МИНОБРНАУКИ РОССИИ

–––––––——————————–––––––

Санкт-Петербургский государственный  
электротехнический университет «ЛЭТИ»

————————————————————

**Web-программирование**

**Лабораторная № 3**

**Тема: СОЗДАНИЕ ВЕБ ПРИЛОЖЕНИЯ С ИСПОЛЬЗОВАНИЕМ EXPRESSJS**

Выполнил: студент группы 3311 Шарпинский Денис

Проверил: Калмыков Михаил Александрович

Санкт-Петербург

Издательство СПбГЭТУ «ЛЭТИ»

2025

Цель работы: Создание веб приложения с использованием библиотеки expressjs, html, css и js.

Перед началом выполнения лабораторной работы следует определиться с предметной областью. Я не буду сильно отклоняться от той, что предложена в методических указаниях. Предметная область приложения – личный блог, в котором пользователи могут указывать свой любимый цвет.

Отсюда сразу же вытекают потребности в следующей функциональности приложения:

1. Регистрация и авторизация новых пользователей.
2. Возможность указывать и сохранять свой любимый цвет.
3. Админ-панель для управления пользователями.

Значит, приложение будет состоять из трёх страниц.

Определим также жизненный цикл приложения:

Первое открытие приложения: пользователь не авторизован, открывается панель регистрации, после регистрации пользователь попадает в профиль

Повторное открытие приложения: пользователь уже был авторизован и его автоматически перенаправляет в профиль.

Особые сценарии:

Открытие админ панели – осуществляется исключительно путём дописывания в адресную строку пути /admin-panel.

Выход пользователя в раздел авторизации – осуществляется либо нажатием кнопки «Выйти», либо путем изменения адресной строки (либо вручную, либо путем роутинга от самого браузера).

В ходе предыдущей работы был создан базовый сервер, работающий на 8000 порту. Для клиентской части создадим папку public. В ней будут храниться html, css и js файлы для работы приложения. Создадим для начала панель авторизации, в ней пользователю будет предоставлена возможность указать логин и пароль, а также кнопка «Авторизоваться».

Необходимые задачи от панели авторизации:

1. Семантическая верстка (указание id у тегов label и input для корректной работы)
2. Реагирование на статус-коды от сервера, если те отличаются от 200-ых (подсветка полей ввода красным цветом)
3. Валидация пустых значений полей ввода (блокирование кнопки «Авторизация»).
4. Навигация в раздел /profile при успешной авторизации / регистрации.

Всё это можно осуществить, используя базовые html, css и js. Но для клиента я также буду использовать typescript, а потому, аналогично 2ой лабораторной работе, я создам конфиг и в папке public. После написания необходимых файлов мы получим следующую структуру папки public:

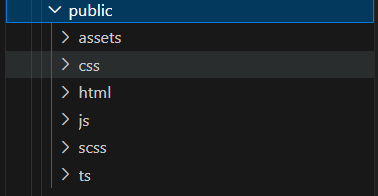


Рис. 3.1

Также вместо css я использовал scss, т к он позволяет использовать вложенность в куда более удобном виде, чем базовый css, а также даёт возможность писать свои функции.

В результате была получена следующая страница:

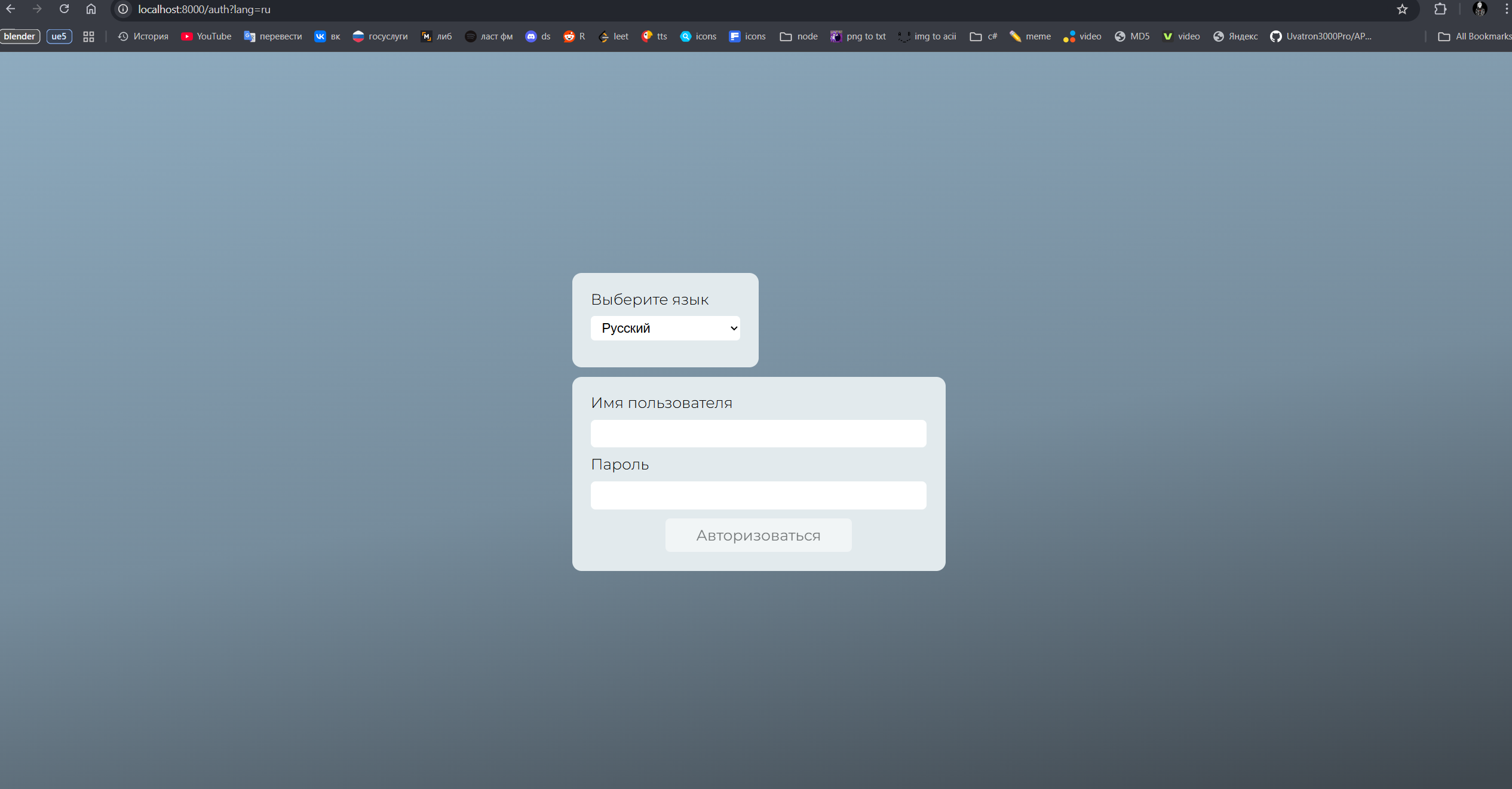


Рис. 3.2

Выводы:

В ходе выполнения лабораторной работы была создана веб-страница для авторизации пользователя с использованием стандартного браузерного api. Для клиентской части также был выбран Typescript, поскольку я предпочитаю его базовому js, т к он поддерживает строгую типизацию.