ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

**«ТОМСКИЙ ЭКОНОМИКО-ПРОМЫШЛЕННЫЙ КОЛЛЕДЖ»**

К ЗАЩИТЕ ДОПУЩЕН

Директор ОГБПОУ «ТЭПК»

\_\_\_\_\_\_\_\_\_\_\_\_ / О.Н. Пояркова

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА ОНЛАЙН ВЕБ-ПРИЛОЖЕНИЕ “РЕЦЕПТЫ БЛЮД”**

ДИПЛОМНЫЙ ПРОЕКТ

*СПЕЦИАЛЬНОСТЬ 09.02.07 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ПРОГРАММИРОВАНИЕ*

Выполнил:

Студент IV курса группы 1912с

\_\_\_\_\_\_\_\_\_ / Николайчук Я.Ю

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Проверил:

Руководитель ДП

\_\_\_\_\_\_\_\_\_ / Летягина И.Е.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025 г.

Томск 2025

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc201191280)

[**ТЕХНОЛОГИИ И ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ** 4](#_Toc201191281)

[**Языки и фреймворки для разработки** 4](#_Toc201191282)

[**Программное обеспечение** 4](#_Toc201191283)

[**ТЕХНИЧЕСКОЕ ЗАДАНИЕ** 5](#_Toc201191284)

[**Назначение и структура документа** 5](#_Toc201191285)

[**Ролевая модель** 6](#_Toc201191286)

[**ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ** 7](#_Toc201191287)

[**Концептуальная модель** 7](#_Toc201191288)

[**Логическая модель данных** 10](#_Toc201191289)

[**Физическая модель данных** 11](#_Toc201191290)

[**Словарь данных** 12](#_Toc201191291)

[**Таблица: User (Пользователь)** 12](#_Toc201191292)

[**Таблица: Recipe (Рецепт)** 12](#_Toc201191293)

[**Таблица: Favorite (Избранное)** 13](#_Toc201191294)

[**Таблица: Comment (Комментарий)** 13](#_Toc201191295)

[**Таблица: Role (Роль)** 13](#_Toc201191296)

[**Таблица: Category (Категория)** 13](#_Toc201191297)

[**Таблица: Rating (Рейтинг)** 13](#_Toc201191298)

[**Таблица: CategoryRecipe (Категории-Рецепт)** 13](#_Toc201191299)

[**Таблица: RecipeIngredient (Рецепт-ингредиент)** 13](#_Toc201191300)

[**Таблица: Ingredient (Ингредиент)** 14](#_Toc201191301)

[**Таблица: Unit (Единица измерения)** 14](#_Toc201191302)

[**Диаграмма последовательностей** 14](#_Toc201191303)

[**Диаграмма Прецендентов** 22](#_Toc201191304)

[**ПРОТОТИПЫ И НАВИГАЦИЯ** 24](#_Toc201191305)

[**Структура страниц** 25](#_Toc201191306)

[**Основные прототипы** 27](#_Toc201191307)

[**РЕАЛИЗАЦИЯ БД** 32](#_Toc201191308)

[**РЕАЛИЗАЦИЯ ВЕБ-САЙТА** 35](#_Toc201191309)

[**Основные страницы** 35](#_Toc201191310)

[**ТЕСТИРОВАНИЕ ВЕБ-САЙТА** 57](#_Toc201191311)

[**РУКОВОДСТВО АДМИНИСТРАТОРА** 61](#_Toc201191312)

[**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ** 62](#_Toc201191313)

[**ЗАКЛЮЧЕНИЕ** 63](#_Toc201191314)

[**СПИСОК ЛИТЕРАТУРЫ** 64](#_Toc201191315)

# **ВВЕДЕНИЕ**

Современные кулинарные сайты предоставляют пользователям удобный доступ к огромному количеству рецептов, позволяя не только находить новые блюда, но и делиться своими кулинарными идеями. Однако многие подобные платформы имеют ограниченный функционал, отсутствие персонализации и сложную навигацию. В связи с этим разработка веб-приложения для публикации и обмена рецептами с интуитивно понятным интерфейсом, возможностью оценки, комментирования и сохранения рецептов является актуальной задачей.

**Описание предметной области**

Предметной областью проекта является проектирование и разработка онлайн веб-приложения “рецепты блюд”.

**Цель дипломного проекта**

Проектирование и разработка онлайн веб-приложение “рецепты блюд”.

**Задачи проекта**

* Анализ предметной области.
* Проектирование структуры базы данных и логики приложения.
* Разработка backend-части на Laravel (роутинг, контроллеры, модели, Blade-шаблоны).
* Создание frontend-части с использованием JavaScript для динамических элементов (лайки, избранное, тема).
* Реализация системы авторизации.
* Добавление функционала для администратора (управление категориями и ингредиентами).
* Внедрение темной/светлой темы с сохранением выбора в localStorage.
* Тестирование и отладка приложения.

Проект разрабатывался индивидуально, включая все этапы:

* **Backend-разработка (Laravel):**
  + Проектирование базы данных и миграций
  + Реализация CRUD для рецептов, комментариев, избранного
  + Настройка аутентификации и ролей (пользователь/админ)
  + Динамические элементы без перезагрузки страницы (лайки, добавление в избранное)
  + Реализация переключения темы (интеграция с localStorage)
  + Тестирование и деплой
* **Frontend-разработка (Blade + JS):**
  + Верстка страниц (главная, рецепты, профиль, админ-панель)
  + Валидация форм на стороне клиента
  + Настройка темной/светлой темы (localStorage)
  + Оптимизация UX/UI

# **ТЕХНОЛОГИИ И ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ**

## **Языки и фреймворки для разработки**

**Blade** — шаблонизатор Laravel, применявшийся для серверного рендеринга страниц с возможностью наследования шаблонов, использования директив и вставки PHP-кода непосредственно в разметку.

**JavaScript** — основной язык для клиентской части, отвечавший за динамическое взаимодействие с пользователем, включая обработку событий, AJAX-запросы и работу с LocalStorage для сохранения темы оформления.

**HTML5** — использовался для семантической верстки страниц с акцентом на доступность и адаптивность, обеспечивая четкую структуру контента и корректное отображение во всех браузерах.

**CSS3** — применялся для стилизации интерфейса с использованием современных возможностей, таких как Flexbox, Grid-верстка, CSS-переменные для переключения тем и плавные анимации.

**MySQL** — реляционная база данных, в которой хранилась вся информация о рецептах, пользователях, комментариях и других сущностях проекта, с оптимизированными запросами и продуманной структурой.

**Laravel** — современный PHP-фреймворк, который был использован для разработки серверной части проекта. Он обеспечил быструю и удобную реализацию RESTful API, а также интеграцию с базой данных. Благодаря встроенным механизмам Laravel была реализована безопасная система контроля доступа с учётом ролей пользователей, авторизацией и аутентификацией. Кроме того, фреймворк позволил легко подключать внешние библиотеки и эффективно организовать бизнес-логику проекта.

## **Программное обеспечение**

**PHPStorm** — основная среда разработки, предоставившая мощные инструменты для работы с PHP, Blade и JavaScript, включая автодополнение кода, встроенный отладчик и интеграцию с системами контроля версий.

**Figma** — Профессиональный инструмент для проектирования интерфейсов, использовался для создания дизайн-макетов и прототипирования веб-приложения. Позволил эффективно работать над визуальной частью проекта и согласовывать решения между дизайнерами и разработчиками.

**phpMyAdmin — это** веб-инструмент для управления базами данных MySQL и MariaDB. Предоставляет удобный графический интерфейс для создания, редактирования и управления базами данных, таблицами, запросами и пользователями. Используется для быстрого и наглядного взаимодействия с базой данных проекта, облегчая администрирование и отладку данных.

# **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

## **Назначение и структура документа**

Настоящий документ рассчитан на использование в качестве ключевого документа, определяющего требование по реализации задачи по проектированию и разработке онлайн веб-приложения “рецепты блюд”.

**Структура технического задания:**

1. **Общие положения**
   * Назначение и структура документа
   * Ролевая модель системы
2. **Проектирование системы**
   * Концептуальная модель
   * Логическая модель данных
   * Физическая модель данных
3. **Пользовательский интерфейс**
   * Навигация и структура интерфейса (веб-версия)
     + Главное меню
     + Структура экранов
     + Основные прототипы

## **Ролевая модель**

Ролевая модель — это структура, описывающая роли и отношения между ними в организации, системе или процессе. Она используется для определения обязанностей, прав и взаимодействий различных участников, а также для управления доступом, распределения ответственности и построения рабочих процессов. Ролевая модель позволяет чётко определить, кто и какие действия может выполнять в рамках системы или проекта. (Табл. 1)

Таблица 1. Ролевая модель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Раздел** | **Функционал** | **Админ** | **Пользователь** | **Гость** |
| Навигация | Главная страница | да | да | да |
|  | Раздел блюд | да | да | да |
| Раздел определенного блюда | да | да | да |
| Раздел моих блюд | да | да | нет |
| Профиль | да | да | нет |
|  | Раздел категорий | да | нет | нет |
|  | Раздел ингредиентов | да | нет | нет |
| Авторизация | Вход | да | да | нет |
| Регистрация | нет | нет | да |
| Выход | да | да | нет |
| Блюда | Просмотр | да | да | да |
| Удаление | да | да(свои) | нет |
| Редактирование | да | да(свои) | нет |
| Определенное блюдо | Просмотр | да | да | да |
| Добавление в избранное | да | да | нет |
| Добавление оценки | да | да | нет |
| Создание комментария | да | да | нет |
|  | Редактирование комментария | да | да(свои) | нет |
|  | Удаление комментария | да | да(свои) | нет |
| Профиль | Просмотр профиля | да | да | нет |
| Смена фото | да | да | нет |
|  | Смена пароля | да | да | нет |
|  | Удаление профиля | да | да | нет |
|  | Просмотр избранных блюд | да | да | нет |
| Категории | Просмотр категорий | да | нет | нет |
|  | Добавление категорий | да | нет | нет |
| Ингредиенты | Добавление ингредиентов | да | нет | нет |

# **ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ**

## **Концептуальная модель**

Концептуальная модель — это абстрактное представление структуры данных, отражающее основные сущности предметной области, их атрибуты и взаимосвязи между ними, без учёта технической реализации. (Рис 1.)

На рисунке 1 представлена схема концептуальной модели, наглядно демонстрирующая взаимосвязи между основными сущностями системы. Как видно из схемы:

**1. Пользователь (User)**

Содержит информацию о зарегистрированных пользователях системы, включая их личные данные, роль и учетные данные для входа.

Поля:

* Идентификатор пользователя – уникальный числовой идентификатор (PK).
* Пароль – хэш пароля для аутентификации.
* Имя – имя пользователя.
* Электронная почта – контактный e-mail, используется для входа (уникальное значение).
* Дата регистрации – дата регистрации пользователя.
* Аватар – ссылка на изображение профиля (может быть пустой).
* Роль – ссылка на роль пользователя (внешний ключ на таблицу "Роли").
* Избранное - ссылка на избранные блюда пользователя (внешний ключ на таблицу "Избранное").

**2. Роль (Role)**

Содержит список доступных ролей в системе и определяет уровень прав доступа.

Поля**:**

* Идентификатор роли – уникальный числовой идентификатор (PK).
* Название роли – имя роли (например: Администратор, Пользователь).

**3. Рецепт (Recipe)**

Содержит информацию о блюдах системы, включая информацию по составу, способ приготовления, сложность, изображение.

Поля:

* Идентификатор рецепта – уникальный числовой идентификатор (PK).
* Описание – словесное описание блюда.
* Время готовки – количество времени для приготовления блюда.
* Сложность – уровень сложности для конкретного блюда.
* Калорийность – примерное количество калорий в блюде.
* Изображение – ссылка на изображение блюда.
* Название – название блюда.
* Дата публикации – дата публикации блюда.
* Способ приготовления – словесный способ приготовления блюда.

**4. Категории-рецепты (Category-recipe)**

Служит для связки и хранения данные таблиц Категория и Рецепт.

Поля:

* Идентификатор категории-рецепты – уникальный числовой идентификатор (PK).

**5. Категории (Category)**

Содержит информацию о категориях, их название.

Поля**:**

* Идентификатор роли – уникальный числовой идентификатор (PK).
* Название категории– название категории (например: Острое, Сладкое).

**6. Рецепт-Ингредиент (Recipe-Ingredient)**

Служит для связи и хранения данных таблиц Ингредиент, Единица измерения, Рецепт.

Поля**:**

* Идентификатор роли – уникальный числовой идентификатор (PK).
* Количество - количество определенного ингредиента.

**7. Ингредиент (Ingredient)**

Содержит информацию о всех ингредиентах.

Поля**:**

* Идентификатор роли – уникальный числовой идентификатор (PK).
* Название - название определенного ингредиента.

**8. Единица измерения (Unit)**

Содержит информацию о единицах измерений, нужные для ингредиентов.

Поля**:**

* Идентификатор роли – уникальный числовой идентификатор (PK).
* Название - краткое название единицы измерения.
* Полное название – полное название единицы измерения.

**9. Избранное (Favorite)**

Содержит информацию о избранные блюд выбранным авторизированным пользователем.

Поля**:**

* Идентификатор роли – уникальный числовой идентификатор (PK).

**10. Оценка (Rating)**

Содержит информацию о оценке блюда.

Поля**:**

* Идентификатор роли – уникальный числовой идентификатор (PK).
* Оценка – оценка, выставленная авторизированными пользователями, считается средний балл.

**11. Комментарий (Comment)**

Содержит информацию о комментариях для каждого блюда.

Поля**:**

* Идентификатор роли – уникальный числовой идентификатор (PK).
* Текст – сам текст комментария, который написал авторизированный пользователь.
* Дата – дата создания комментария.

Концептуальная модель включает 11 основных сущностей, полностью покрывающих функциональные требования системы.

Модель обеспечивает основу для последующих этапов проектирования БД и разработки сайта.

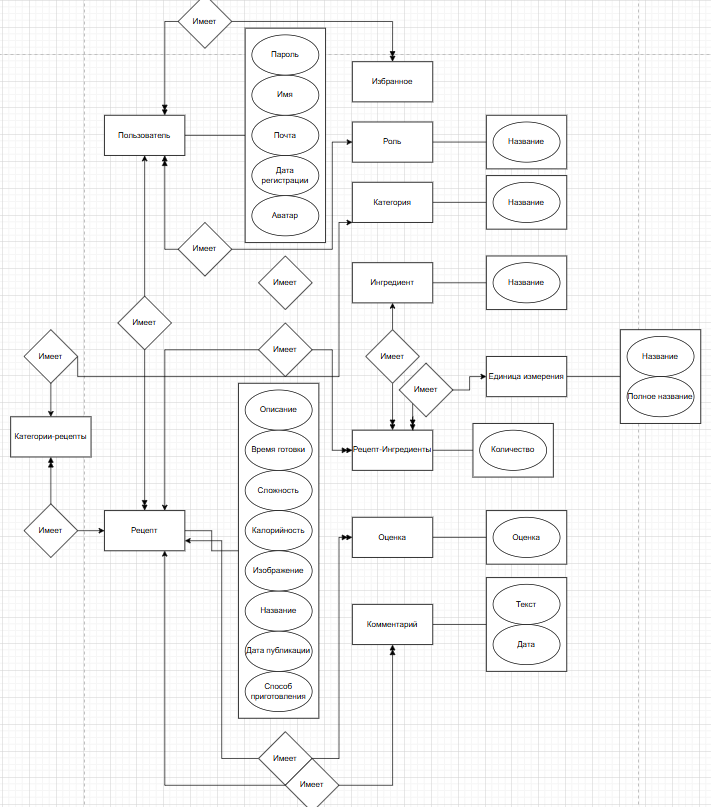


Рисунок 1 - Концептуальная модель

## **Логическая модель данных**

Логическая модель данных — это структурированное представление данных, описывающее сущности, их атрибуты, типы данных и связи между сущностями с учётом логики взаимодействия, но без привязки к конкретной системе управления базами данных.

