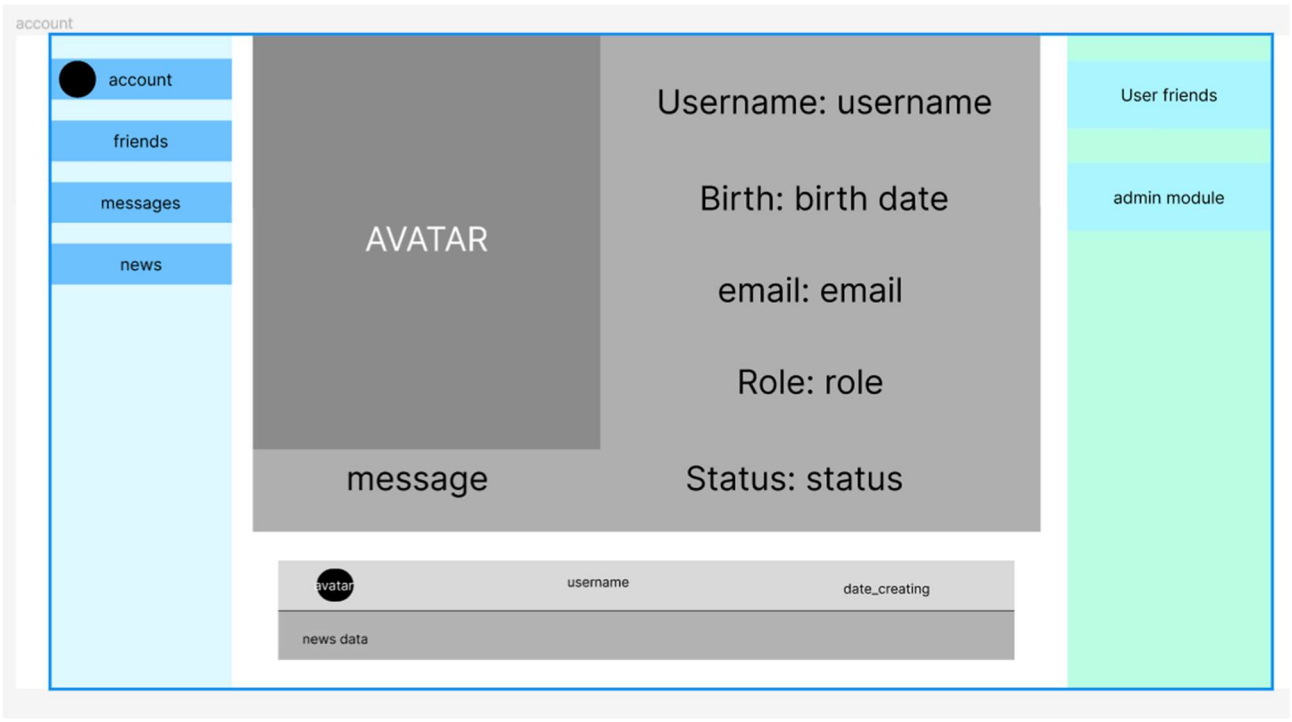


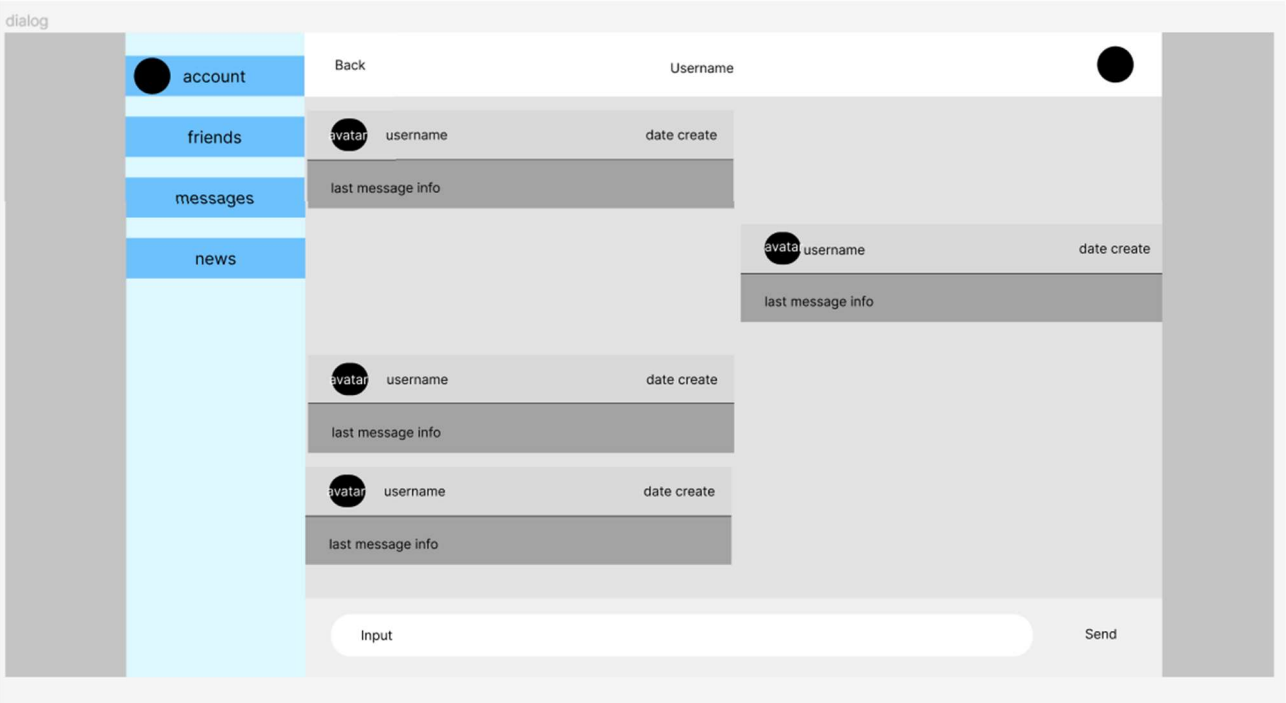
registration



main


messages
account
friends
messages
news
avatar
username
last message info
avatar
username
last message info
avatar
username
last message info
avatar
username
last message info


account
account
friends
messages
news
AVATAR
Username: username
Birth: birth date
email: email
Role: role
Status: status
message
User friends
admin module
avatar
username
date_creating
news data



2. Для серверной части был модифицирован файл router.js из 3й лабораторной

```
97  > router.get("/authorization/login", (req, res) => {···
111   })
112
113 > router.post("/authorization/signup", JsonParser, (req, res) => {···
143   })
144
145 > router.get("/account/user_page", JsonParser, (req, res) => {···
162   })
163
164 > router.get("/account/friends", JsonParser, (req, res) => {···
177   })
178
179 > router.get("/account/news", JsonParser, (req, res) => {···
201   })
202
203 > router.get("/account/messages", JsonParser, (req, res) => {···
236   })
237
238 > router.get("/account/dialog", JsonParser, (req, res) => {···
254   })
```

```
256 > router.get("/account/friends/go_to_dialog", JsonParser, (req, res) => {···
302   })
303
304 > router.post("/account/user_page/change_image", JsonParser, (req, res) => {···
319   })
320
321 > router.post("/account/friends/add_friend", JsonParser, (req, res) => {···
340   })
341
342 > router.post("/account/friends/delete_friend", JsonParser, (req, res) => {···
362   })
363
364 > router.get("/account", JsonParser, (req, res) => {···
408   })
```

