

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Бэк-энд разработка

Отчет

Домашнее задание 3: Документирование API средствами
swagger и Postman

Выполнил:

Лазебный Всеволод

Группа К3344

Проверил:

Добряков Д. И.

Санкт-Петербург

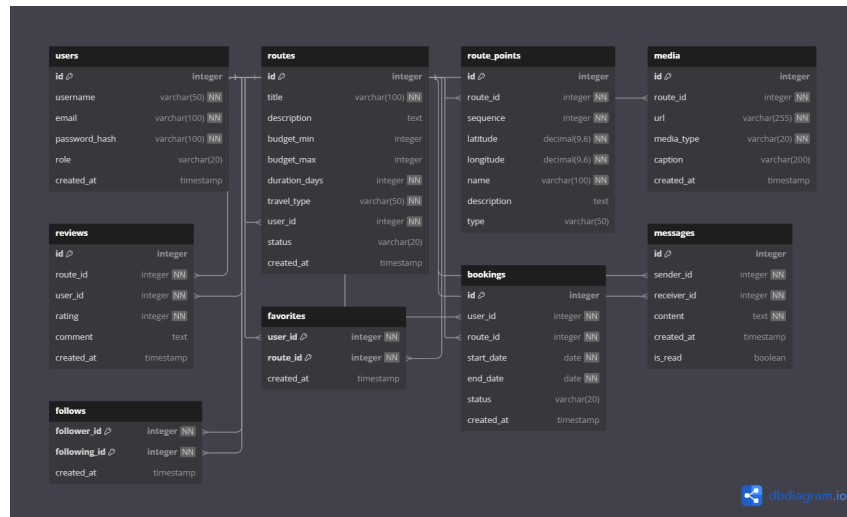
2025 г.

Задача

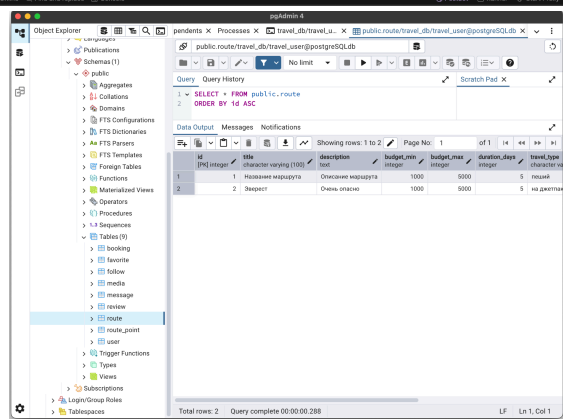
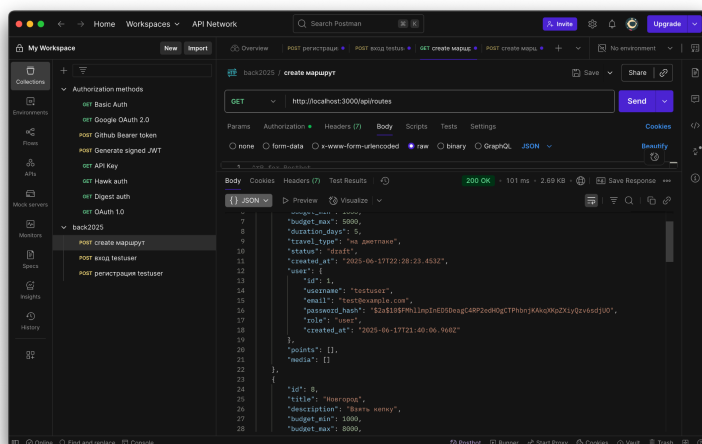
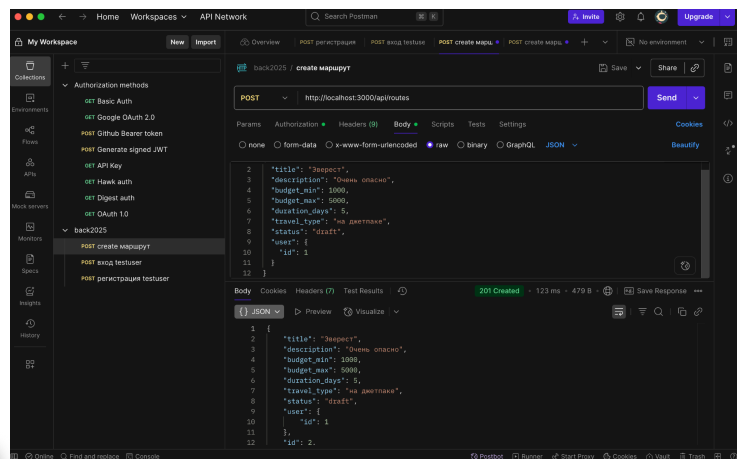
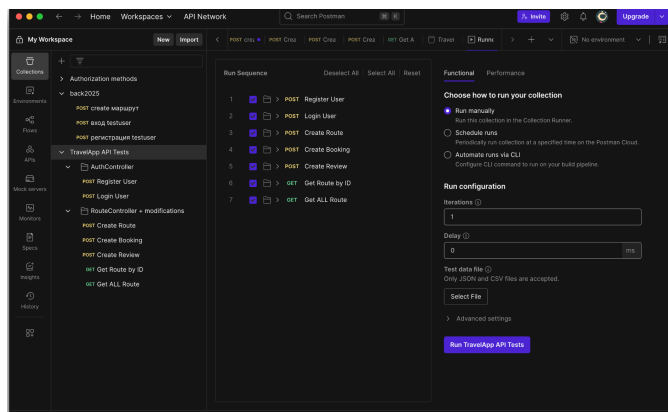
1. Реализовать автодокументирование средствами swagger;
2. Реализовать документацию API средствами Postman.