На основании концептуальной модели была разработана логическая модель данных, которая включает:

* Точные типы данных для каждого атрибута.
* Подробные характеристики связей (кардинальность, обязательность).
* Нормализованную структуру таблиц.  
  Ниже представлены скриншоты получившейся модели (Рис. 2).

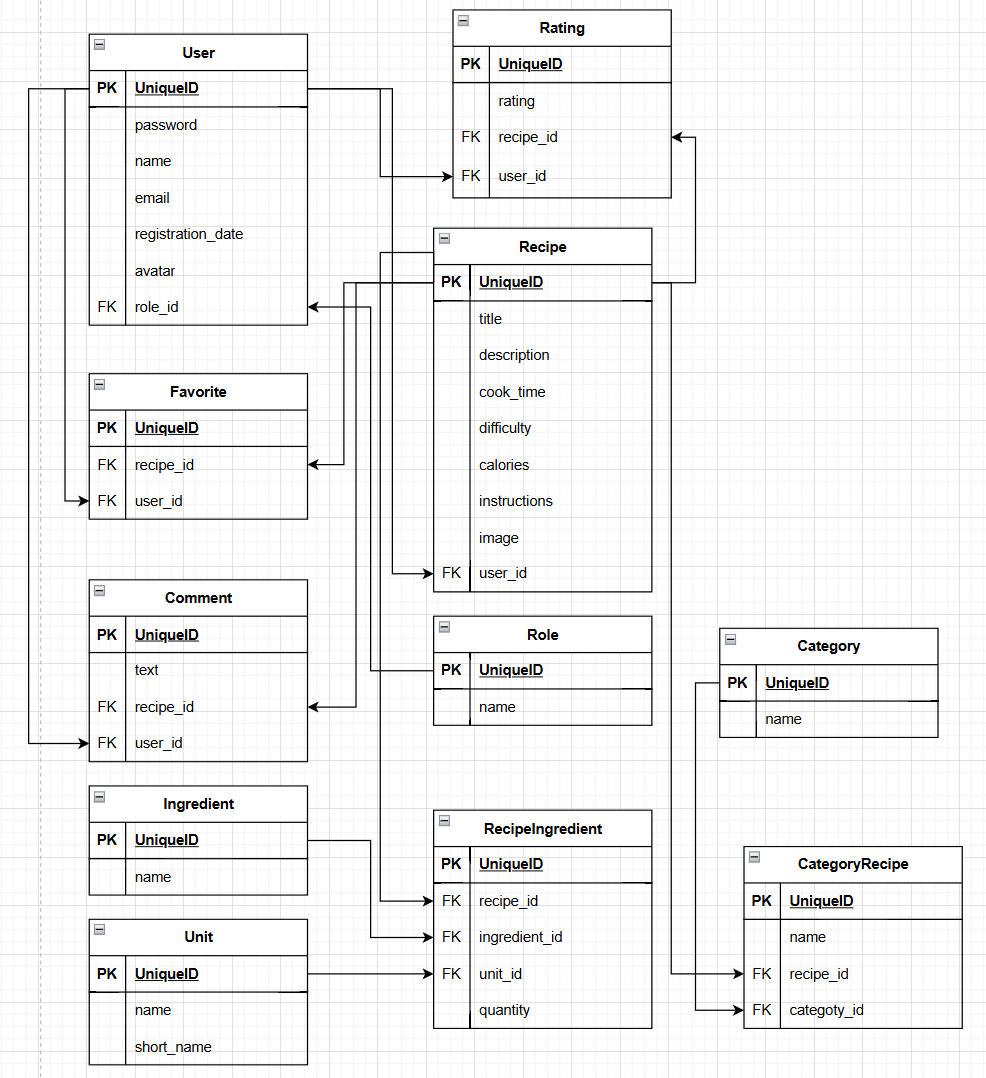


Рисунок 2 - Логическая модель

## **Физическая модель данных**

Физическая модель данных — это модель данных, описанная с помощью средств конкретной системы управления базами данных (СУБД). Физическая модель данных строится на базе логической путем добавления особенностей конкретной СУБД. К таким особенностям могут относиться поддерживаемые СУБД типы данных, соглашения о присвоении имен таблицам, атрибутам и т.д. Физическая модель данных фактически является готовым заданием на создание БД, имея которое можно реализовать БД в выбранной СУБД.

Выбранной СУБД является MySQL. Используя логическую модель, построим физическую модель данных (См Рис 3):

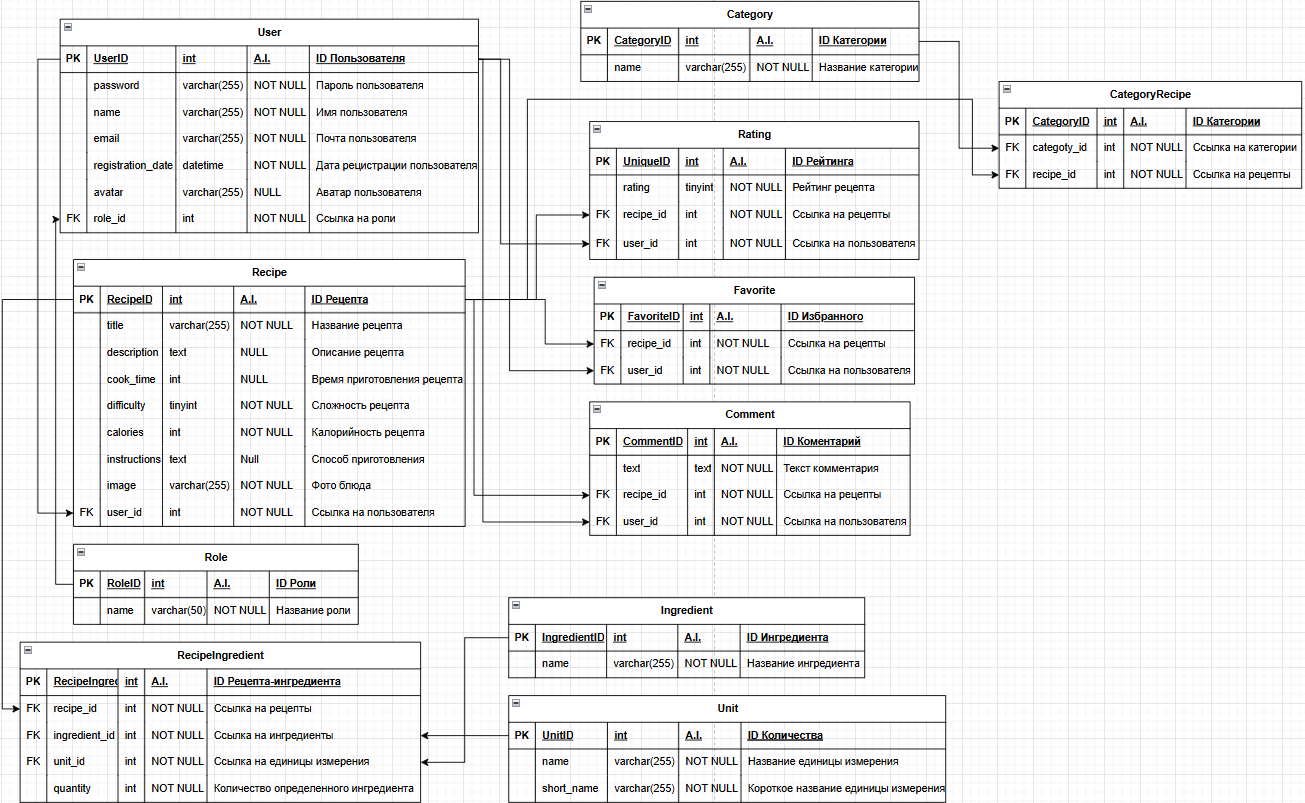


Рисунок 3 - Физическая модель

## **Словарь данных**

**Словарь данных** — это метаданные, которые описывают структуру, атрибуты и свойства объектов базы данных (таблиц, столбцов, индексов, ограничений и т. д.). Он содержит информацию о данных, хранящихся в БД, и используется для управления, документирования и контроля целостности системы.

### **Таблица: User (Пользователь)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор пользователя |
| password | varchar(255) | Обязательное | Пароль пользователя |
| name | varchar(255) | Опционально | Имя пользователя |
| email | varchar(255) | Обязательное | Почта пользователя |
| registration\_date | datetime | Обязательное | Дата регистрации пользователя |
| avatar | varchar(255) | Опционально | Аватар пользователя |
| role\_id | int | FK, Обязательное | Ссылка на роль пользователя |

### **Таблица: Recipe (Рецепт)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор рецепта |
| title | varchar(255) | Обязательное | Название рецепта |
| description | text | Обязательное | Описание рецепта |
| cook\_time | int | Обязательное | Время приготовления рецепта |
| difficulty | tinyint | Обязательное | Сложность рецепта |
| calories | int | Обязательное | Калорийность рецепта |
| instructions | text | Обязательное | Способ приготовления |
| image | varchar(255) | Обязательное | Фото блюда |
| user\_id | int | FK, Обязательное | Ссылка на пользователя |

### **Таблица: Favorite (Избранное)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор избранного |
| recipe\_id | int | FK, Обязательное | Ссылка на рецепт |
| user\_id | int | FK, Обязательное | Ссылка на пользователя |

### **Таблица: Comment (Комментарий)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор комментария |
| text | text | Обязательное | Текст комментария |
| recipe\_id | int | FK, Обязательное | Ссылка на рецепт |
| user\_id | int | FK, Обязательное | Ссылка на пользователя |

### **Таблица: Role (Роль)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор роли |
| name | varchar(50) | Обязательное | Название роли |

### **Таблица: Category (Категория)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор категории |
| name | varchar(255) | Обязательное | Название категории |

### **Таблица: Rating (Рейтинг)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор рейтинга |
| rating | tinyint | Обязательное | Рейтинг рецепта |
| recipe\_id | int | FK, Обязательное | Ссылка на рецепт |
| user\_id | int | FK, Обязательное | Ссылка на пользователя |

### **Таблица: CategoryRecipe (Категории-Рецепт)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор категории-рецепт |
| category\_id | int | FK, Обязательное | Ссылка на категорию |
| recipe\_id | int | FK, Обязательное | Ссылка на рецепт |

### **Таблица: RecipeIngredient (Рецепт-ингредиент)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор рецепт-ингредиент |
| recipe\_id | int | FK, Обязательное | Ссылка на рецепты |
| recipe\_id | int | FK, Обязательное | Ссылка на ингредиенты |
| user\_id | int | FK, Обязательное | Ссылка на единицы измерения |
| quantity | int | Обязательное | Количество определенного ингредиента |

### **Таблица: Ingredient (Ингредиент)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор ингредиента |
| name | varchar(255) | Обязательное | Название ингредиента |

### **Таблица: Unit (Единица измерения)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Атрибут** | **Тип** | **Ограничение** | **Описание** |
| id | int | PK, авто инкремент | Идентификатор единицы измерения |
| name | varchar(255) | Обязательное | Название единицы измерения |
| short\_name | varchar(255) | Обязательное | Короткое название единицы измерения |

## **Диаграмма последовательностей**

**Диаграмма последовательностей (Sequence Diagram)** — это один из видов диаграмм в языке моделирования UML (Unified Modeling Language), который визуализирует взаимодействие между объектами (или компонентами, актерами) в рамках одного конкретного сценария, с акцентом на порядок и время передачи сообщений.

