САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Фронт-энд разработка

Отчет по теме: «вёрстка сайта средствами HTML, CSS и Bootstrap»

Лабораторная работа №1

Выполнил:

Суворин И. А.

Группа 1.1

Проверил:
Добряков Д. И.

Санкт-Петербург

2025 г.

Для выполнения ЛР1 был выбран вариант №5 «Сервис для обмена рецептами и кулинарных блогов».

Структура проекта:

- Index.html – главная страница со списком рецептов,
- Search.html – страница поиска рецептов по названию и сложности,
- Profile.html – личный кабинет пользователя,
- Js/app.js – основной файл с логикой работы приложения,
- Js/data.js – данные для сайта, рецепты,
- Css/style.css – дополнительные стили или правки, что не покрыл Bootstrap,
- Assets/img/….png – место для хранения картинок png-формата.

Первоначальная версия ЛР была переделана. В текущей версии были упрощены следующие аспекты:

— Index.html – главная страница была упрощена: были убраны лишние описания и кнопки, шапка стала проще, стили и сетка карточек уменьшены. Здесь отображается список доступных рецептов. Страница содержит контейнер, который заполняется при помощи JS. Сами же карточки формируются на основе данных из data.js и данных, которые добавил пользователь.

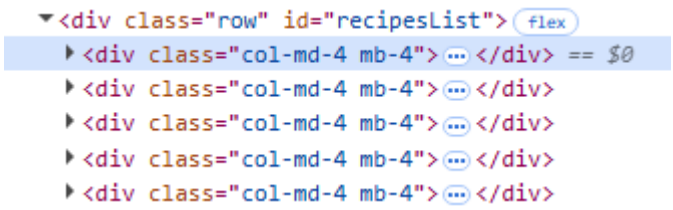

Рисунок 1 – работа контейнера

— Profile.html – страница предназначена для входа пользователя в систему и управлением собственными рецептами. На данном этапе пользователь может только добавить рецепт самостоятельно, удаление происходит через очистку LocalStorage. Версия файла была сильно упрощена:

— Были убраны вкладки (для работы со своими рецептами и добавленными) и модальные окна,

— Была упрощена структура расположения блоков – они идут друг за другом.