3. Для реализации cors были дописаны файлы router.js и server.js

```
12  router.use((req, res, next) => {
13      res.header("Access-Control-Allow-Origin", "*")
14      res.header("Access-Control-Allow-Headers", "Content-Type")
15      res.header('Access-Control-Allow-Methods', 'GET, POST')
16      next()
17  })
```

```
15  const corsOptions = {
16    "credentials": true,
17    'origin': true,
18    'methods': 'GET,HEAD,PUT,PATCH,POST,DELETE',
19    'allowedHeaders': 'Authorization,X-Requested-With,X-HTTP-Method-Override,Content-Type,Cache-Control,Accept'
20  }
21
22  const app = express()
23
24  // Подключение обработчика шаблонов ejs, шаблоны - в папке views
25  app.set("view engine", "ejs")
26  app.set("views", `./views`)
27
28  //Настройка приложения
29  app.use("/public", express.static("public")) // Указание статической папки public
30  app.use(express.json())                      // Обработка параметров в body
31  app.use("/", router)                         // Использование маршрутизации
32  app.use(cors(corsOptions))
```

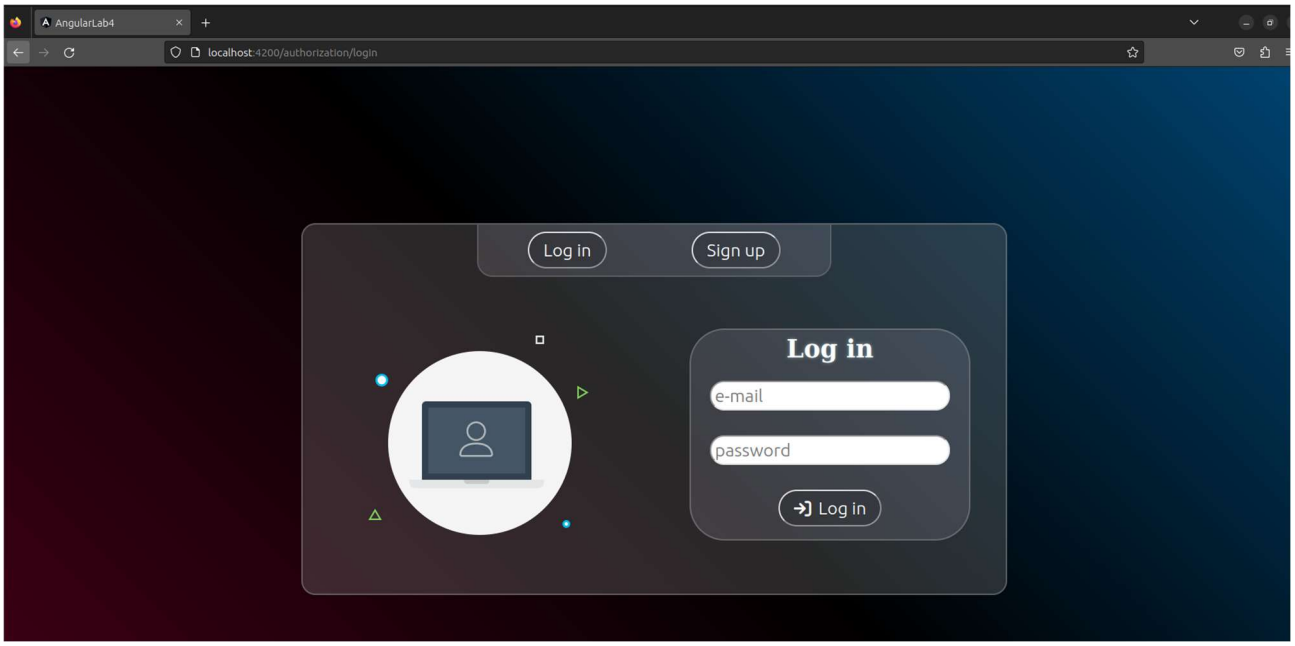
4. Для сокетного взаимодействия в файле server.js дописан следующий код