Авторизация



Рисунок 4. Авторизация

Регистрация



Рисунок 5. Регистрация

Просмотр списка блюд

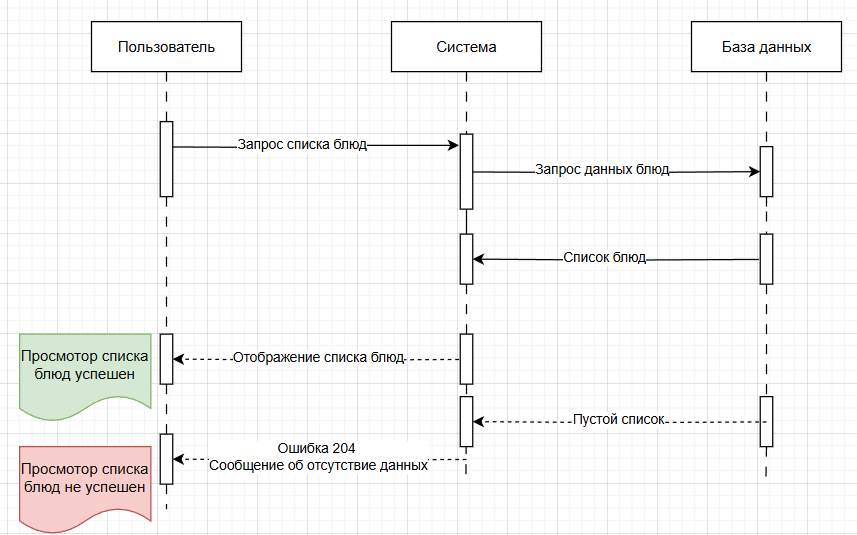


Рисунок 6. Просмотр списка блюд

Просмотр полной информации

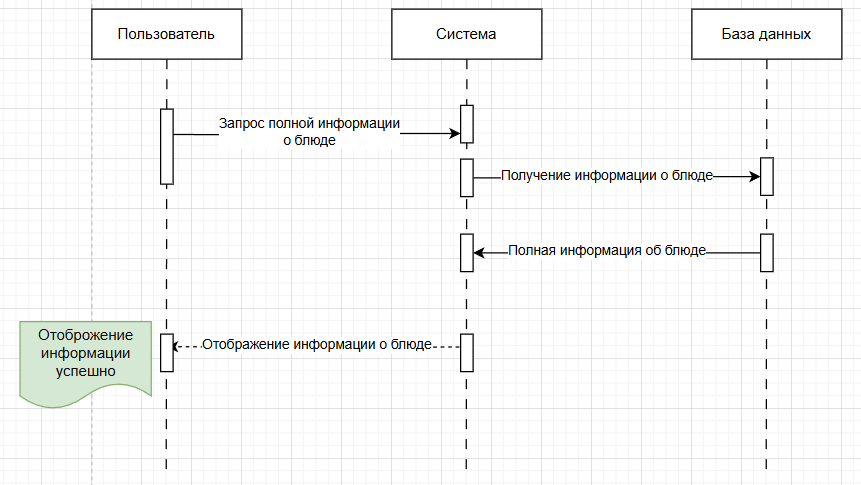


Рисунок 7. Просмотр полной информации

Выставление оценки блюду



Рисунок 8. Выставление оценки блюду

Написание комментария блюду

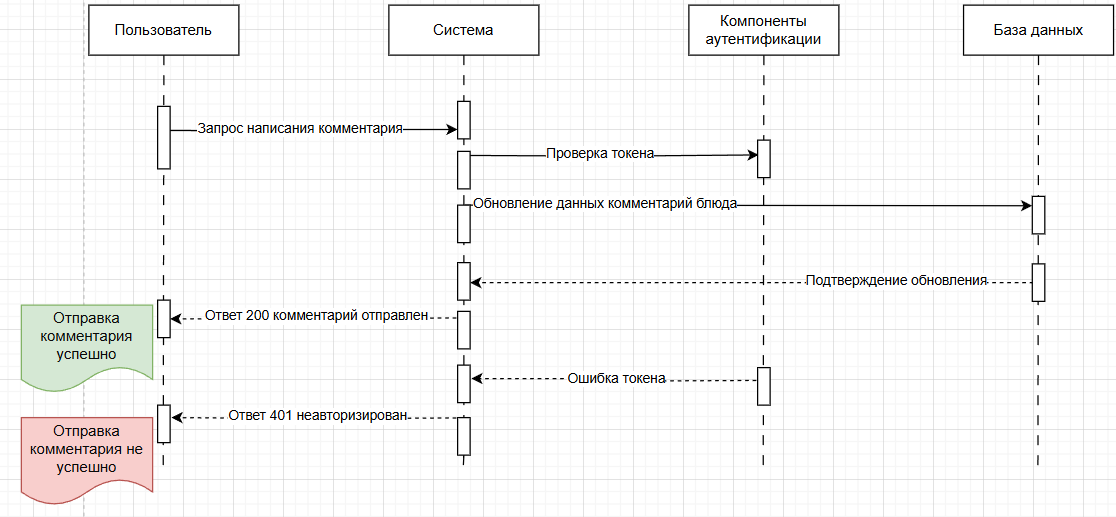


Рисунок 9. Написание комментария блюду

Добавление блюда в избранное

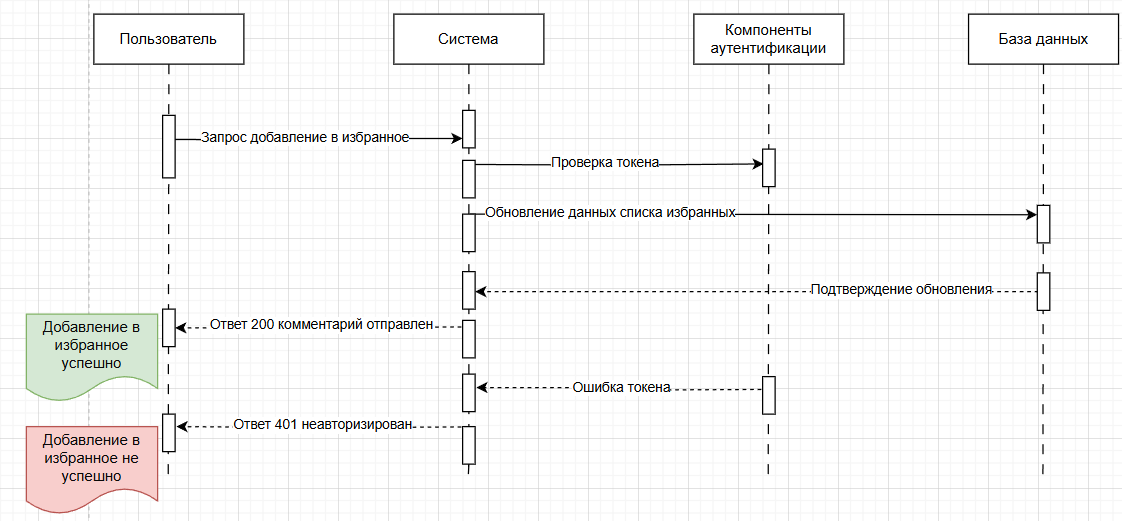


Рисунок 10. Добавление блюда в избранное

Просмотр личного кабинета

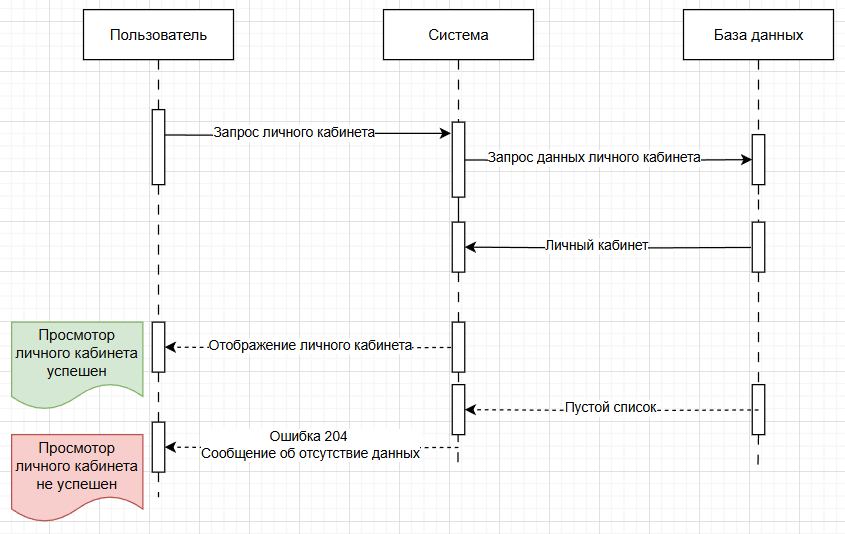


Рисунок 11. Просмотр личного кабинета

Изменение пароля пользователя

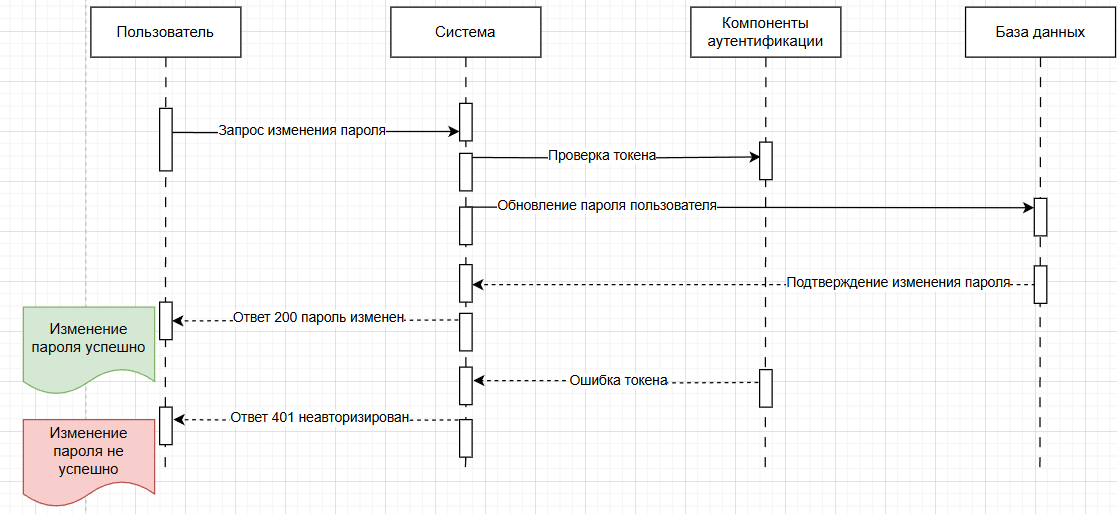


Рисунок 12. Изменение пароля пользователя

Просмотр избранных рецептов



Рисунок 13. Просмотр избранных рецептов

Добавление нового блюда

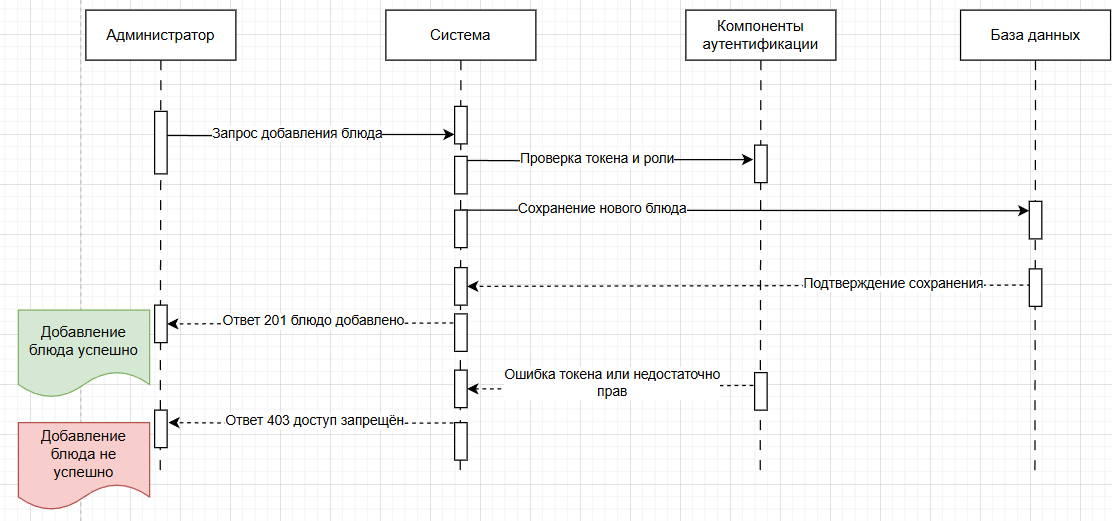


Рисунок 14. Добавление нового блюда

Удаление блюда

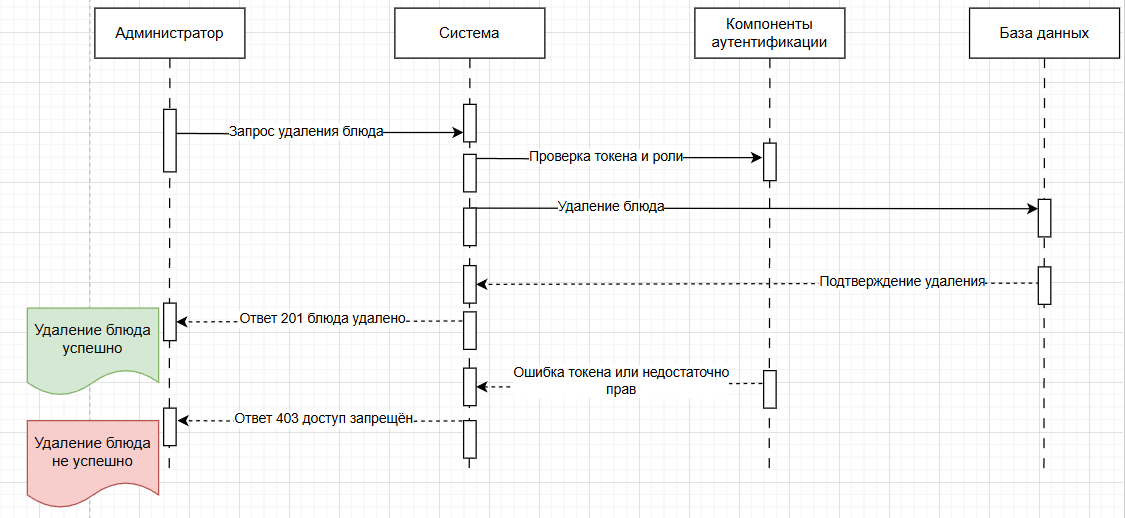


Рисунок 15. Удаление блюд

Редактирование блюда

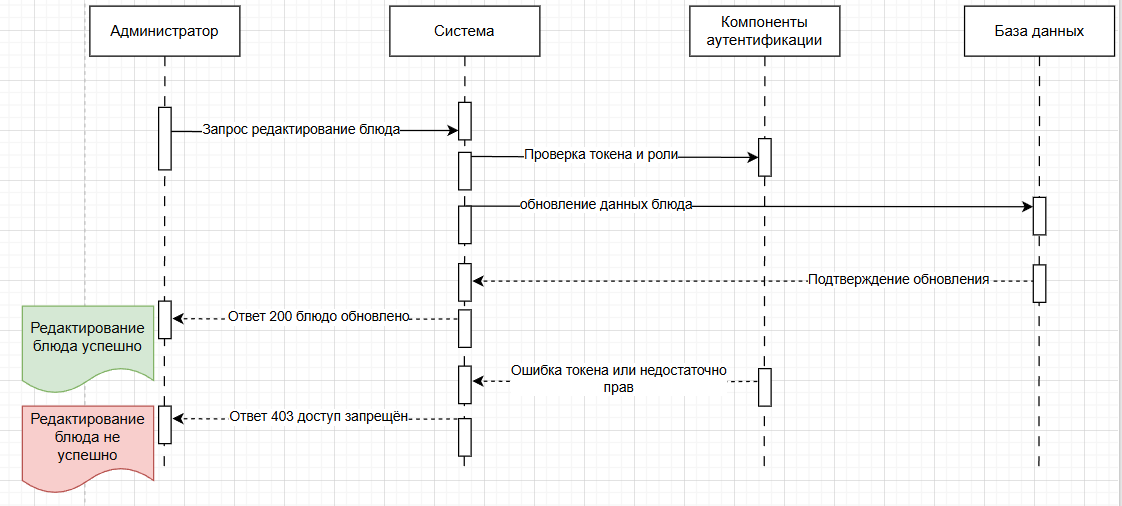


Рисунок 16. Редактирование блюда

Удаление комментария

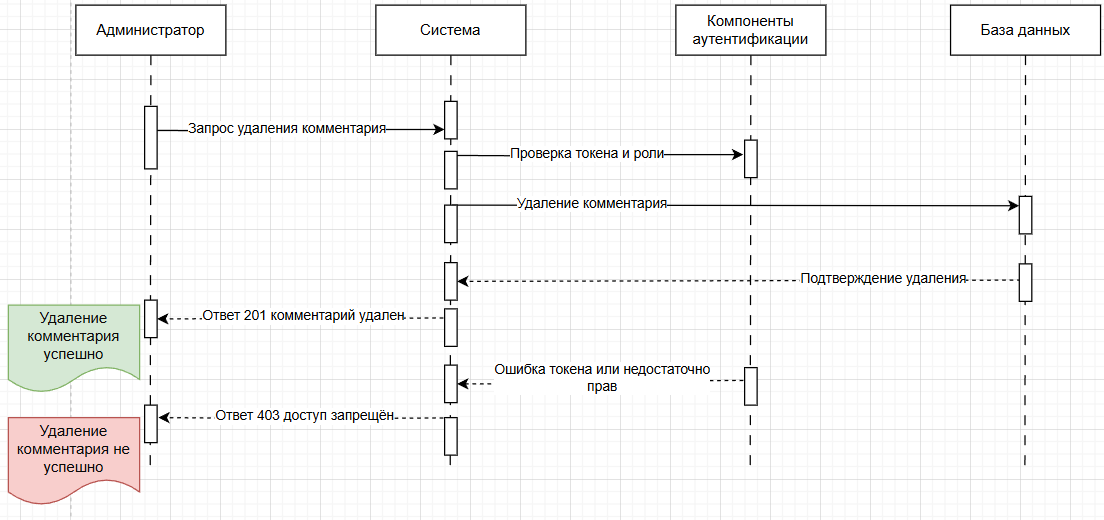


Рисунок 17. Удаление комментария

Редактирование комментария

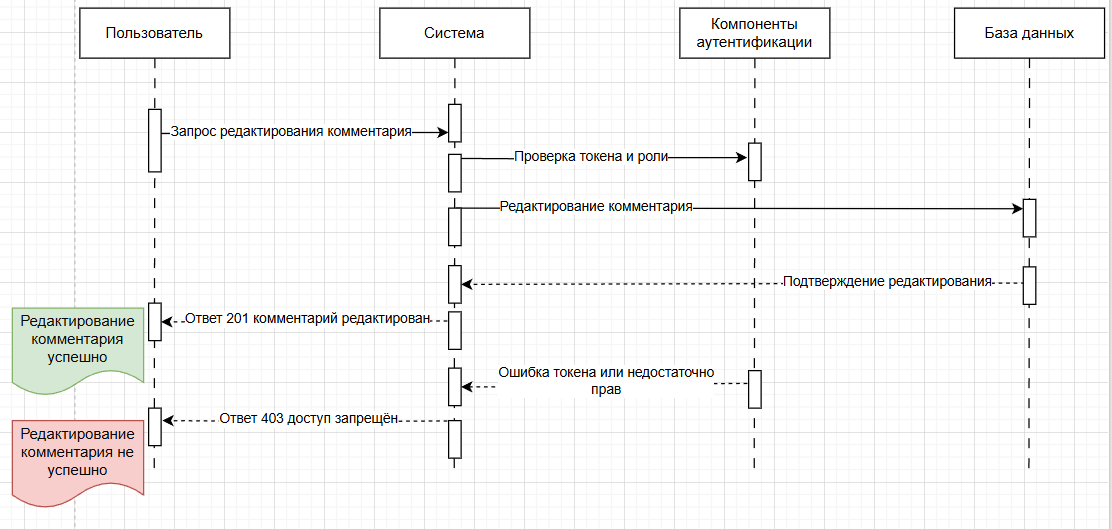


Рисунок 18. Редактирование комментария

## **Диаграмма Прецендентов**

**Диаграмма прецедентов (Use Case Diagram)** — это один из видов диаграмм UML, который визуализирует взаимодействие между системой и её внешними пользователями (актерами) через набор функциональных возможностей (прецедентов).

Модуль пользователи

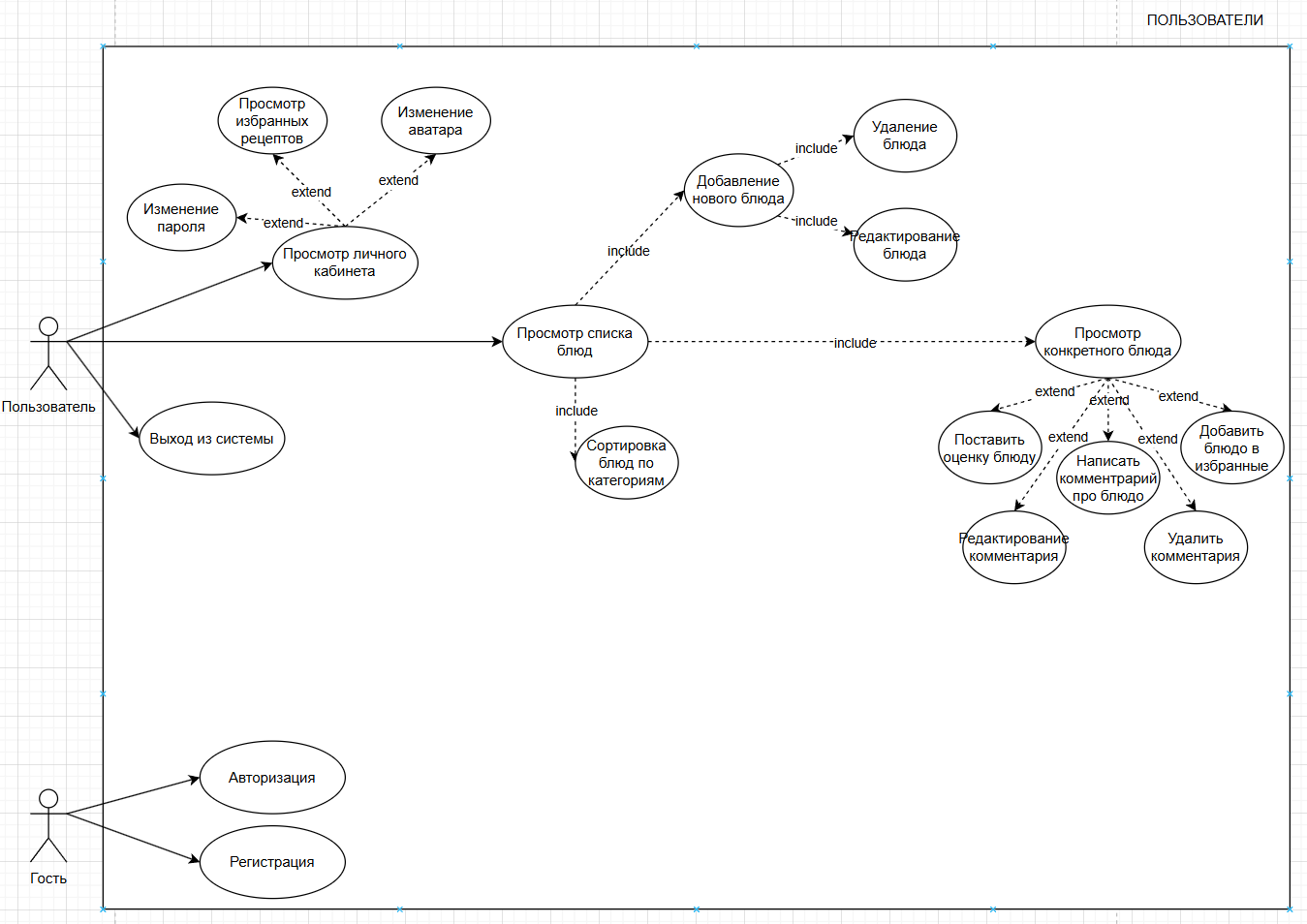


Рисунок 19. Модуль пользователя

Модуль Пользователь отвечает за все возможности авторизированного пользователя на сайте рецепты блюд.

Функциональность модуля:

* Вход в систему.
* Просмотр и управление личным кабинетом.
* Просмотр и функционал блюд и его подробной версии.
* CRUD-операции над рецептами.

Модуль Администратора

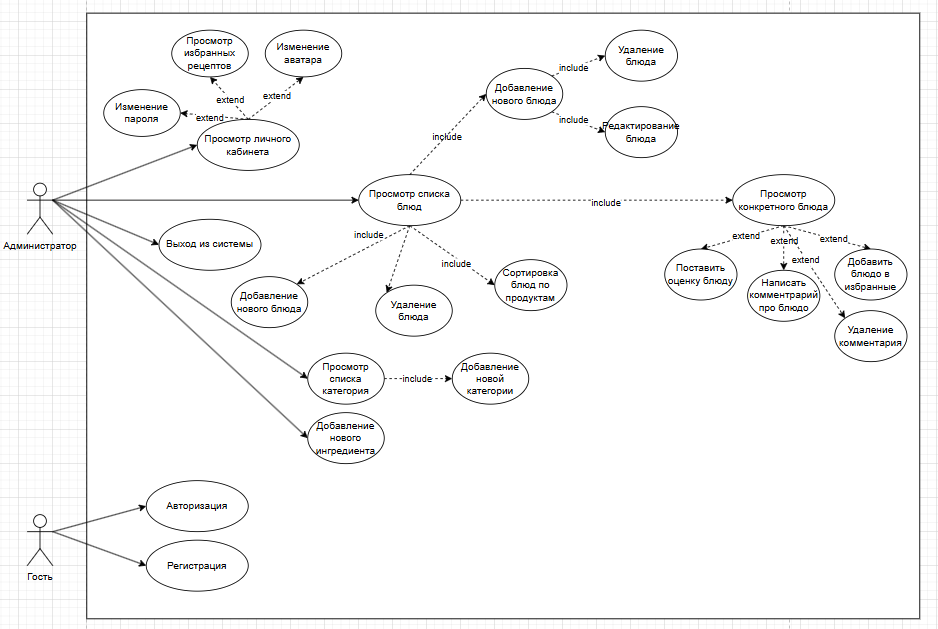


Рисунок 20. Модуль администратора

Модуль Администратора отвечает за все возможности авторизированного пользователя на сайте рецепты блюд, а также скрытые функции администратора.

Функционал модуля:

* Все функции пользователя.
* Просмотр, создание категорий.
* Создание ингредиентов.

# **ПРОТОТИПЫ И НАВИГАЦИЯ**

Раздел посвящен проектированию пользовательского интерфейса и навигационной логики кулинарного портала. Сайт реализован на Laravel с использованием Blade-шаблонов. Основываясь на пользовательских сценариях и задачах, была реализована следующая структура:

* Современный, лаконичный и визуально привлекательный интерфейс
* Удобная и логичная навигация
* Адаптивная верстка для различных устройств
* Гибкий доступ к функциям в зависимости от роли пользователя

Особое внимание уделено:

* Визуальной иерархии и удобству восприятия информации
* Согласованной цветовой палитре и доступности интерфейса
* Простоте взаимодействия с карточками рецептов и навигацией

**Навигация и структура интерфейса**

Кулинарный портал предоставляет пользователю быстрый доступ к подборкам рецептов, разбитым по категориям (например, *Завтраки*, *Салаты*), с возможностью перейти к подробному описанию каждого блюда.

**Главное меню**

Меню навигации размещается в верхней части экрана и отображается в зависимости от роли пользователя:

**1. Гость**

* Главная
* Рецепты
* Вход
* Регистрация
* Смена цветной темы

**2. Авторизованный пользователь**

* Главная
* Рецепты
* Профиль
* Выйти
* Смена цветной темы

**3. Администратор**

* Главная
* Рецепты
* Профиль
* Категории
* Ингредиенты
* Выйти
* Смена цветной темы

## **Структура страниц**

Данный раздел описывает ключевые экраны системы и их функциональные элементы. Интерфейс разработан с учетом:

* Ролевой модели (разный набор функций для гостей, организаторов и администраторов)
* Принципов юзабилити (интуитивная навигация, минималистичный дизайн, адаптивность)

Основные экраны системы включают:

1. **Главная страница**

* Краткое описание сайта.
* Список разных категорий с блюдами.
* Кнопка подробнее на каждой карточке с блюдом.

1. **Страница авторизации**

* Авторизация форма входа с полями для ввода email и пароля.
* Кнопка войти.

1. **Страница регистрации**

* Регистрация с полями имя, почта, пароль, подтверждение пароля, выбор аватара.
* Кнопка зарегистрироваться.

1. **Страница рецепты**

* Список всех существующих рецептов (ограничение 9 рецептов) с краткой информацией о блюде, его оценка, время приготовления, кнопка подробнее для просмотра полной информации о рецепте.
* Поисковая строка рецептов.
* Выбор категории.
* Кнопка применить фильтры.
* Кнопка сброса фильтров.
* Нижнее навигационное меню для просмотра других рецептов.
* Кнопка Мои рецепты, доступная только авторизированным пользователям и администратору.

1. **Мои рецепты** (для пользователей и администратора)

* Список созданных рецептов
* Управление рецептами (редактирование, удаление)
* Кнопка Добавить рецепт для добавления рецепта.

1. **Добавить рецепт** (для пользователей и администратора)

* Форма заполнения на добавление рецепта. Включает в себя: название, описание, время приготовления, калорийность, сложность, ингредиенты, категории, инструкция по приготовлению, изображение.
* Кнопка создать рецепт, которая создает рецепт при правильном заполнении формы.

1. **Профиль пользователя (пользователь и администратор)**

* Основные данные пользователя (никнейм, email, дата регистрации).
* Аватар пользователя.
* Список избранных рецептов пользователем.
* Кнопка редактировать профиль и удаления аккаунта.

1. **Редактировать рецепт (пользователь и администратор)**

* Форма заполнения на редактирование рецепта. Включает в себя: название, описание, время приготовления, калорийность, сложность, ингредиенты, категории, инструкция по приготовлению, изображение.
* Кнопка Обновить рецепт, которая обновляет рецепт при правильной заполнении формы.

1. **Редактирование профиля** (**пользователь и администратор)**

* Форма заполнения на смену данных пользователя, поля имя, почта, аватар.
* Кнопка Сохранить изменения, работает при корректном заполнении формы.
* Кнопка Сменить пароль.

1. **Смена пароля** (**пользователь и администратор)**

* Форма заполнения на смену пароля пользователя, поля новый пароль, подтверждение пароля.
* Кнопка Сменить пароль.
* Кнопка Назад к профилю.

1. **Подробное описание рецепта**

* Страница с полным описанием рецепта: его картинка, время приготовления, калорийность, сложность приготовления, название, оценка, краткое описание, список ингредиентов, способ приготовления.
* Возможность оценить рецепт по пятибалльной системе.
* Возможность оставить комментарий о рецепте.
* Отображение комментариев с возможностью его редактировать и удалять.

1. **Категории** (**администратор)**

* Список всех категорий, при нажатии на которых переносит на просмотр всех блюд их этой категории.
* Кнопка Добавить категорию.

1. **Добавить категории** (**администратор)**

* Форма заполнения для добавления категории с полем Название категории.
* Кнопка добавить категорию.

1. **Выбранная категории** (**администратор)**

* Список всех блюд из данной категории.
* Кнопка Назад.
* Кнопка подробнее.

1. **Ингредиенты** (**администратор)**

* Форма на создание нового ингредиента с полем Название ингредиента.
* Кнопка Добавить ингредиент.

## **Основные прототипы**

Главная - страница является основным входом в систему для всех пользователей. На ней размещена приветственная надпись и подборка рецептов, разбитая по категориям. Страница предоставляет общее представление о проекте и позволяет сразу перейти к просмотру блюд без необходимости авторизации.