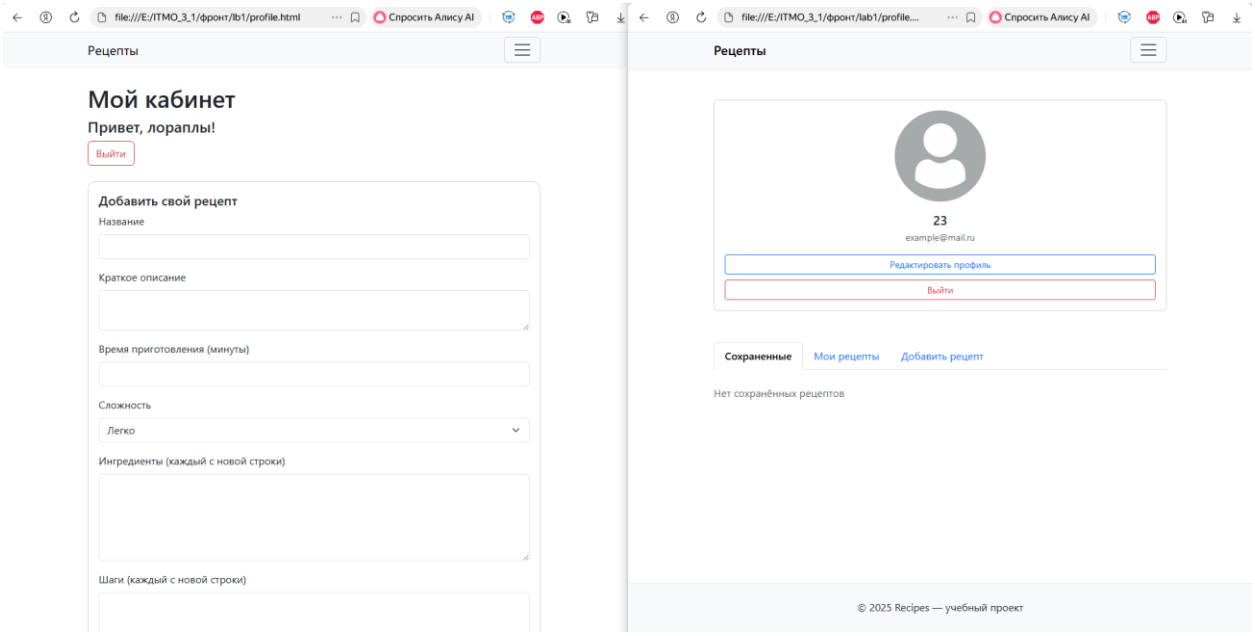

Рисунок 2 – визуальное сравнение (слева новая версия)

— Recipe.html – в прошлой версии была реализована верстка с галереей и чекбоксами для данных. В новой идет простая структура, где js вставляет рецепт. Само добавление реализовано через html-форму. При отправке данные сохраняются в LocalStorage в «myCustomRecipes». Были убраны локальные стили для тех же чекбоксов и галереи.

— Search.html – были убраны лишние фильтры и оставлены только необходимые (текст + сложность).

— Data.js – идентичный файлы, но в новой версии сокращен на 2 рецепта.

— App.js:

— Были изменены общий подход и структура. Если раньше код выглядел перегруженным; использовались вспомогательные функции, довольно сложная логика рендера карточек; реализованы лайки и комментарии добавлением рецепта в избранное; авторизация была более нагруженной, то сейчас общая структура стала более читаемой, так как стала разделена на простые логические функции, которые отвечают одной конкретной задаче. Также заметно снизился функционал.

— Была изменена авторизация пользователя. Раньше был реализован список хранения пользователей, что позволяло перемещаться между аккаунтами и работать с рецептами «индивидуально», более сложная форма для заполнения данных пользователя (email + пароль + подтверждение пароля + загрузка фото для аватара). Сейчас достаточно имени пользователя (есть ограничение на длину [от 2 исмволов]) и его пароля.

— Был изменен подход к хранению данных. Если раньше могли использоваться разные источники данных (массив «RECIPES», кастомные рецепты, избранные рецепты и лайки), то сейчас весь набор данных состоит из данных по рецептам из массива и кастомных.

```
// получить все рецепты: из data.js + добавленные пользователем
function getAllRecipes() {
    let custom = JSON.parse(localStorage.getItem("myCustomRecipes") || "[]");
    return recipes.concat(custom);
}
```

— Раньше карточка создавалась через «renderCard», которая содержала в себе работу с лайками и прочим, то есть всем, что было на странице. Теперь же карточка строится в функции «showRecipes». Карточка буквально создается построчно из информации по data.js

```
// показ карточек рецептов (на главной и поиске)
function showRecipes(list) {
    let container = document.getElementById("recipesList");
    if (!container) return;

    container.innerHTML = ""; //очистка контейнера перед добавлением новых

    for (let i = 0; i < list.length; i++) {
        let r = list[i]; // r - текущий рецепт
        let card =
            '<div class="col-md-4 mb-4">' +
                '<div class="card h-100 shadow-sm">' +
                    '<img src="' + r.image + '" class="card-img-top" alt="' +
r.title + '" style="height:200px; object-fit:cover;">' +
                        '<div class="card-body d-flex flex-column">' +
                        '<h5 class="card-title">' + r.title + '</h5>' +
                        '<p class="card-text text-muted small">' + (r.short ||
'') + '</p>' +
                        '<div class="mt-auto">' +
                            '<p class="mb-2"><small>Время: ' + r.time + ' мин • '
+ r.difficulty + '</small></p>' +
                            '<a href="recipe.html?id=' + r.id + '" class="btn
btn-primary btn-sm">Открыть рецепт</a>' +
                        '</div>' +
                    '</div>' +
                '</div>' +
            '</div>';
        container.innerHTML += card;
    }
}
```

— Изменен переход на страницу рецепта. Для этого используется функция «showSingleRecipe», где id рецепта извлекается из URL

```
    // цель вытащить id
    let params = new URLSearchParams(window.location.search); //
window.location.search будет иметь вид ?id
    let id = params.get('id');
```