Ход работы

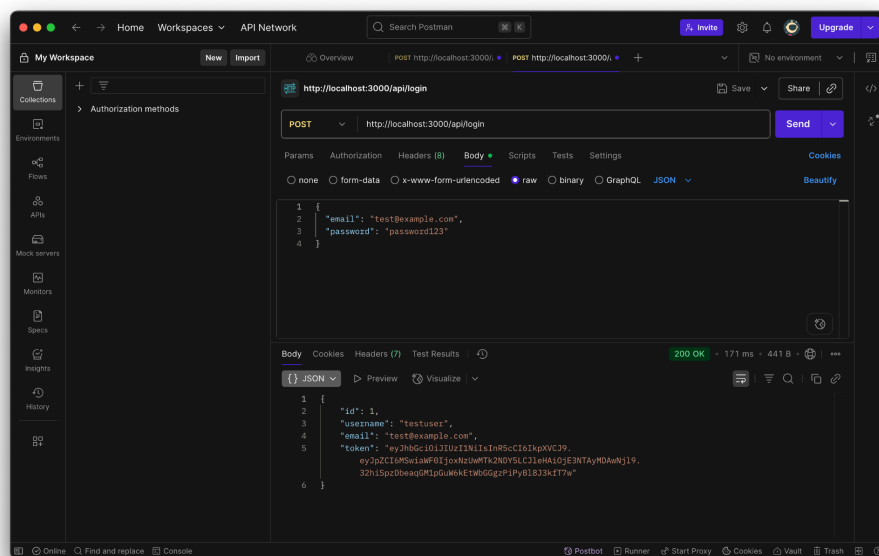
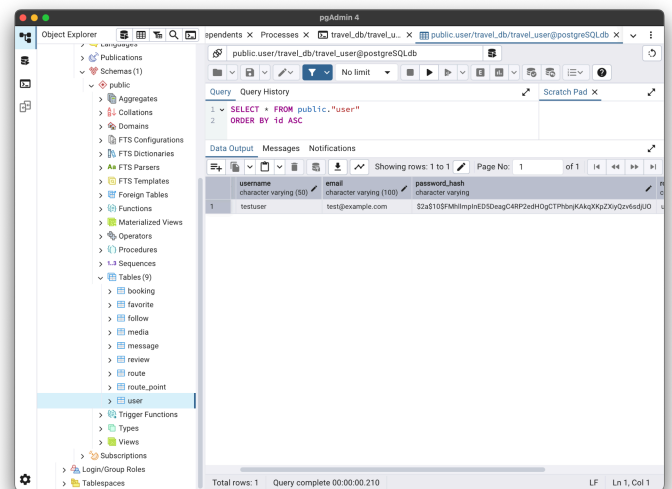
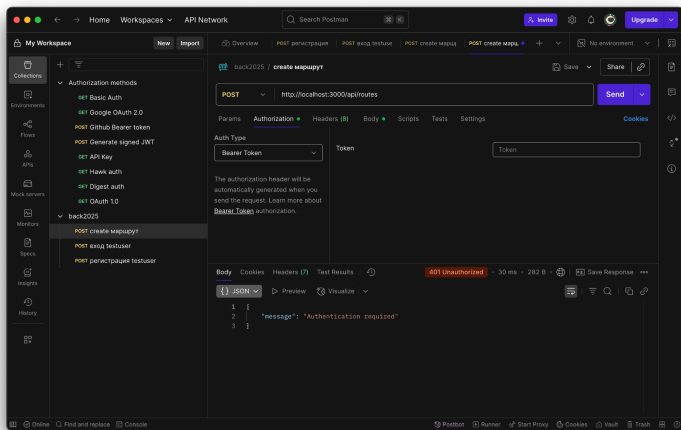
В рамках ДЗ1 была спроектирована следующая схема данных:



Для начала я сразу обратился к Postman, тк совсем недавно взаимодействовал с ним в рамках курса Web-программирование:



На скриншотах видно, что я достаточно потыкался в работу Postman, как можно заметить, всё успешно работает благодаря получению токена. Там, где требуется по условию нужна авторизация пользователя:



Затем предстояла работа с swagger. Для начала была произведена настройка экспресс-приложения в скрипте:

```

20
21 // Создание Express приложения
22 const app : Express = express();
23 app.use(express.json());
24
25 // Swagger configuration
26 const swaggerOptions: swaggerJsdoc.Options = {
27   definition: {
28     openapi: '3.0.0',
29     info: {
30       title: 'Travel Platform API',
31       version: '1.0.0',
32       description: 'API documentation for Travel Platform',
33     },
34     servers: [
35       {
36         url: 'http://localhost:3000/api',
37         description: 'Development server',
38       },
39     ],
40     components: {
41       securitySchemes: {
42         bearerAuth: {
43           type: 'http',
44           scheme: 'bearer',
45           bearerFormat: 'JWT',
46         },
47       },
48     },
49     security: [
50       {
51         bearerAuth: [],
52       },
53     ],

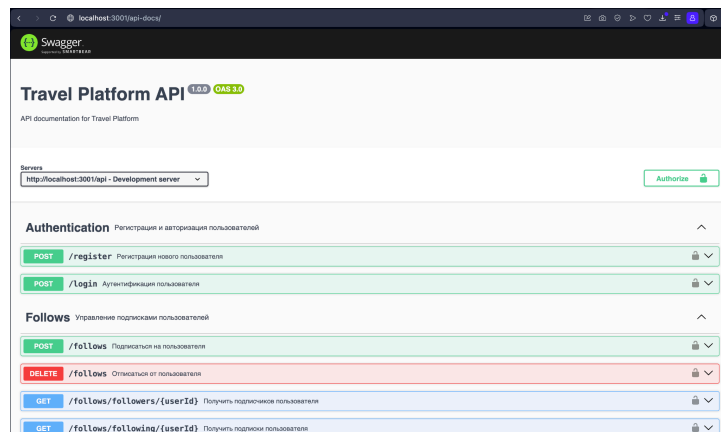
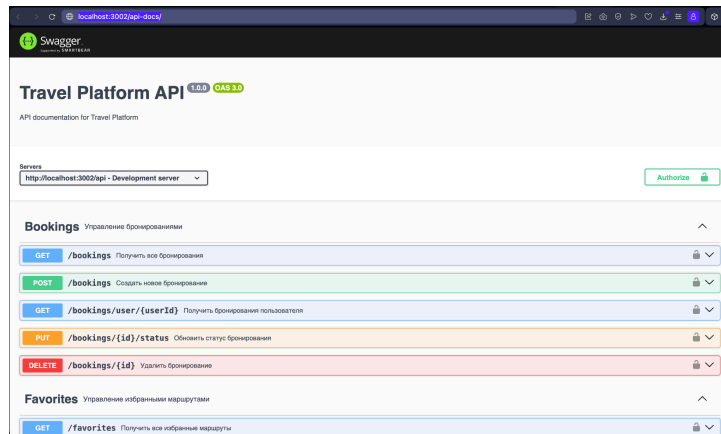
```

```

39     ],
40     components: {
41       securitySchemes: {
42         bearerAuth: {
43           type: 'http',
44           scheme: 'bearer',
45           bearerFormat: 'JWT',
46         },
47       },
48     },
49     security: [
50       {
51         bearerAuth: [],
52       },
53     ],
54   },
55   apis: ['./src/routes.ts', './src/controllers/*.ts'],
56 };
57
58 const swaggerSpec : object = swaggerJsdoc(swaggerOptions);
59 app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerSpec));
60
61 app.use("/api", router);
62

```

А затем я убедился, что всё успешно работает:



Вывод

В ходе работы я вспомнил о работе с postman, а также освежил в памяти знания о настройке swagger и взаимодействии с ним.