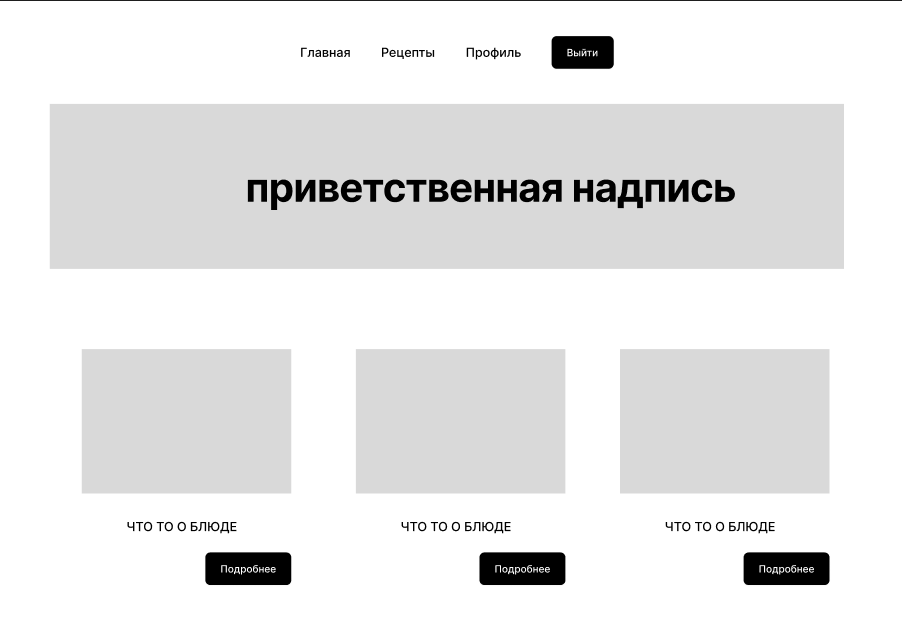


Рисунок 21. Главная страница прототип

Страница с рецептами - отображает все доступные блюда в виде карточек с изображением, названием и краткой информацией. Пользователь может воспользоваться поиском и фильтрами по категориям. При клике на рецепт открывается подробная страница с описанием, ингредиентами, шагами приготовления и возможностью оставить комментарий или оценку.

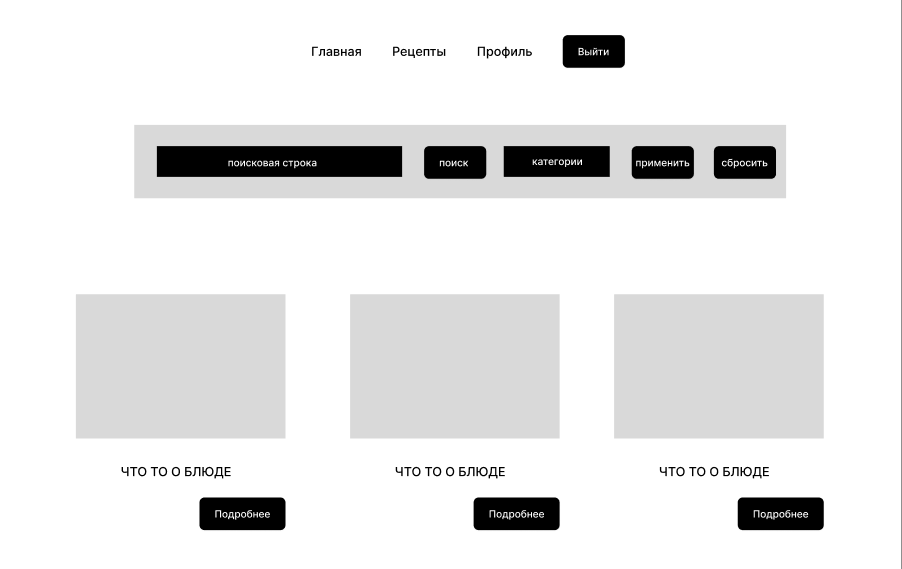


Рисунок 22. Страница блюд прототип

Страница подробного рецепта - содержит полную информацию о выбранном блюде: изображение, описание, список ингредиентов и пошаговую инструкцию приготовления. Также отображаются время готовки, калорийность и сложность. Зарегистрированные пользователи могут оставить комментарий, поставить оценку и добавить рецепт в избранное.

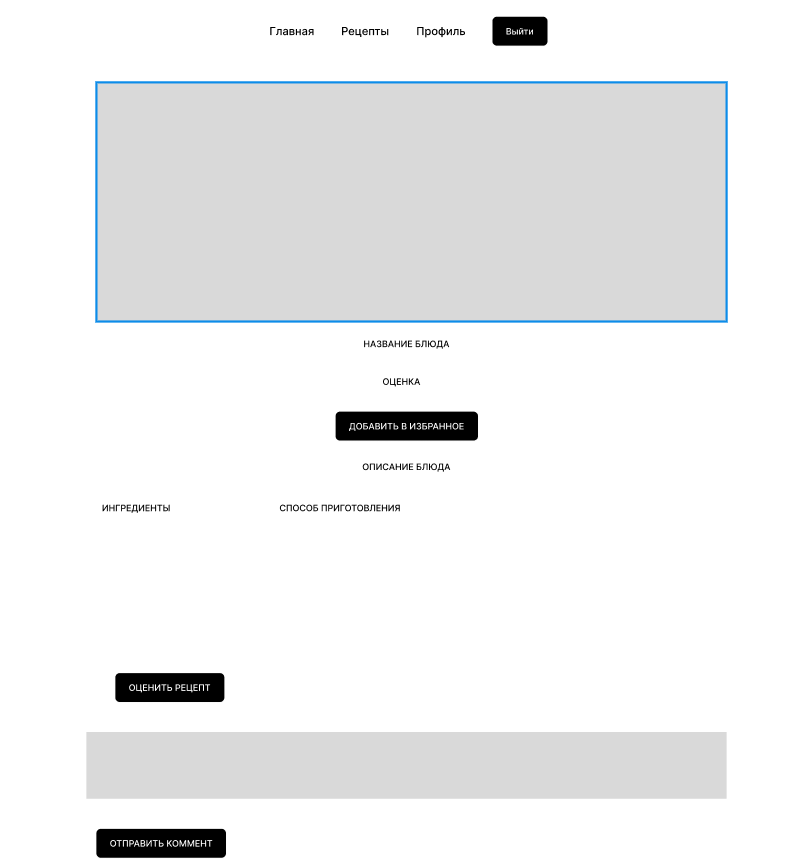


Рисунок 23. Подробное описание прототип

Страница профиля - отображает основную информацию о пользователе: имя, email и аватар. Здесь также представлен список добавленных рецептов и избранных блюд. Пользователь может изменить свои данные, обновить пароль или удалить учетную запись.

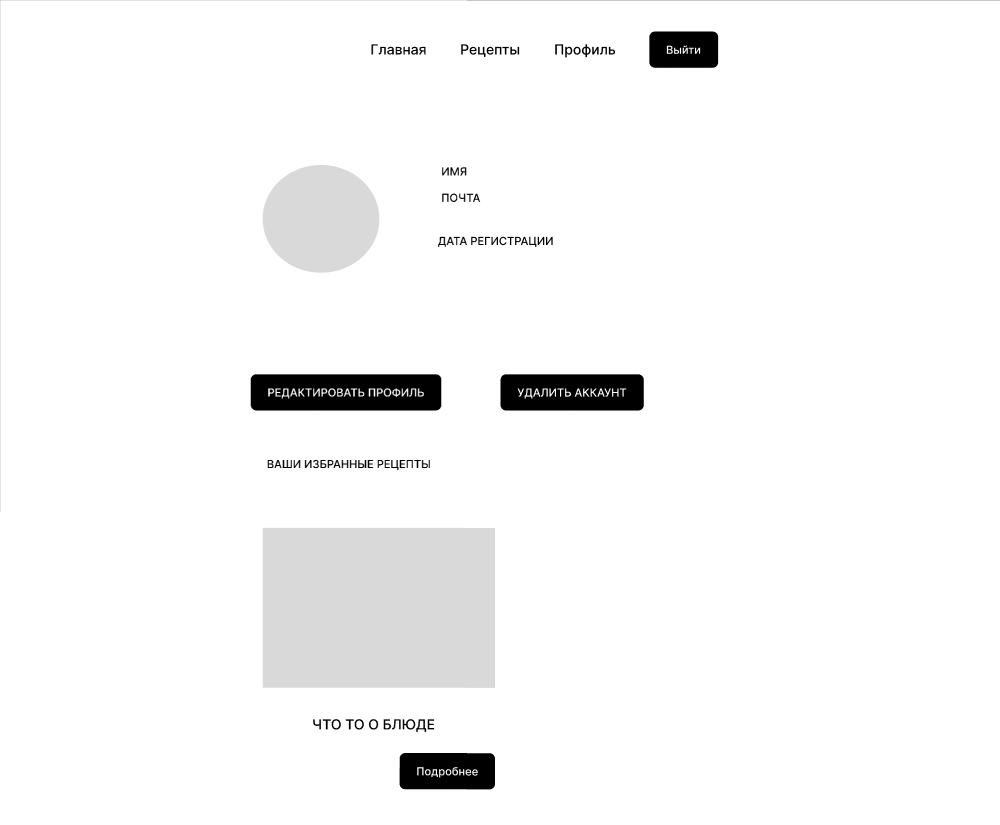


Рисунок 24. Профиль прототип

Страница создания рецепта – отображает форму заполнения для создания рецепта: название, описание, время приготовления, калорийность, сложность, ингредиенты, категории, инструкции и изображение.



Рисунок 25. Создание рецепта прототип

# **РЕАЛИЗАЦИЯ БД**

База данных спроектирована на основе физической модели и реализована в **phpMyAdmin**. Она включает все необходимые таблицы для хранения информации о пользователях, рецептах, категориях, ингредиентах.

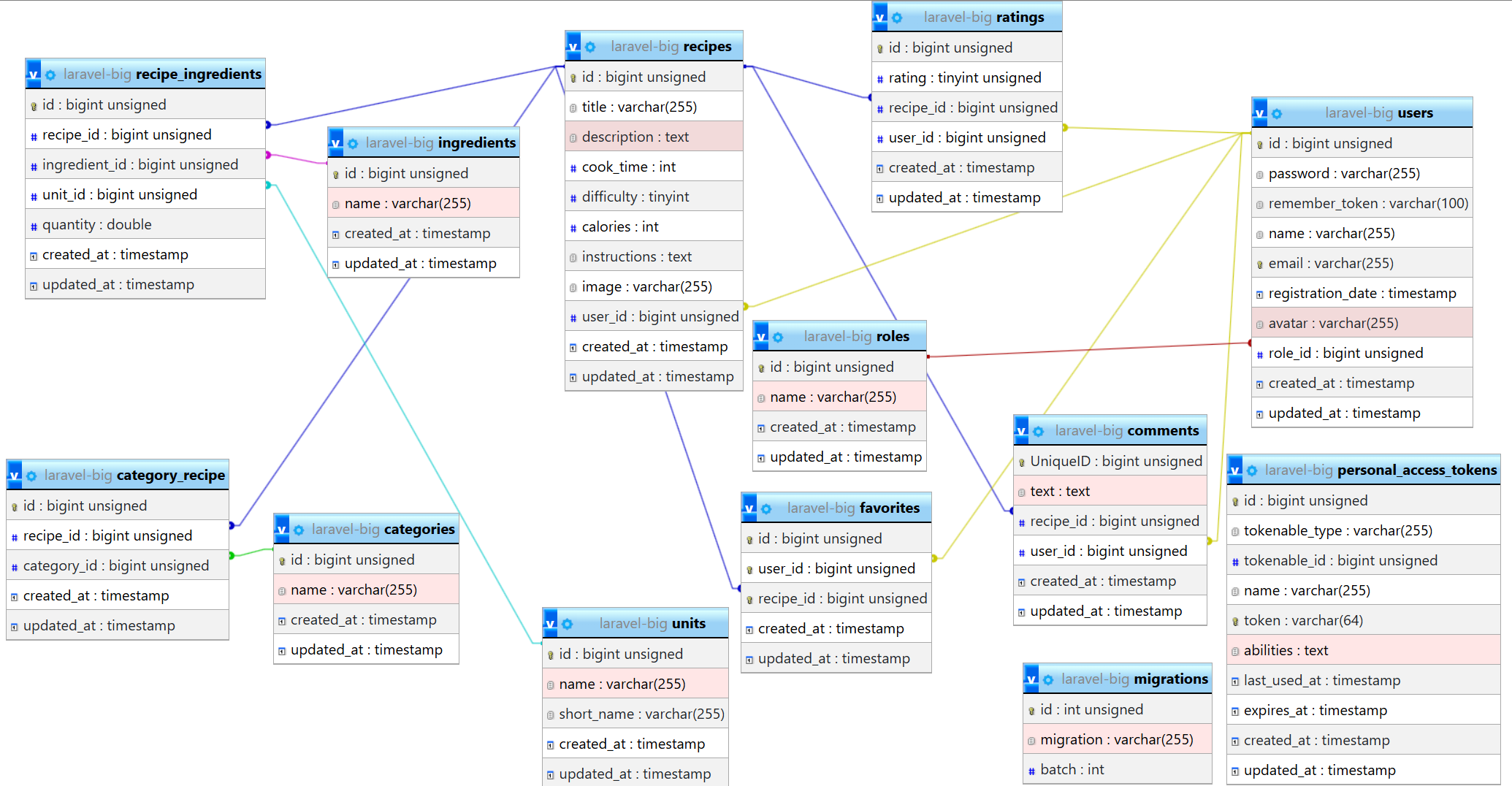


Рисунок 26. Дизайнер базы данных

Примеры данных в основных таблицах

Таблица recipes:

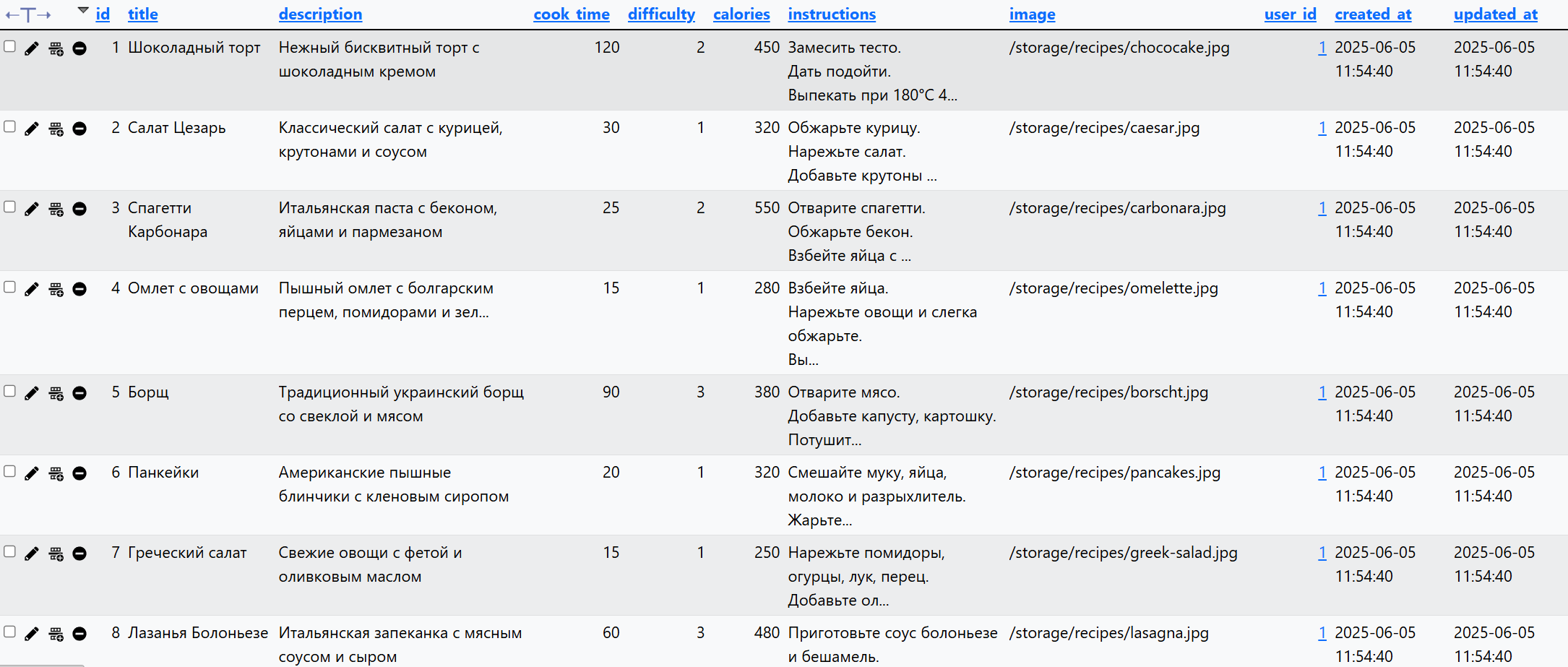


Рисунок 27. Таблица recipes

Таблица category:

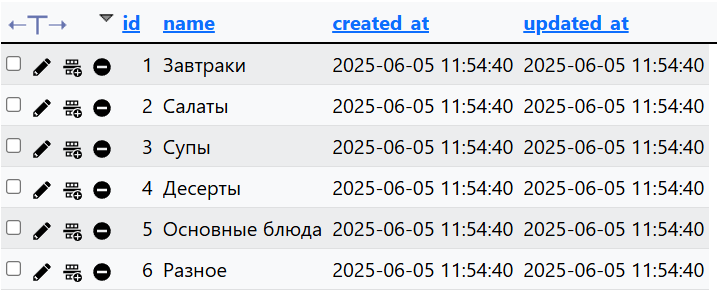


Рисунок 28. Таблица categories

Таблица ingredients:

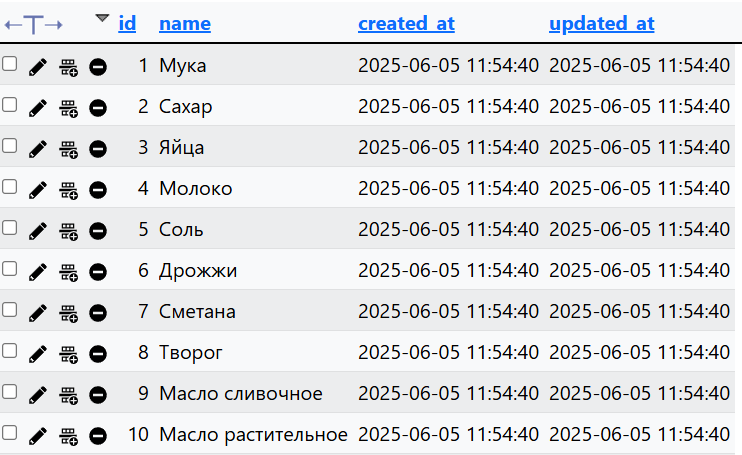


Рисунок 29. Таблица ingredients

Таблица recipe\_ingredients:

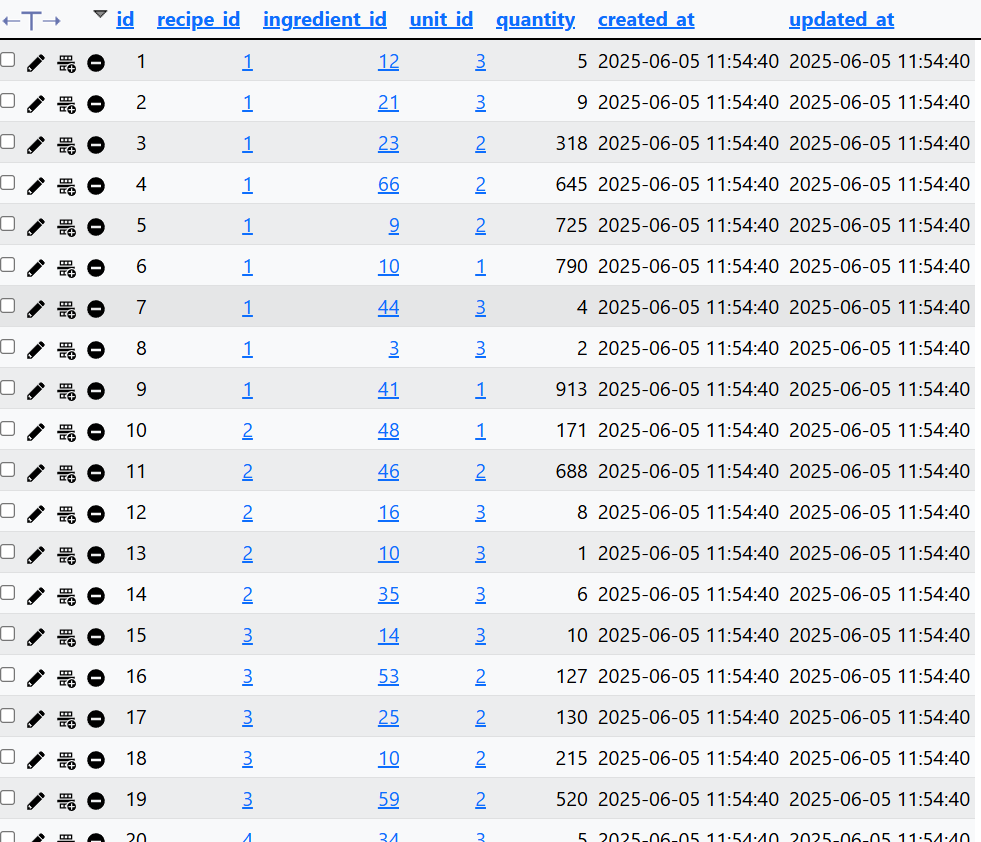


Рисунок 30. Таблица recipe\_ingredients

# **РЕАЛИЗАЦИЯ ВЕБ-САЙТА**

Веб-сайт системы «Рецепты блюд» создан с использованием Laravel и Blade. Интерфейс разработан на основе принципов юзабилити и адаптивного дизайна, что обеспечивает комфортную работу на различных устройствах. Пользовательский опыт ориентирован на простоту и удобство: навигация интуитивна, внешний вид страниц согласован и соответствует разработанным прототипам.

Функциональность сайта включает авторизацию, регистрацию, просмотр и управление рецептами, комментарии, оценки и добавление в избранное. В зависимости от роли пользователя доступен соответствующий набор возможностей. Интерфейс взаимодействует с серверной частью через контроллеры и модели Laravel, а динамические элементы реализованы с использованием JavaScript.

Реализация сайта позволила добиться стабильной работы, понятного взаимодействия с системой и соответствия требованиям, изложенным в техническом задании.

## **Основные страницы**

**Главная страница**

**Как попасть:**

* При переходе на сайт по адресу (домашняя страница, /);
* Через логотип в шапке сайта (если реализовано).

**Что можно делать:**

* Просматривать категории и рецепты;
* Переходить по категориям;
* Открывать карточки рецептов;
* Использовать меню и переключать тему оформления.

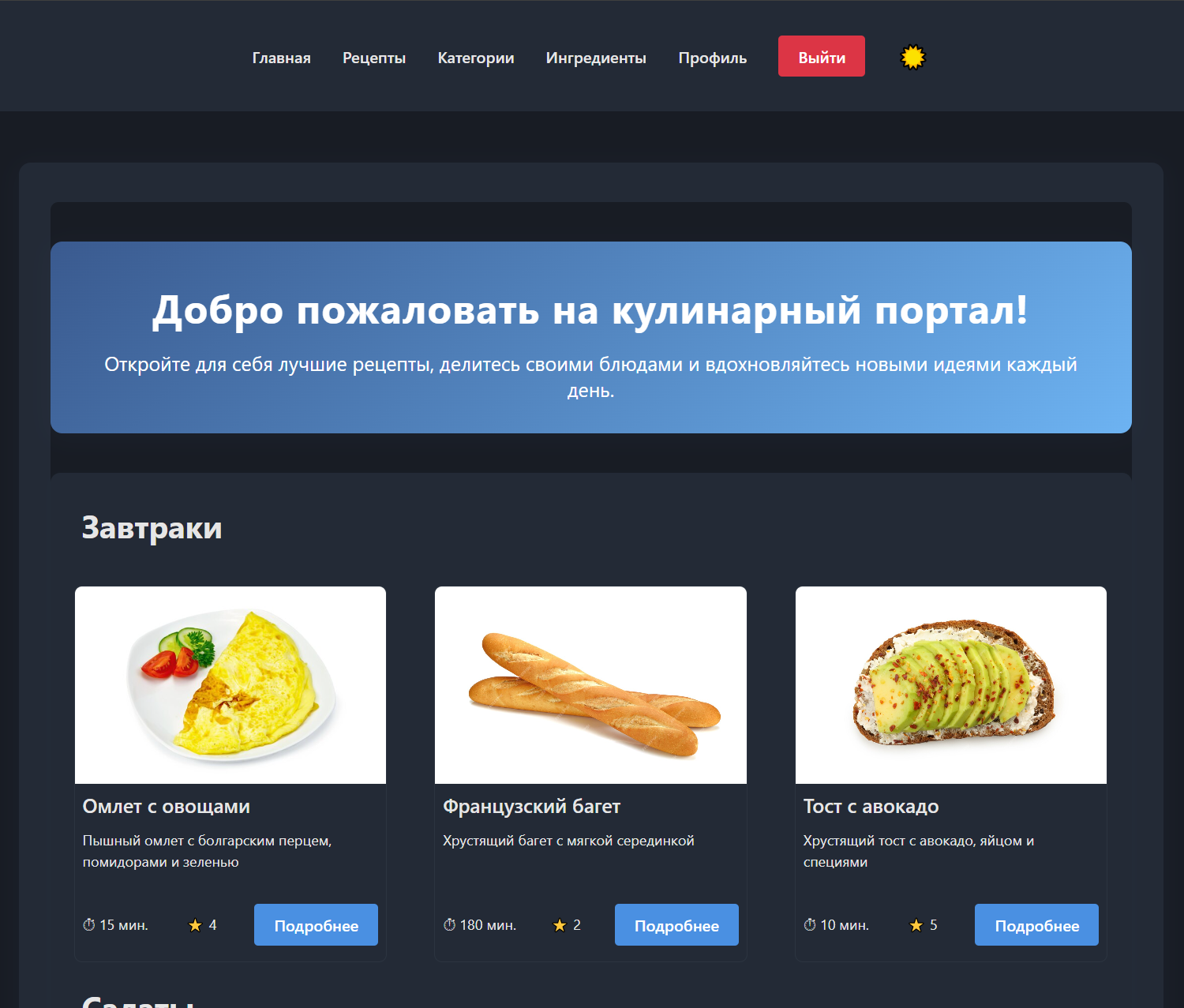


Рисунок 31. Главная страница

**Страница рецепта (подробно)**

**Как попасть:**

* Через кнопку «Подробнее» в карточке рецепта на любой странице (например, «Рецепты», «Категория» или «Главная»).

**Что можно делать:**

* Ознакомиться с полным рецептом;
* Оценить блюдо;
* Добавить в избранное;
* Прокомментировать.

В листинге реализована страница подробный рецепт, где отображается вся информация о рецепте, возможность оставить оценку, комментарий, добавить в избранное, редактирование сообщение и удаление.

Листинг 1. Страница подробный рецепт

@extends('layouts.layout')

@section('title', $recipe->title)

@section('content')

<div *class*="container recipe-detail-container">

*<!-- Header сексция -->*

    <div *class*="recipe-header">

        <div *class*="recipe-image-container">

*<!-- Основное изображение рецепта -->*

            <img *src*="{{ asset($recipe->image) }}" *alt*="{{ $recipe->title }}" *class*="recipe-main-image">

            <div *class*="recipe-overlay">

*<!-- Быстрая информация о рецепте (время, калории, сложность) -->*

                <div *class*="recipe-quick-info">

                    <span *class*="info-item"><i *class*="fas fa-clock"></i> {{ $recipe->cook\_time }} мин</span>

                    <span *class*="info-item"><i *class*="fas fa-fire"></i> {{ $recipe->calories }} ккал</span>

                    <span *class*="info-item">

                        <i *class*="fas fa-signal"></i>

                        @if ($recipe->difficulty == 1)

                            Легко

                        @elseif ($recipe->difficulty == 2)

                            Средне

                        @elseif ($recipe->difficulty == 3)

                            Сложно

                        @endif

                    </span>

                </div>

            </div>

        </div>

        <div *class*="recipe-title-section">

*<!-- Название рецепта и действия пользователя -->*

            <h1 *class*="recipe-title">{{ $recipe->title }}</h1>

            <div *class*="recipe-actions">

                <div *class*="rating-display">

*<!-- Отображение средней оценки -->*

                    <div *class*="stars">

                        @for ($i = 1; $i <= 5; $i++)

                            <i *class*="fas fa-star {{ $i <= ($recipe->ratings\_avg\_rating ?? 0) ? 'active' : '' }}"></i>

                        @endfor

                    </div>

                    <span *class*="rating-value">{{ number\_format($recipe->ratings\_avg\_rating ?? 0, 1) }}/5</span>

                </div>

*<!-- Кнопки добавления/удаления из избранного -->*

                @if ($favorites && $favorites->contains($recipe->id))

                    <form *action*="{{ route('recipes.favorite', $recipe->id) }}" *method*="POST" *class*="favorite-form">

                        @csrf

                        @method('DELETE')

                        <button *type*="submit" *class*="btn btn-danger btn-sm">

                            <i *class*="fas fa-heart"></i> Удалить из избранного

                        </button>

                    </form>

                @else

                    <form *action*="{{ route('recipes.favorite', $recipe->id) }}" *method*="POST" *class*="favorite-form">

                        @csrf

                        <button *type*="submit" *class*="btn btn-primary btn-sm">

                            <i *class*="far fa-heart"></i> Добавить в избранное

                        </button>

                    </form>

                @endif

            </div>

        </div>

    </div>

    <div *class*="recipe-description-section">

*<!-- Описание рецепта -->*

        <p *class*="recipe-description">{{ $recipe->description }}</p>

    </div>

    <div *class*="recipe-content-grid">

        <div *class*="ingredients-section">

*<!-- Список ингредиентов -->*

            <h2 *class*="section-title"><i *class*="fas fa-list"></i> Ингредиенты</h2>

            <ul *class*="ingredients-list">

                @foreach ($recipe->ingredients as $ingredient)

                    <li *class*="ingredient-item">

                        <i *class*="fas fa-check"></i>

                        <span>

                            {{ $ingredient->name }}

                            @if($ingredient->pivot->quantity)

                                — {{ $ingredient->pivot->quantity }}

                                @if($ingredient->pivot->unit)

                                    {{ $ingredient->pivot->unit->short\_name }}

                                @endif

                            @endif

                        </span>

                    </li>

                @endforeach

            </ul>

        </div>

*<!-- Секция инструкции -->*

        <div *class*="instructions-section">

*<!-- Пошаговая инструкция приготовления -->*

            <h2 *class*="section-title"><i *class*="fas fa-utensils"></i> Способ приготовления</h2>

            <div *class*="instructions-content">

                @php

                    $text = trim($recipe->instructions);

                    // Сначала пробуем разбить по шаблону '1. ... 2. ...'