— Раньше добавление происходило через модальное окно. Сейчас это реализовано через HTML-форму. Данные берутся из ее полей и ID (для рецепта, так ка ку каждого свой) создается через «id: Date.now()». Вот так выглядит полная форма.

```
                let newRecipe = {
                    id: Date.now(),
                    title: title,
                    short: short,
                    time: Number(time),
                    difficulty: difficulty,
                    image: "assets/img/example.jpg",
                    ingredients: ingredients,
                    steps: steps,
                    author: localStorage.getItem("myuser") || "Аноним"
                };
```

— Сейчас в профиле отображаются только рецепты, которые он сделал как пользователь.

```
// показ своих рецептов в профиле
function showMyRecipes() {
    let container = document.getElementById("myRecipesList");
    if (!container) return;

    container.innerHTML = "";

    let custom = JSON.parse(localStorage.getItem("myCustomRecipes") || "[]");
    let user = localStorage.getItem("myuser");

    let myList = [];
    for (let i = 0; i < custom.length; i++) {
        if (custom[i].author === user) {
            myList.push(custom[i]);
        }
    }

    if (myList.length === 0) {
        container.innerHTML = '<p class="text-muted">У тебя пока нет своих
рецептов</p>';
```

```
        return;
    }

    for (let i = 0; i < myList.length; i++) {
        let r = myList[i];
        let card =
            '<div class="col-md-6 mb-3">' +
                '<div class="card">' +
                    '<div class="card-body">' +
                        '<h5>' + r.title + '</h5>' +
                        '<p class="small text-muted">' + (r.short || '') + '</p>'
+
                        '<small>Время: ' + r.time + ' мин • ' + r.difficulty +
'</small>' +
                        '<a href="recipe.html?id=' + r.id + '" class="btn btn-sm
btn-outline-primary mt-2">Открыть</a>' +
                    '</div>' +
                '</div>' +
            '</div>';
        container.innerHTML += card;
    }
}
```

— Css/style.css – адаптирован под карточки и блоки. Теперь идет упор на конкретные элементы внутри самой карточки (расположение кнопки с отступом и расположение текста с отступом + незначительный визуальный эффект)

Результат работы приложения:

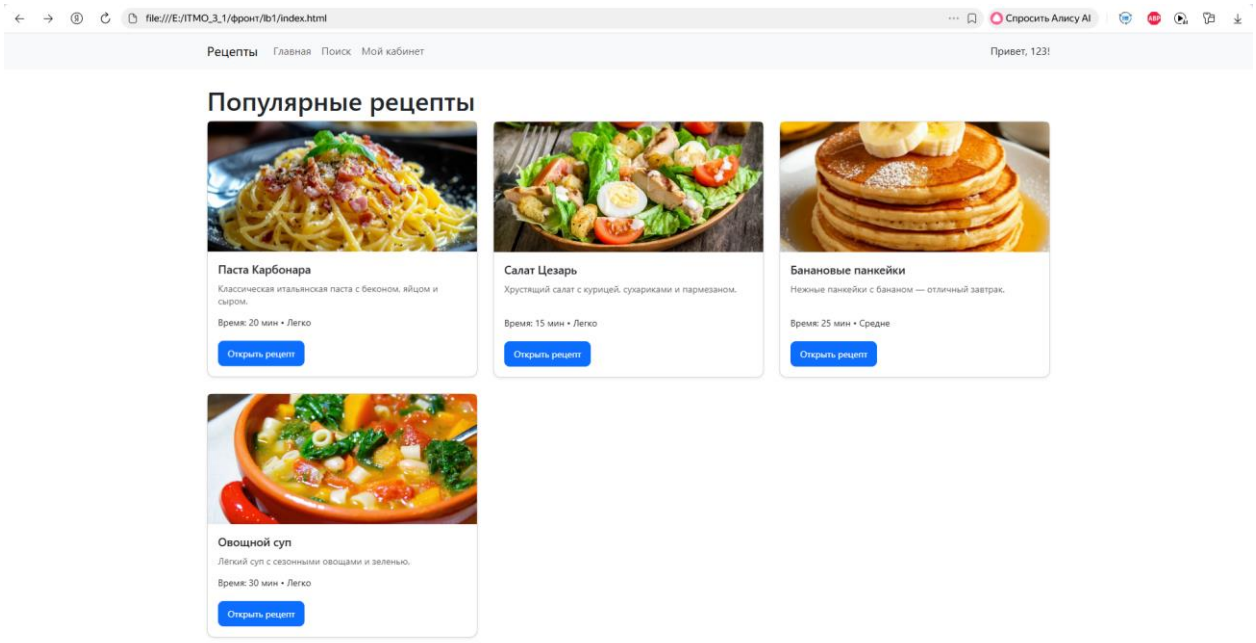

Рисунок 3 – Главная страница

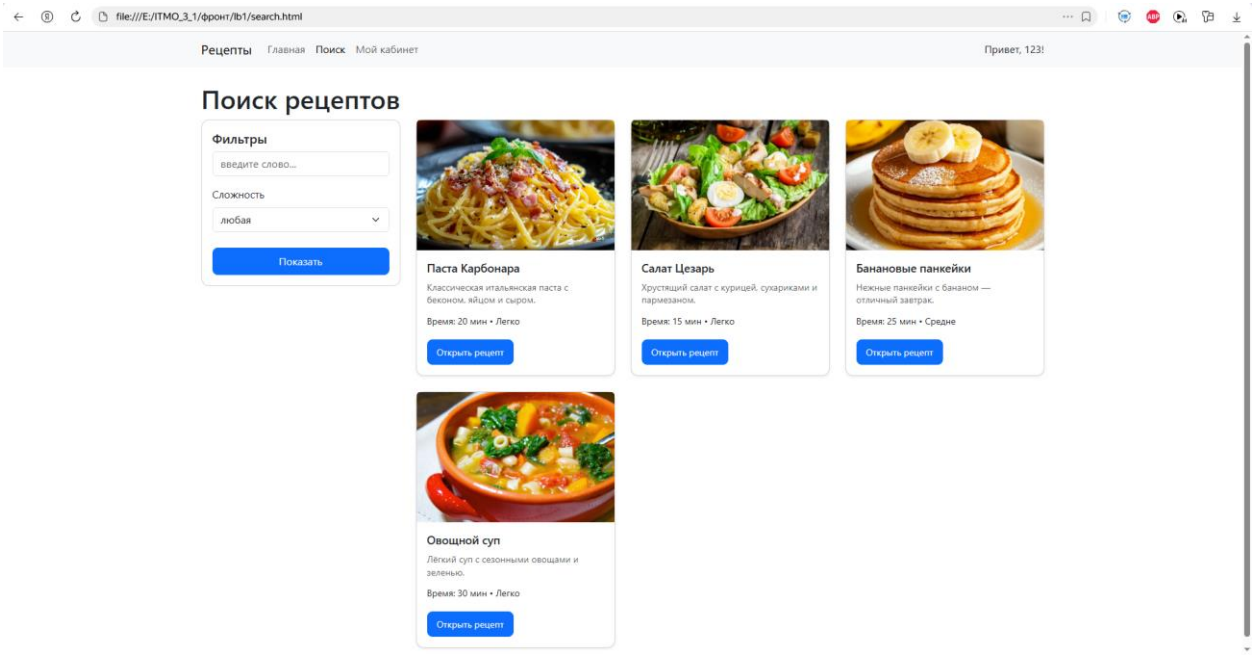

Рисунок 4 – Поиск

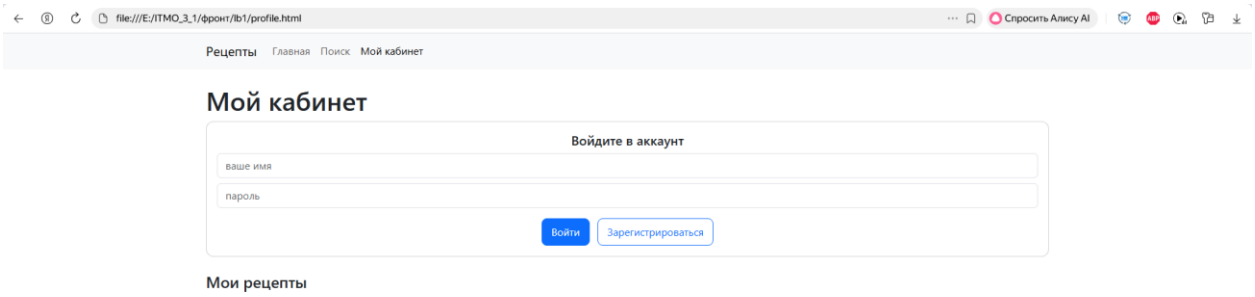

