

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

Дисциплина: Фронт-энд разработка

Отчет

Лабораторная работа №1

Выполнил:

Пиотуховский Александр

К3441

**Проверил:
Добряков Д. И.**

Санкт-Петербург

2025 г.

Задача

Необходимо будет выполнить вёрстку сайта средствами HTML, CSS и Bootstrap. Продумать и реализовать моменты, в которых необходим JS (например, открытие модальных окон).

Ход работы

Первым делом были изучены эндпоинты, которые можно было бы использовать для взаимодействия веб-страницы с бэкенд сервисов в будущем. На основе API было определено 7 основных страниц:

- index.html - главная страница с популярными рецептами,
- search.html - поиск и фильтрация рецептов,
- recipe.html - детальная страница рецепта,
- profile.html - профиль пользователя с информацией о нём,
- post.html - детальная страница записи (в дальнейшем пост),
- login.html - экран авторизации,
- register.html - экран регистрации.

Поскольку в требованиях было использование Bootstrap были подключены внешние библиотеки Bootstrap, Bootstrap Icons, Bootstrap JS. Подключение осуществляется с помощью CDN, поскольку это позволяет кэшировать данные у пользователя, не требует установку зависимостей, гарантирует получение актуальной версии.

В дальнейшем содержание страницы будет загружаться с бэкенд сервиса, поэтому в этой работе был добавлены js скрипты, которые позволили динамически подгружать моки данных с целью сымитировать реальную загрузку.

На рисунке 1 изображена структура проекта.

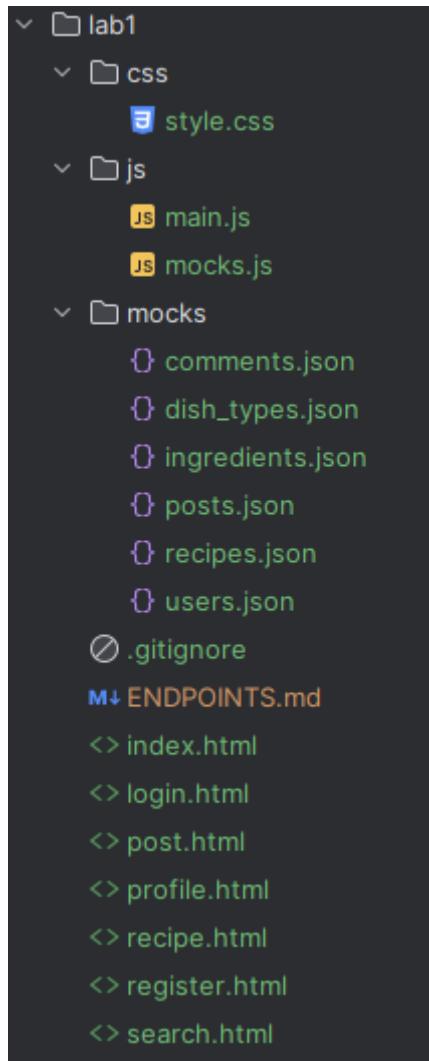


Рисунок 1 – структура проекта

Все стили записаны в css файле. В работе были использованы стили, которые разрабатывались автором во время изучения смежных дисциплин в университете. На рисунке 2 изображена палитра приложения.

```
1      /* Палитра */
2      :root {
3      ■ --color-white: #FFFFFF;
4      ■ --color-text: #BFBFBF;
5      ■ --color-primary: #F77D05;
6      ■ --color-primary-dark: #944b03;
7      ■ --color-secondary: #EEEEEE;
8      ■ --color-secondary-dark: #DDDDDD;
9      ■ --color-secondary-text: #232323;
10     ■ --color-bg: #3E3E3E;
11     ■ --color-card-bg: #232323;
12     ■ --color-border: #3E3E3E;
13     --border-radius: 8px;
14     --transition: all 0.3s ease;
15 }
```

Рисунок 2 – палитра приложения

В приложении использовался js для динамической загрузки данных. При открытии страницы в первую очередь отрисовывается спиннер, далее идёт запрос на получение данных, полученный ответ рендерится на странице, после чего спиннер скрывается. На рисунке 3 изображен пример динамической загрузки рецептов на главной странице.

```

<div class="loading-spinner">
    <div class="spinner"></div>
</div>

<div class="recipes-grid content-hidden">
</div>
</div>
</section>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<script src="js/main.js"></script>
<script src="js/mocks.js"></script>

<script>
    async function loadIndexRecipes() {
        const spinner = document.querySelector(".loading-spinner");
        const grid = document.querySelector(".recipes-grid");

        try {
            const recipes = await MockData.loadJSON("mocks/recipes.json");
            const users = await MockData.loadJSON("mocks/users.json");

            if (!recipes || !users) {
                throw new Error("Failed to load data");
            }

            grid.innerHTML = recipes.map(recipe => {
                const author = users.find(u => u.id === recipe.authorId);
                return MockData.renderRecipeCard(recipe, author);
            }).join("");

            spinner.style.display = "none";
            grid.classList.remove("content-hidden");
        } catch (error) {
            console.error("Error:", error);
            spinner.innerHTML = '<p class="text-danger">Ошибка загрузки данных</p>';
        }
    }
    loadIndexRecipes();
</script>

```

Рисунок 3 – пример динамической загрузки и рендера рецептов

На каждой странице активно используются bootstrap компоненты. Компонент navbar отвечает за активную навигацию (шапку), в которой есть ссылки на другие страницы. Продуман тогл, который отображается только на мобильных устройствах (с помощью тега collapse). С помощью компонента badge были реализованы бейджи. В работе они используются для красивых “тегов” у заказов: они могут содержать информацию о типе блюда, сложность рецепта, количество фотографий и комментариев.