                    $steps = preg\_split('/(?<=\d\.|\d\))\s\*/u', $text, -1, PREG\_SPLIT\_NO\_EMPTY);

                    // Если не найдено несколько шагов, разбиваем по переводу строки

                    if (count($steps) <= 1) {

                        $steps = preg\_split('/\r?\n|\r/', $text);

                    }

                @endphp

                @if(count($steps) > 1)

                    <ol *style*="padding-left: 1.2rem;">

                        @foreach($steps as $step)

                            @if(trim($step) !== '')

                                <li>{{ trim($step) }}</li>

                            @endif

                        @endforeach

                    </ol>

                @else

                    <div>{{ $steps[0] }}</div>

                @endif

            </div>

        </div>

    </div>

*<!-- Секция рейтинга -->*

    <div *class*="rating-section">

*<!-- Блок для выставления оценки рецепту -->*

        <h2 *class*="section-title"><i *class*="fas fa-star"></i> Оценить рецепт</h2>

        <div *class*="star-rating" *data-recipe-id*="{{ $recipe->id }}" *data-user-rating*="{{ $userRating ?? 0 }}">

            @for ($i = 1; $i <= 5; $i++)

                <i *class*="fas fa-star star" *data-value*="{{ $i }}"></i>

            @endfor

        </div>

    </div>

*<!-- Секция комментариев -->*

    <div *class*="comments-section">

        <h2 *class*="section-title"><i *class*="fas fa-comments"></i> Комментарии</h2>

*<!-- Форма комментариев -->*

        <form *method*="POST" *action*="{{ route('recipes.comment', $recipe) }}" *class*="comment-form">

            @csrf

            <div *class*="form-group">

                <textarea *name*="text" *class*="form-control" *rows*="3" *placeholder*="Поделитесь своим мнением о рецепте..." *required*></textarea>

            </div>

            <button *type*="submit" *class*="btn btn-primary">

                <i *class*="fas fa-paper-plane"></i> Отправить комментарий

            </button>

        </form>

*<!-- Лист комментариев -->*

        <div *class*="comments-list">

            @foreach ($recipe->comments as $comment)

                <div *class*="comment-card">

                    <div *class*="comment-header">

                        <div *class*="comment-user">

                            <i *class*="fas fa-user-circle"></i>

                            <strong>{{ $comment->user->name }}</strong>

                        </div>

                        <span *class*="comment-date" *style*="margin-left:auto;">{{ $comment->created\_at->diffForHumans() }}</span>

                        @auth

                            @if(auth()->user()->role\_id === 1 || auth()->user()->id === $comment->user\_id)

*<!-- Кнопка редактирования комментария -->*

                                <button *type*="button" *class*="edit-comment-btn" *title*="Редактировать" *onclick*="toggleEditForm('{{ $comment->UniqueID }}')">

                                    <i *class*="fas fa-edit"></i>

                                </button>

*<!-- Кнопка удаления комментария -->*

                                <form *method*="POST" *action*="{{ route('comments.destroy', ['comment' => $comment->UniqueID]) }}" *style*="display:inline; margin-left: 0.5rem;">

                                    @csrf

                                    @method('DELETE')

                                    <button *type*="submit" *class*="delete-comment-btn" *title*="Удалить" *onclick*="return confirm('Удалить комментарий?')">

                                        <i *class*="fas fa-trash-alt"></i>

                                    </button>

                                </form>

                            @endif

                        @endauth

                    </div>

                    <div *class*="comment-content" *id*="comment-content-{{ $comment->UniqueID }}">

                        {{ $comment->text }}

                    </div>

                    @auth

                        @if(auth()->user()->role\_id === 1 || auth()->user()->id === $comment->user\_id)

                            <div *class*="edit-comment-form" *id*="edit-form-{{ $comment->UniqueID }}" *style*="display: none;">

*<!-- Форма редактирования комментария -->*

                                <form *method*="POST" *action*="{{ route('comments.update', ['comment' => $comment->UniqueID]) }}" *class*="mt-2">

                                    @csrf

                                    @method('PUT')

                                    <div *class*="form-group">

                                        <textarea *name*="text" *class*="form-control" *rows*="3" *required*>{{ $comment->text }}</textarea>

                                    </div>

                                    <div *class*="mt-2">

                                        <button *type*="submit" *class*="btn btn-primary btn-sm">

                                            <i *class*="fas fa-save"></i> Сохранить

                                        </button>

                                        <button *type*="button" *class*="btn btn-secondary btn-sm" *onclick*="toggleEditForm('{{ $comment->UniqueID }}')">

                                            <i *class*="fas fa-times"></i> Отмена

                                        </button>

                                    </div>

                                </form>

                            </div>

                        @endif

                    @endauth

                </div>

            @endforeach

        </div>

    </div>

</div>

*<!-- Добавляем стили для темной темы комментариев -->*

<style>

*.dark-theme* *.comment-card*,

*.dark-theme* *.comment-header*,

*.dark-theme* *.comment-content*,

*.dark-theme* *.comment-date*,

*.dark-theme* *.comment-user* strong {

    color: var(--text) !important;

}

*.dark-theme* *.comment-card* {

    background: var(--card-bg) !important;

    border: 1px solid var(--border) !important;

}

</style>

*<!-- Подключаем скрипт для работы с рейтингом -->*

<script *src*="{{ asset('assets/js/rating.js') }}"></script>

@push('scripts')

<script>

const isGuest = @json($isGuest ?? false);

document.addEventListener('DOMContentLoaded', function() {

*// Обработка клика по звездам рейтинга*

    document.querySelectorAll('.star-rating .star').forEach(function(star) {

        star.addEventListener('click', function(e) {

            if (isGuest) {

                window.location.href = '/login';

                return;

            }

        });

    });

});

*// Функция для переключения формы редактирования комментария*

function toggleEditForm(commentId) {

    const contentDiv = document.getElementById(`comment-content-${commentId}`);

    const editForm = document.getElementById(`edit-form-${commentId}`);

    if (contentDiv.style.display !== 'none') {

        contentDiv.style.display = 'none';

        editForm.style.display = 'block';

    } else {

        contentDiv.style.display = 'block';

        editForm.style.display = 'none';

    }

}

</script>

@endpush

@endsection

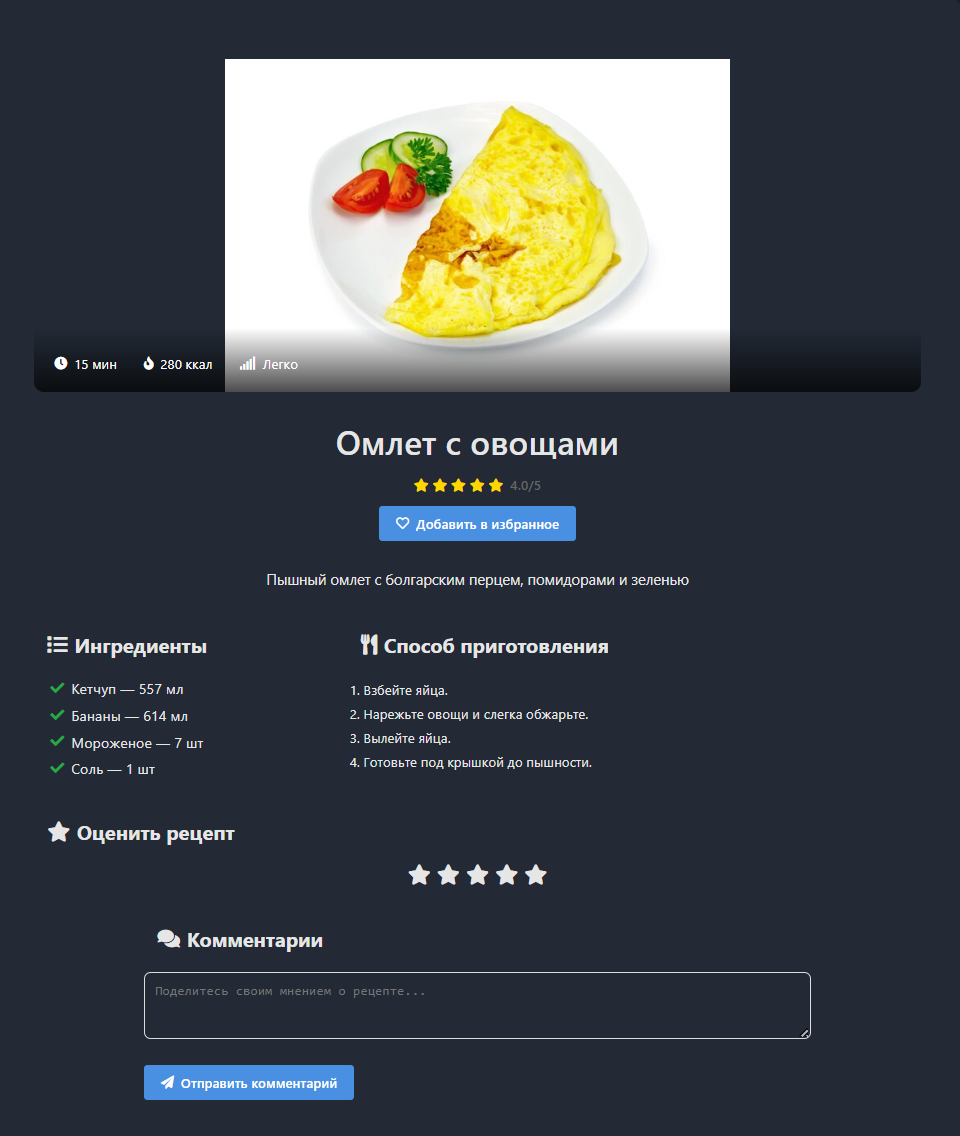


Рисунок 32. Подробный рецепт страница

**Рецепты**

**Как попасть:**

* Через пункт меню «Рецепты»;
* Через главную страницу (например, при клике на категорию).

**Что можно делать:**

* Искать по названию;
* Фильтровать по категориям;
* Сбрасывать фильтры;
* Переходить к полному описанию рецепта;
* Перейти в «Мои рецепты» (если авторизован).

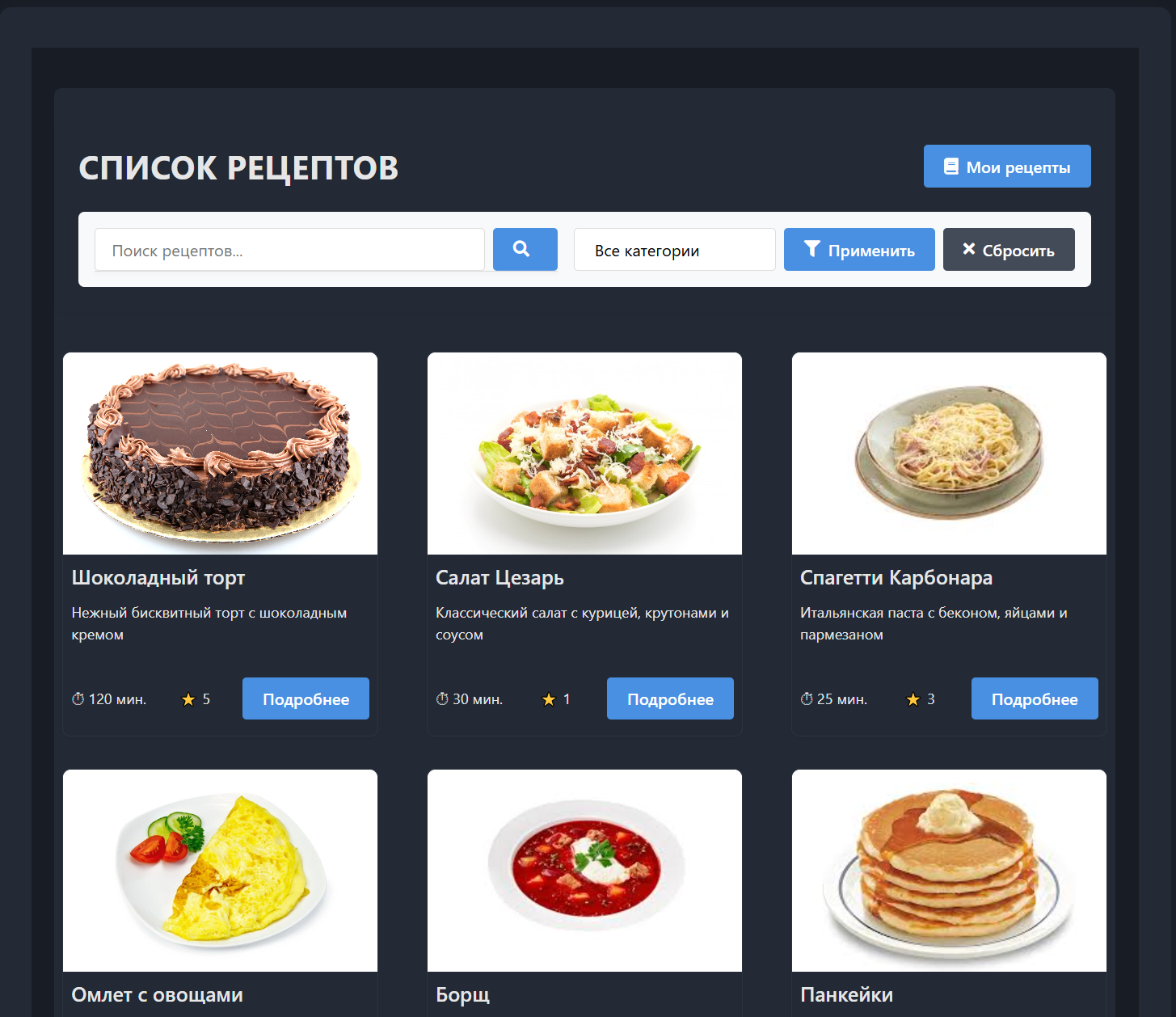


Рисунок 33. Рецепты страница

**Мои рецепты**

**Как попасть:**

* Через кнопку «Мои рецепты» на странице рецептов (если авторизован);
* Через пункт меню (если реализовано).

**Что можно делать:**

* Просматривать свои рецепты;
* Редактировать и удалять их;
* Добавлять новые рецепты.