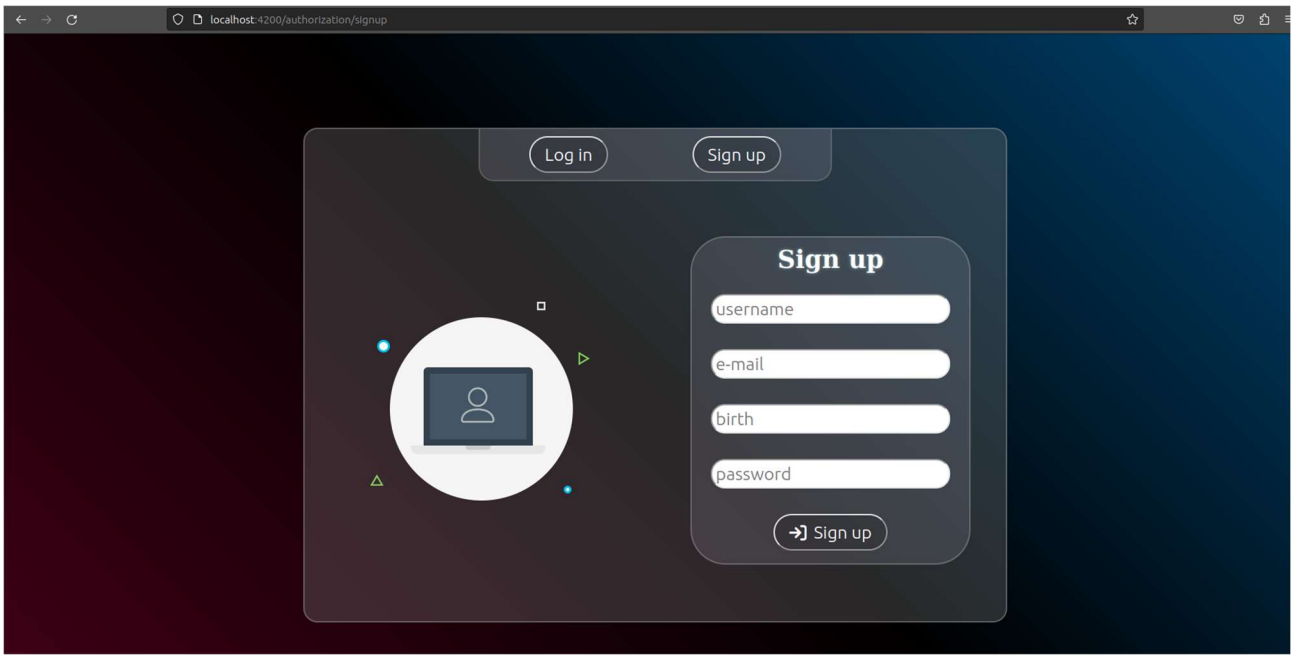
```
46  // Создание сокета
47  let socket_io = new Server(http_server, { cors: corsOptions })
48  socket_io.on('connection', (socket) => {
49    console.log("connect soket " + socket.id)
50
51    socket.on('news', (news) => {···
60    })
61
62    socket.on('message', (data) => {···
78    })
79
80    socket.on('dialog', (data) => {···
82    })
83  })
```

