МИНОБРНАУКИ РОССИИ

–––––––——————————–––––––

Санкт-Петербургский государственный  
электротехнический университет «ЛЭТИ»

————————————————————

**Web-программирование**

**Лабораторная № 9**

**Тема: МОДУЛЬНОЕ ТЕСТИРОВАНИЕ WEB -ПРИЛОЖЕНИЯ**

Выполнил: студент группы 3311 Шарпинский Денис

Проверил: Калмыков Михаил Александрович

Санкт-Петербург

Издательство СПбГЭТУ «ЛЭТИ»

2025

Цель работы: написать модульные тесты для приложения и произвести тестирование.

Для написания модульных тестов ещё во второй лабораторной работе была установлена библиотека jest, а также дополнительно ts-jest (для поддержки ts) и supertest (для выполнения http запросов).

Для модульного тестирования необходимо создать два файла:

1. jest.config.js
2. server.test.ts

В конфигурационном файле я указал, что буду использовать пресет ts-jest.

В файле server.test.ts были реализованы модульные тесты для основных маршрутов сервера. С помощью библиотеки supertest производится эмуляция HTTP-запросов к express-приложению, а библиотека jest используется как тестовый раннер. Для каждого теста создаётся изолированное приложение Express с подключёнными middleware (express.json(), cookieParser(), express.static()), а также временно определяются тестовые данные о пользователях.

Были протестированы следующие сценарии:

1. POST /api/auth – проверка логики авторизации:
   * создание нового пользователя при отсутствии в базе;
   * успешная авторизация при верных данных;
   * ошибка (401) при неправильном пароле.
2. GET /api/users – возврат списка пользователей без раскрытия паролей.
3. GET /api/user/:id – получение информации о конкретном пользователе, проверка корректной обработки несуществующего ID.
4. DELETE /api/user – удаление пользователя по ID, с обработкой ошибок при невалидном или отсутствующем ID.
5. POST /api/color – изменение цвета пользователя с записью в cookie, а также проверка редиректа на /auth для неавторизованных пользователей.
6. GET / – проверка, что при обращении к корневому маршруту сервер возвращает 404.

Кроме того, в тестах проверялась вспомогательная функция getLanguage, отвечающая за выбор языка интерфейса в зависимости от query-параметра.

Результаты выполнения тестов:

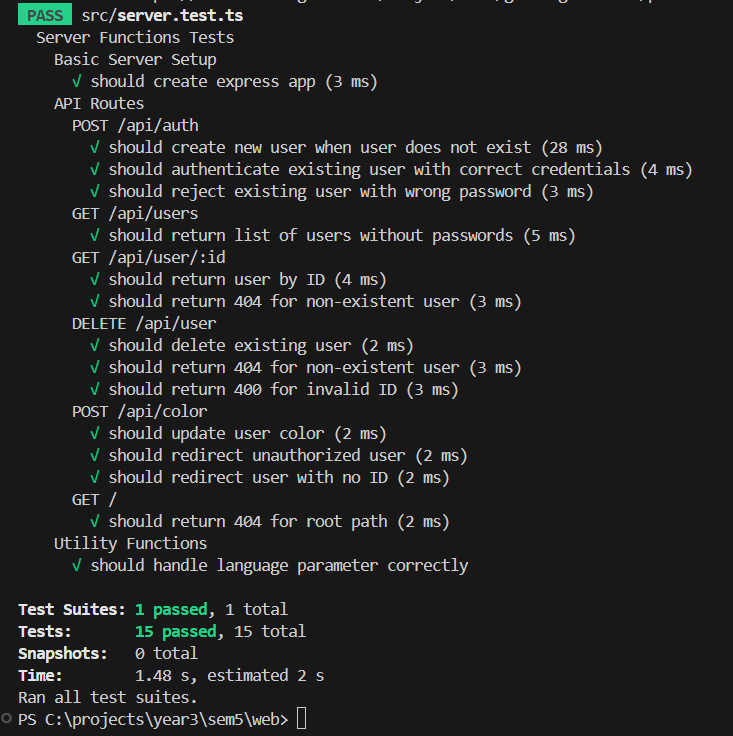


Рис 9.1

Выводы:

В ходе выполнения лабораторной работы было проведено модульное тестирование приложения.  
Для этого использовались библиотеки jest и supertest, позволившие эмулировать HTTP-запросы и проверять корректность работы серверных маршрутов.