Рисунок 5 – работа профиля (работа в не аккаунта)

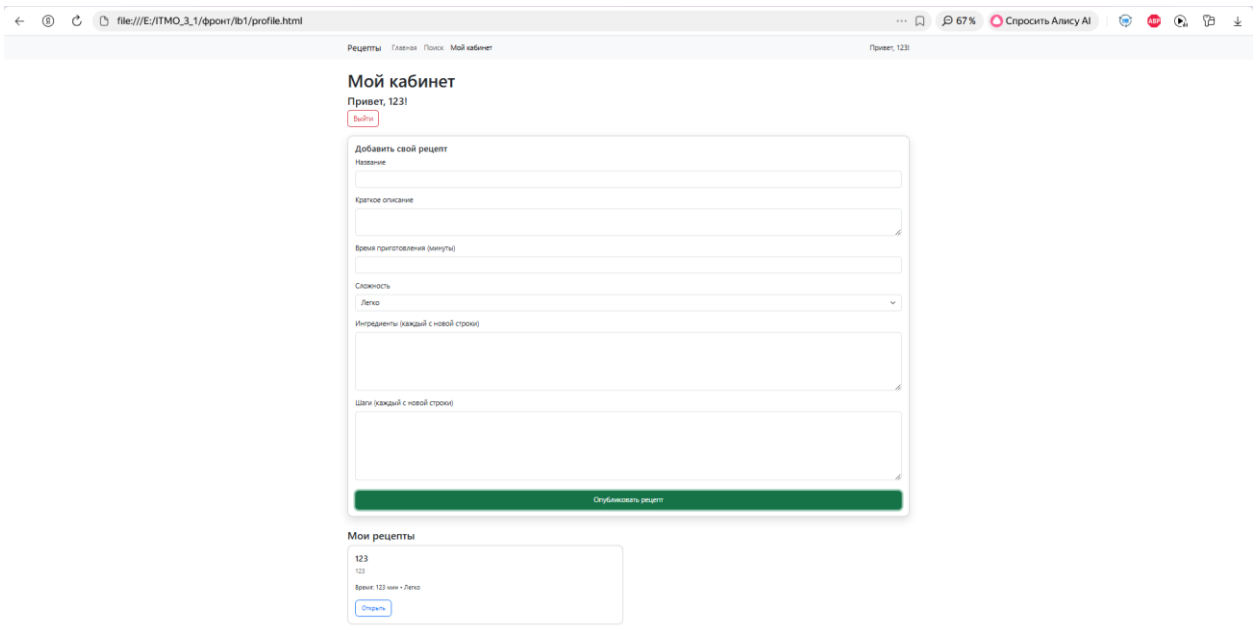

Рисунок 6 – работа профиля (работа в аккаунте)