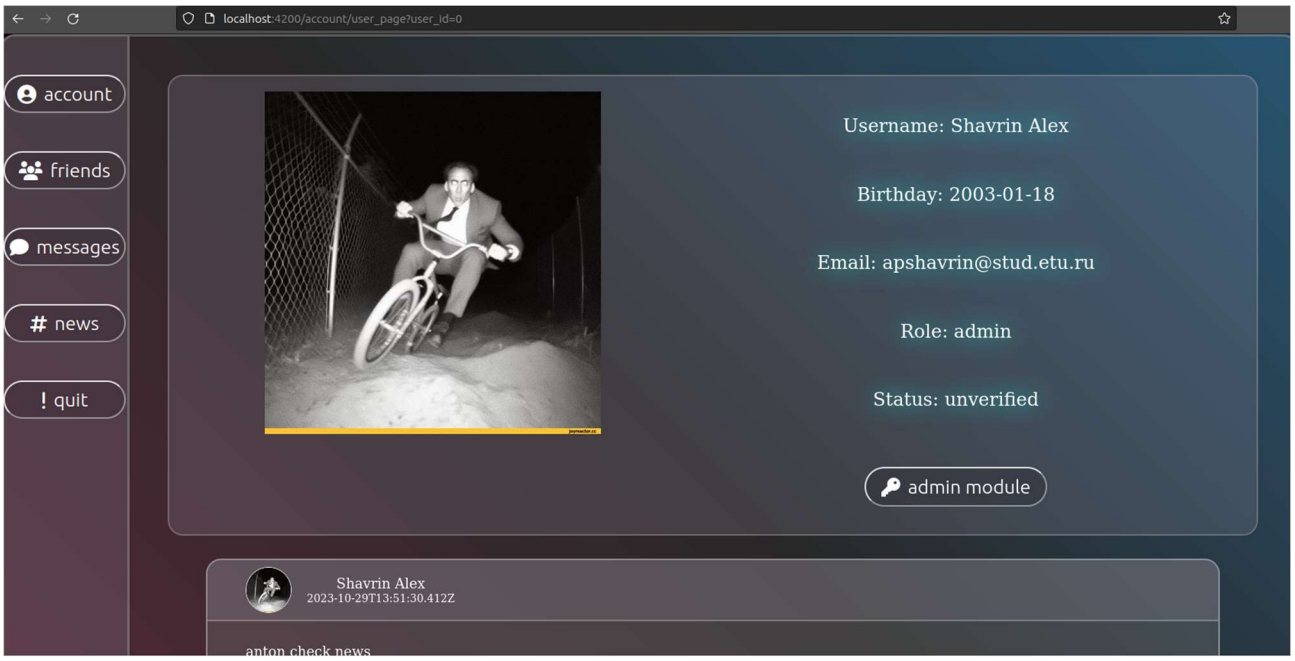
5. Написаны jest тесты для серверной части

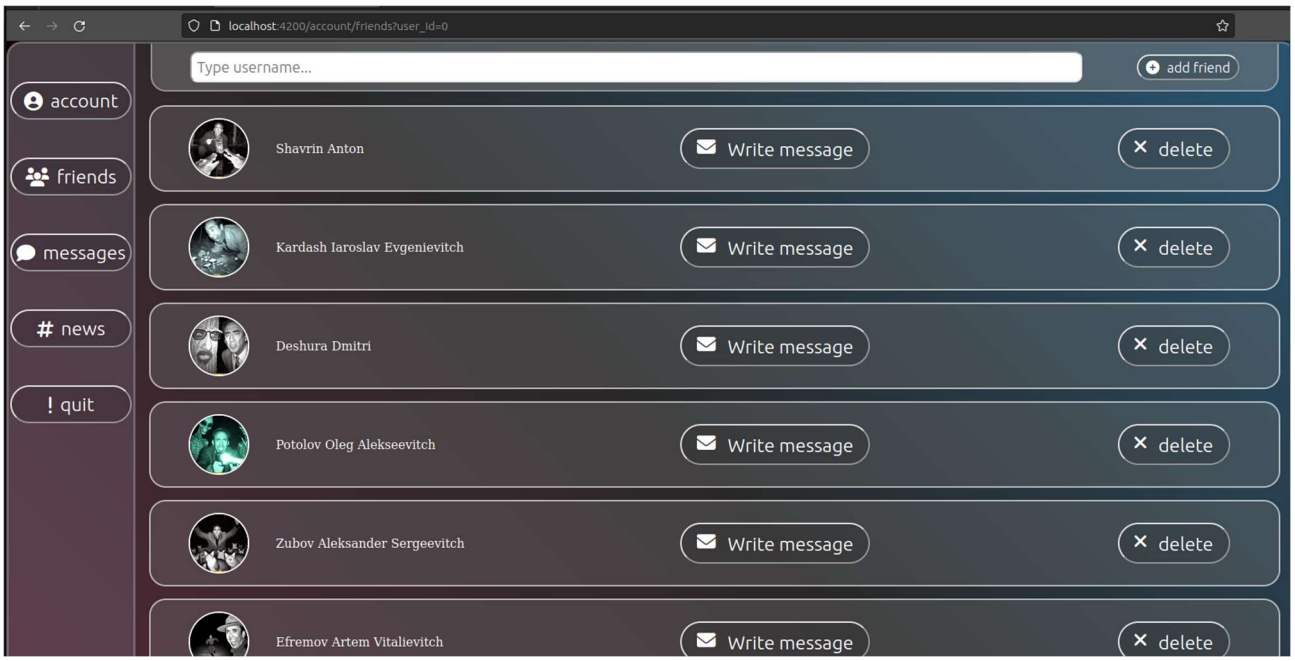
```
describe("test", () => {
    test("login", () => {…
    })

    test("signup", () => {…
    })

    test("account", () => {…
    })

    test("user_page", () => {…
    })

    test("friends", () => {…
    })

    test("news", () => {…
    })

    test("messages", () => {…
    })

    test("dialog", () => {…
    })

    test("go to dialog", () => {…
    })

});
```