С помощью JS в шапке сайта всегда подсвечивается страница, на которой пользователь находится прямо сейчас. Из полученного url

извлекается имя файла, после чего находится ссылка с таким же href и к нему добавляется класс `active`. Также была написана функция, которая форматирует абсолютное время в относительное: сегодня, вчера, X дней назад.

У рецептов и постов могут быть комментарии. Поскольку API позволяет сделать древовидные комментарии (ответы), на странице таковые тоже могут быть отрисованы с помощью рекурсивной функции. На рисунке 4 изображен итоговый вид комментариев.

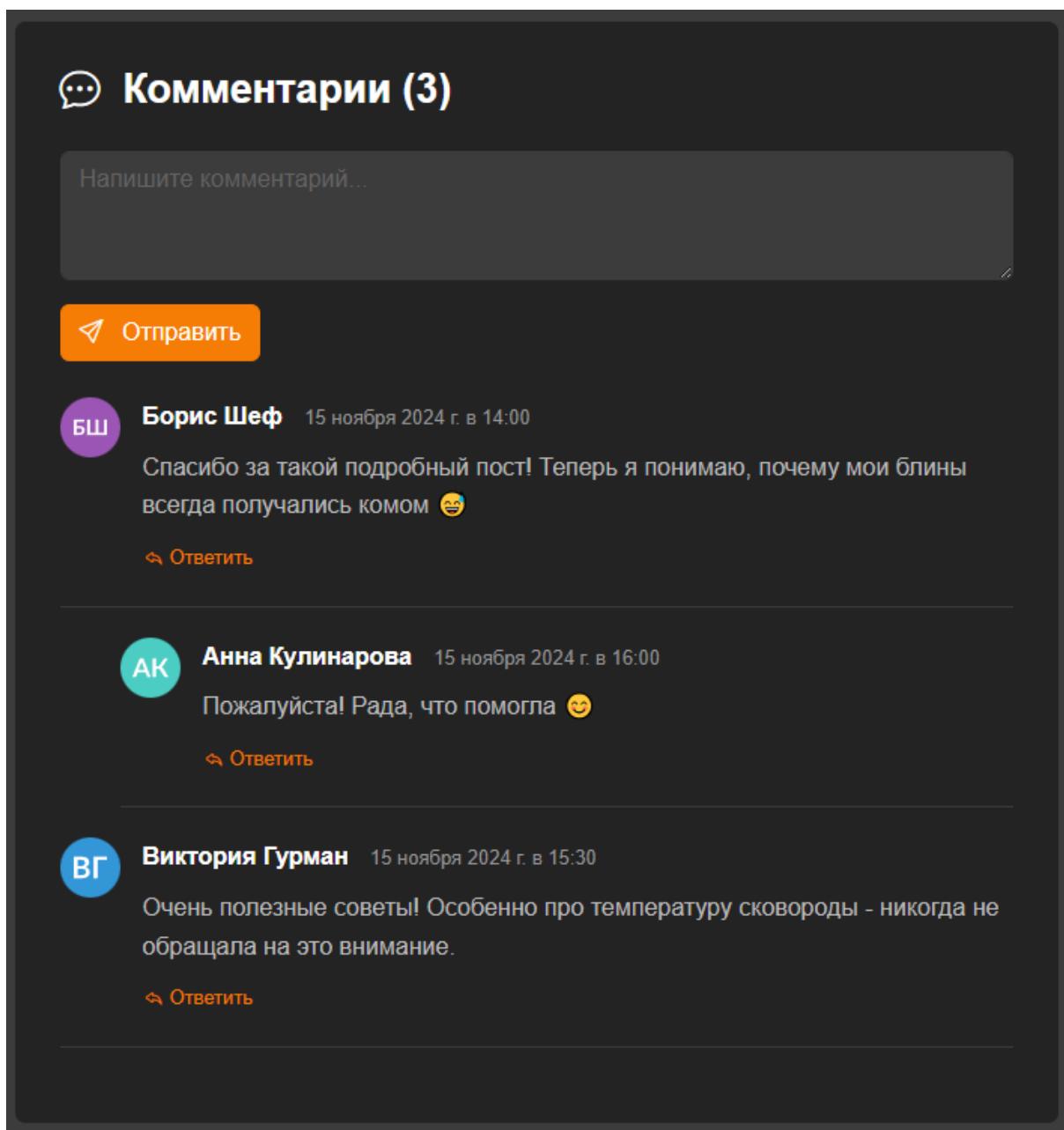


Рисунок 4 – отрисованные комментарии

Вывод

В рамках лабораторной работы была выполнена вёрстка сайта средствами HTML, CSS, Bootstrap. Базовое понимание flexbox и grid, полученное в результате выполнения домашнего задания 1, сделало работу с bootstrap проще. В дальнейшем к этому варианту можно будет добавить настоящие rest запросы на бэк, чтобы исключить моки из проекта.