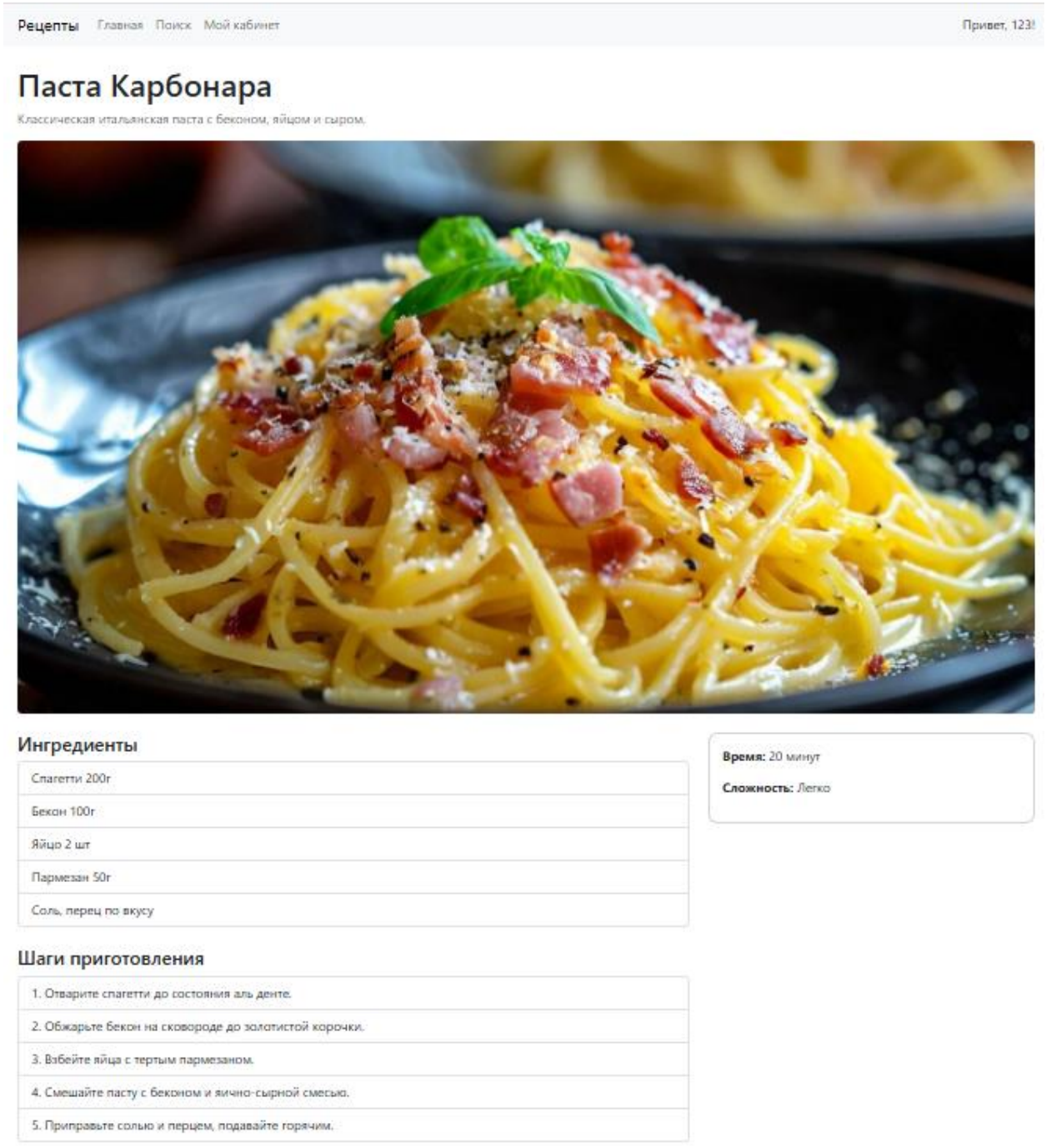

Рисунок 7 – работа страницы рецепта

ВЫВОД

В ходе выполнения лабораторной работы было разработано клиентское веб-приложение для работы с рецептами с использованием HTML, CSS и JavaScript. Приложение позволяет просматривать список рецептов, осуществлять поиск, добавлять собственные рецепты и переходить на страницу отдельного рецепта.

В процессе переделки работы основное внимание было уделено упрощению структуры проекта и логики JS. Часть функциональности была удалена, что позволило лучше разобраться в принципах работы с DOM-элементами, обработкой событий и хранением данных в LocalStorage.

В результате проделанной работы стало понятнее, как реализуется динамическое формирование HTML-контента, передача параметров через URL и организация взаимодействия между несколькими страницами без использования серверной части.