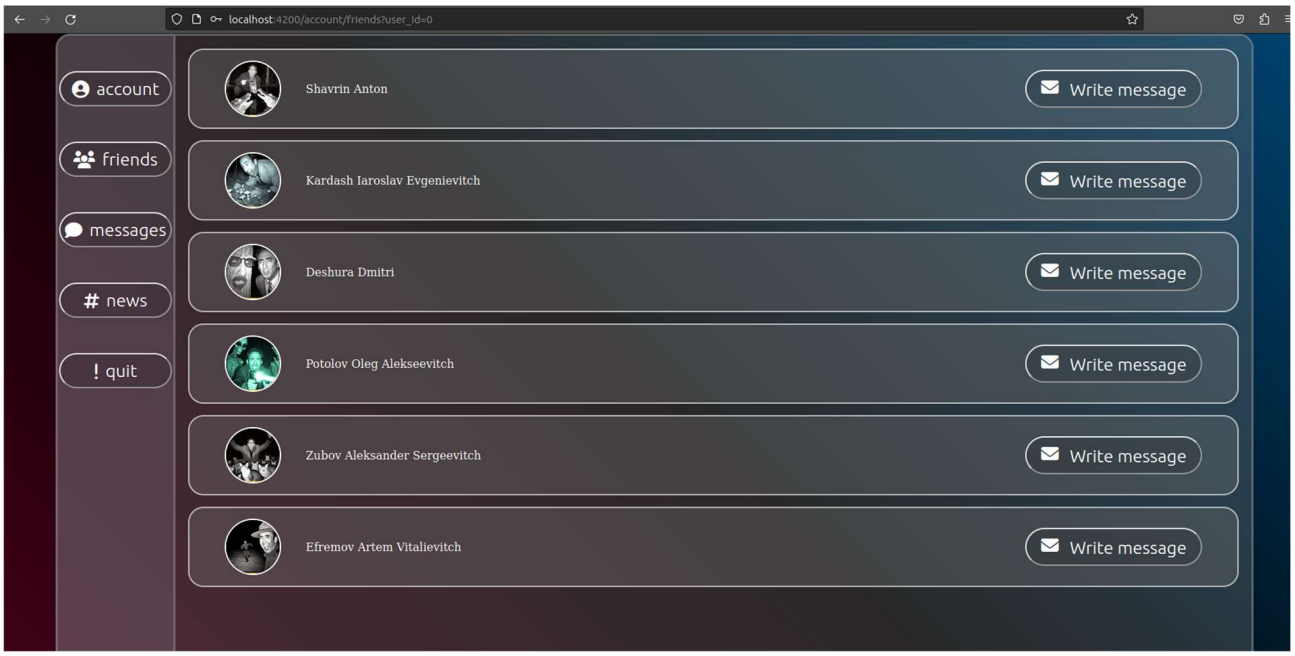
6. Итоговый внешний вид приложения:

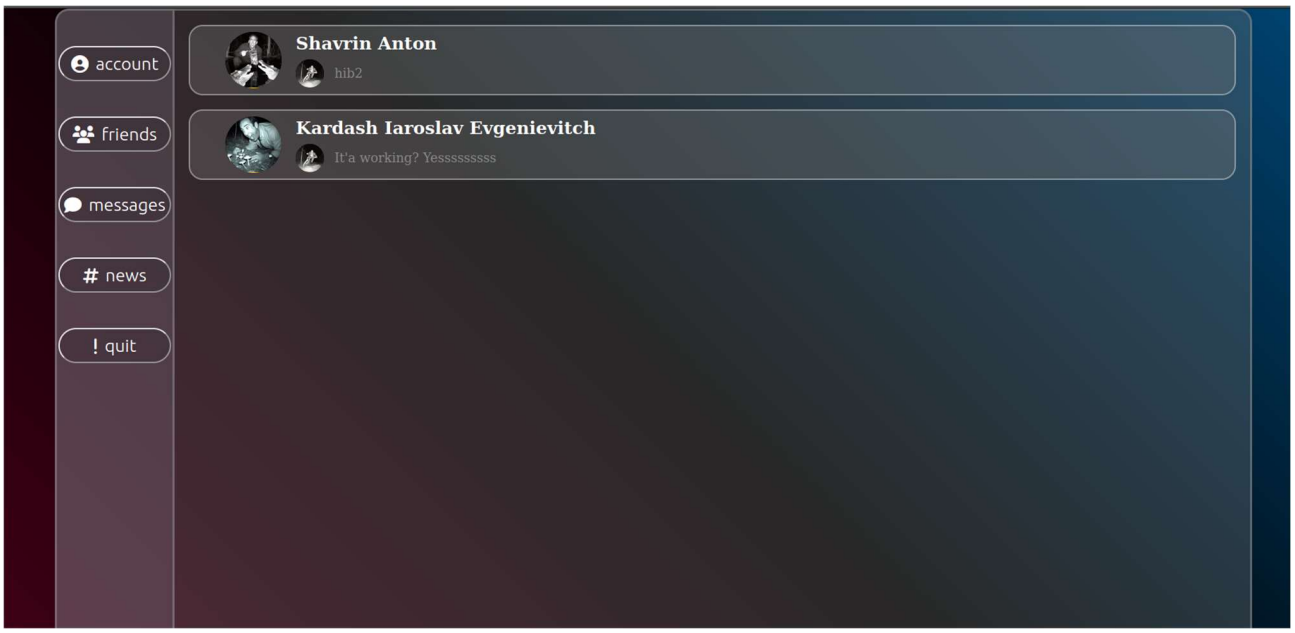


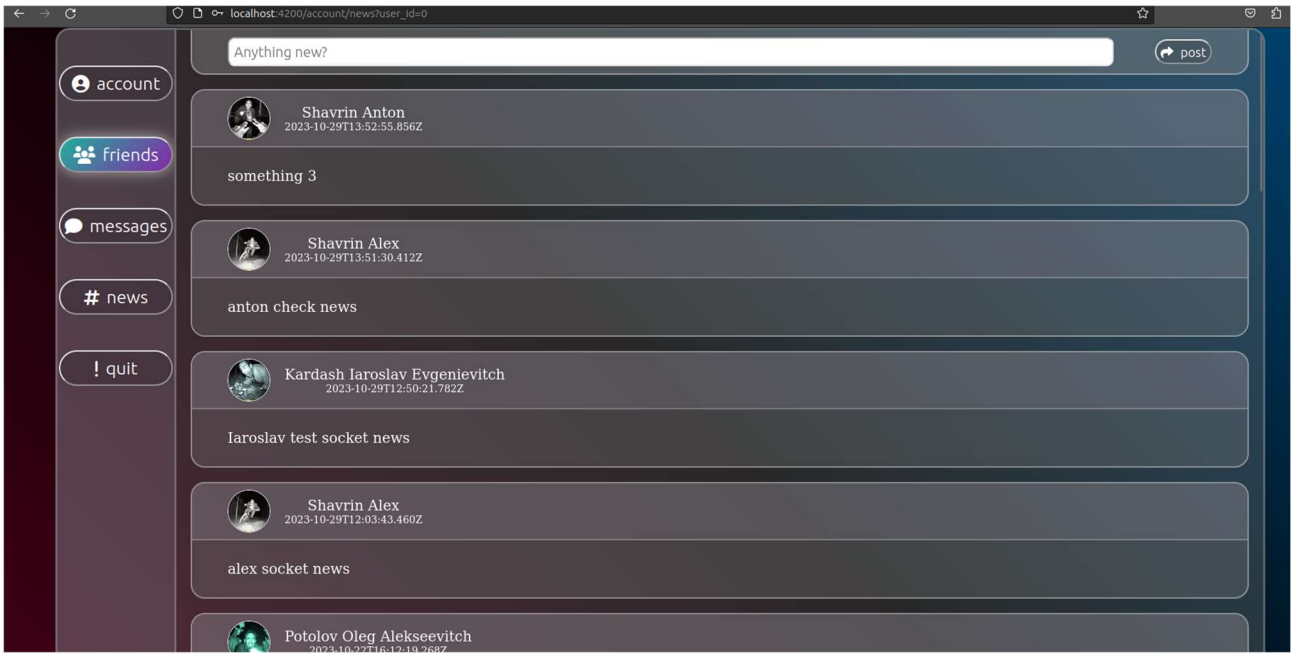

localhost:4200/authorization/signup
Log in
Sign up
Sign up
username
e-mail
birth
password
Sign up


localhost:4200/account/user_page?user_Id=0
account
friends
messages
news
quit
Username: Shavrin Alex
Birthday: 2003-01-18
Email: apshavrin@stud.etu.ru
Role: admin
Status: unverified
admin module
Shavrin Alex
2023-10-29T13:51:30.412Z
anton check news


localhost:4200/account/friends?user_Id=0
Type username...
add friend
account
friends
messages
news
quit
Shavrin Anton
Write message
delete
Kardash Iaroslav Evgenievitch
Write message
delete
Deshura Dmitri
Write message
delete
Potolov Oleg Alekseevitch
Write message
delete
Zubov Aleksander Sergeevitch
Write message
delete
Efremov Artem Vitalievitch
Write message
delete


localhost:4200/account/friends?user_Id=0
account
friends
messages
news
quit
Shavrin Anton
Write message
Kardash Iaroslav Evgenievitch
Write message
Deshura Dmitri
Write message
Potolov Oleg Alekseevitch
Write message
Zubov Aleksander Sergeevitch
Write message
Efremov Artem Vitalievitch
Write message


account
friends
messages
news
quit
Shavrin Anton
hib2
Kardash Iaroslav Evgenievitch
It'a working? Yesssssssss


localhost:4200/account/news?user_id=0
Anything new?
post
account
friends
messages
news
quit
Shavrin Anton
2023-10-29T13:52:55.856Z
something 3
Shavrin Alex
2023-10-29T13:51:30.412Z
anton check news
Kardash Iaroslav Evgenievitch
2023-10-29T12:50:21.782Z
Iaroslav test socket news
Shavrin Alex
2023-10-29T12:03:43.460Z
alex socket news
Potolov Oleg Alekseevitch

Выводы.

Изучены основы языка TypeScript и особенности 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.