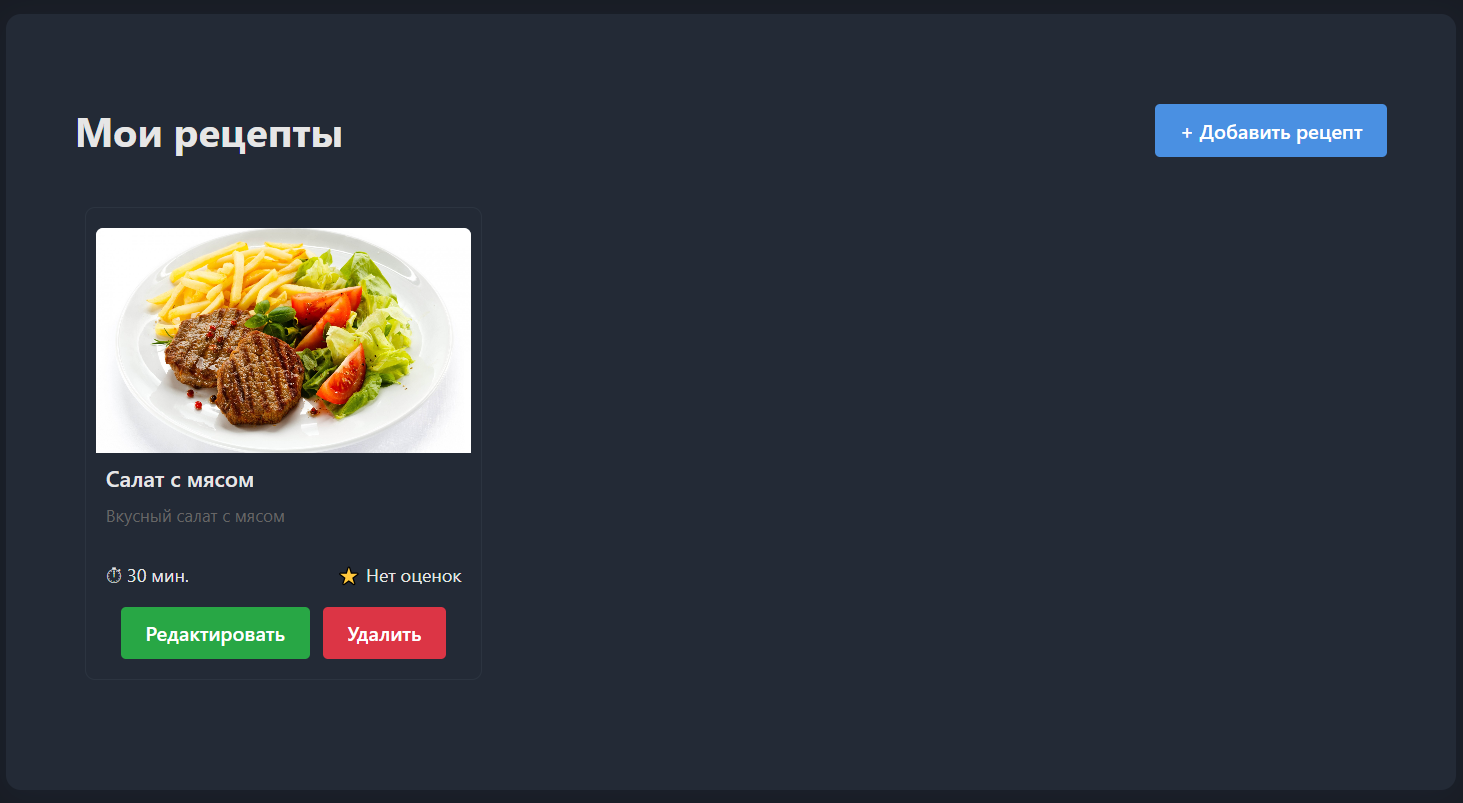


Рисунок 34. Мои рецепты страница

**Добавление рецепта**

**Как попасть:**

* Через кнопку «Добавить рецепт» на странице «Мои рецепты».

**Что можно делать:**

* Заполнять форму рецепта;
* Добавлять категории и ингредиенты;
* Загружать изображение;
* Сохранять рецепт.

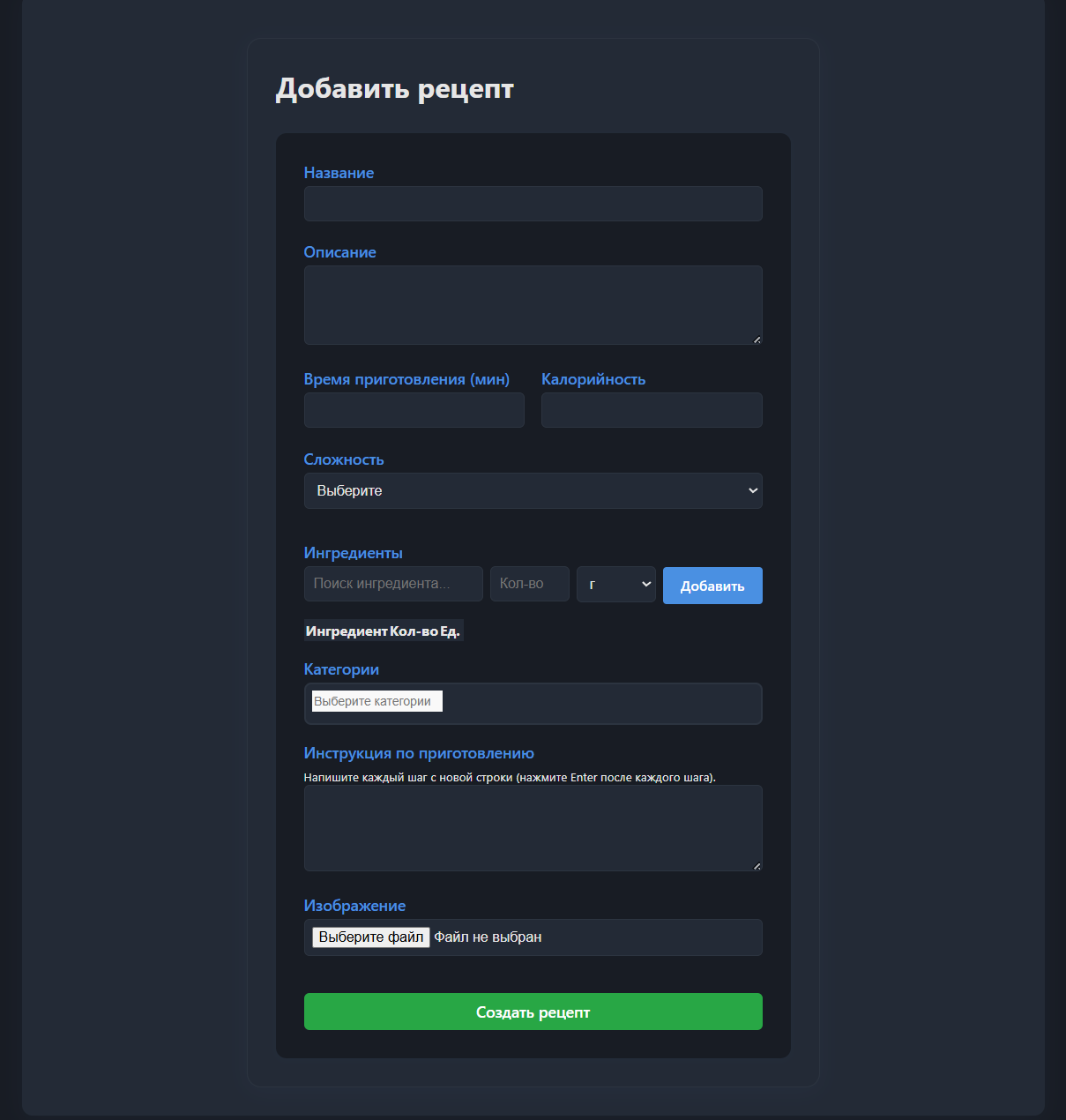


Рисунок 35. Добавление рецепта страница

**Редактирование рецепта**

**Как попасть:**

* Через кнопку «Редактировать» на карточке своего рецепта (в «Мои рецепты»).

**Что можно делать:**

* Менять любые данные рецепта;
* Заменять изображение;
* Сохранять изменения.

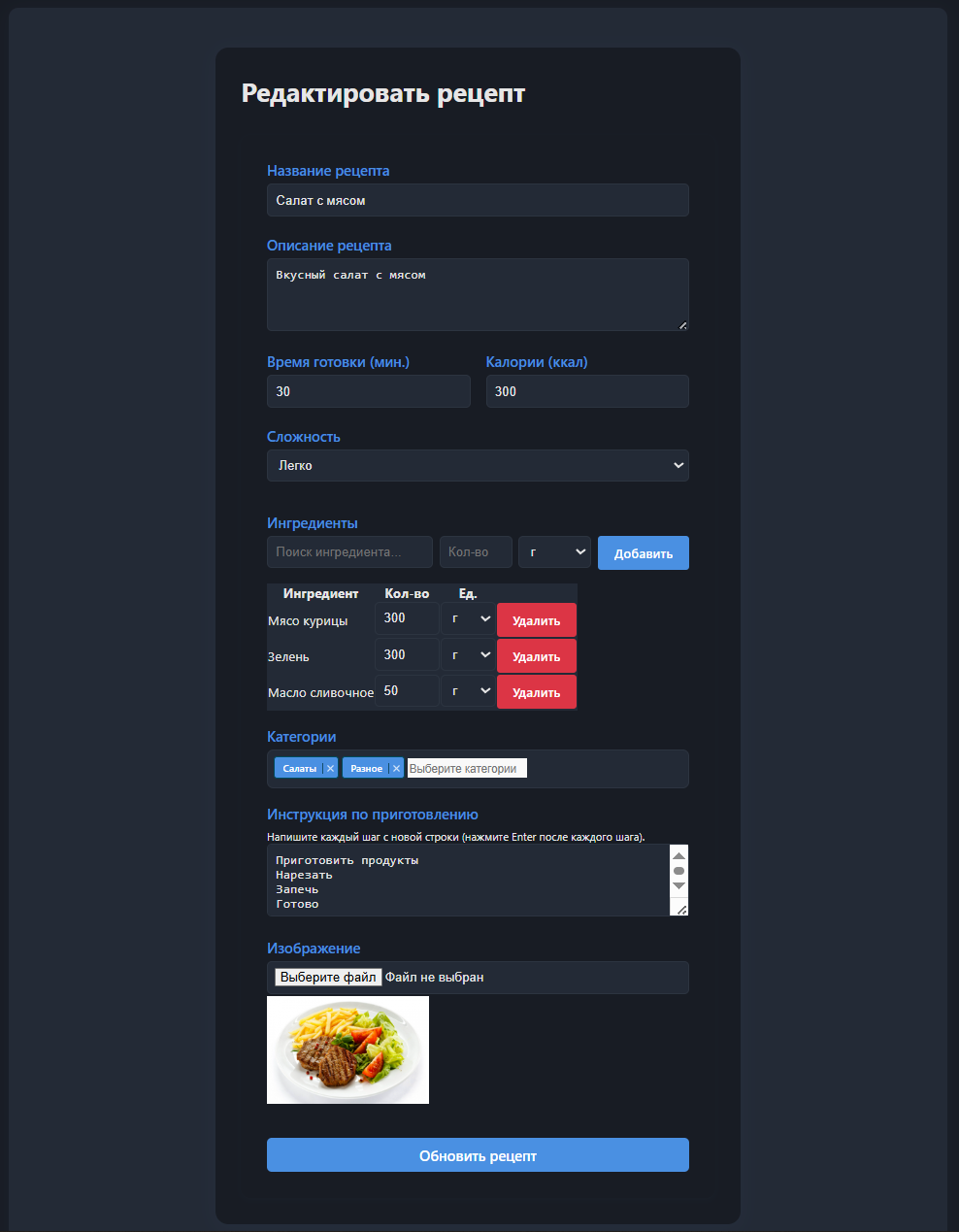


Рисунок 36. Редактирование рецепта страница

**Профиль пользователя**

**Как попасть:**

* Через пункт меню «Профиль» (если авторизован).

**Что можно делать:**

* Посмотреть свои данные;
* Перейти к редактированию или смене пароля;
* Удалить аккаунт.

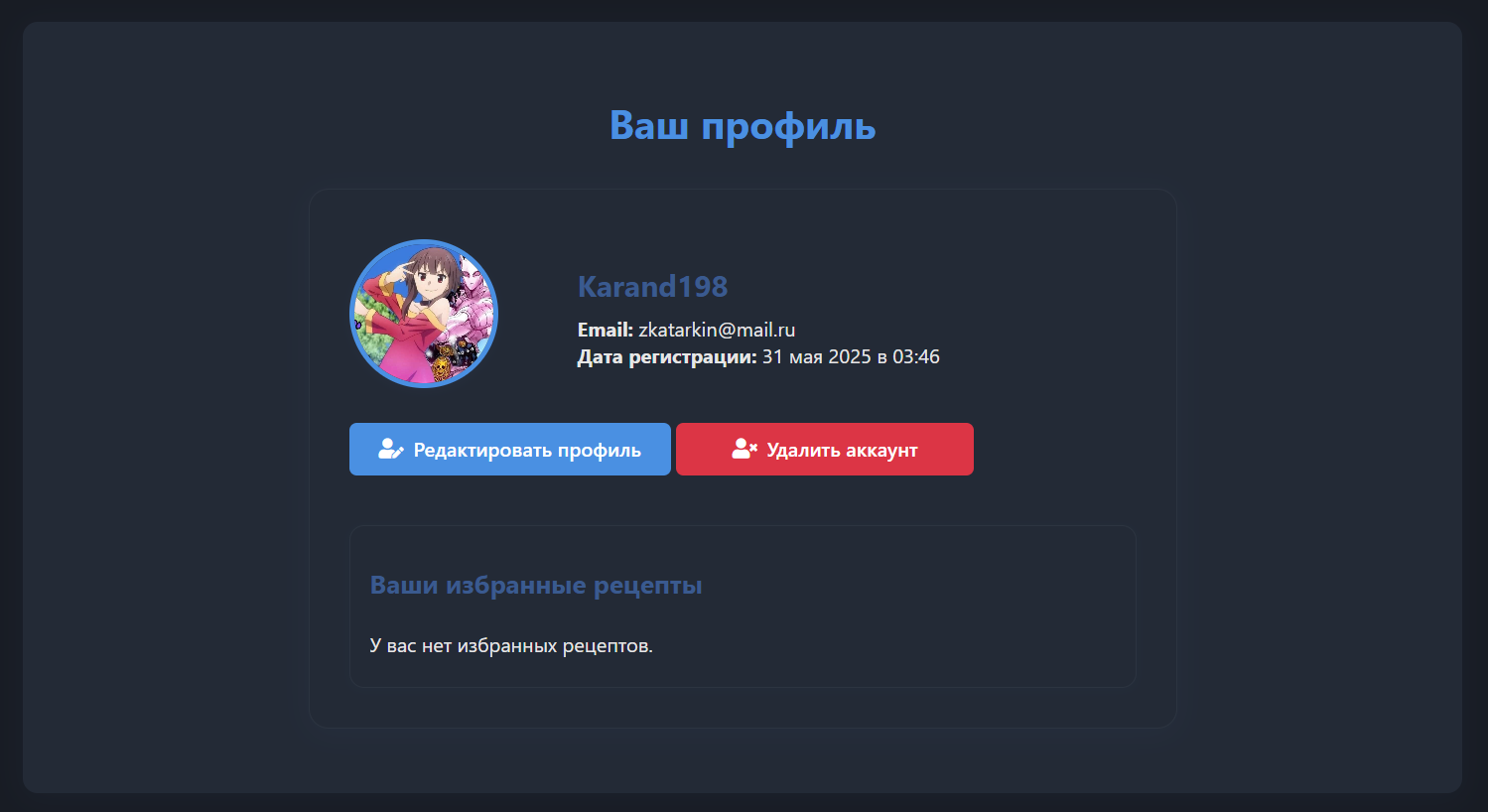


Рисунок 37. Профиль страница

**Редактирование профиля**

**Как попасть:**

* Через нажатия кнопки Редактировать профиль на странице профиля.

**Что можно делать:**

* Сменить имя;
* Сменить почту;
* Сменить аватар;
* Сохранить изменения;
* Перейти на страницу смены пароля.

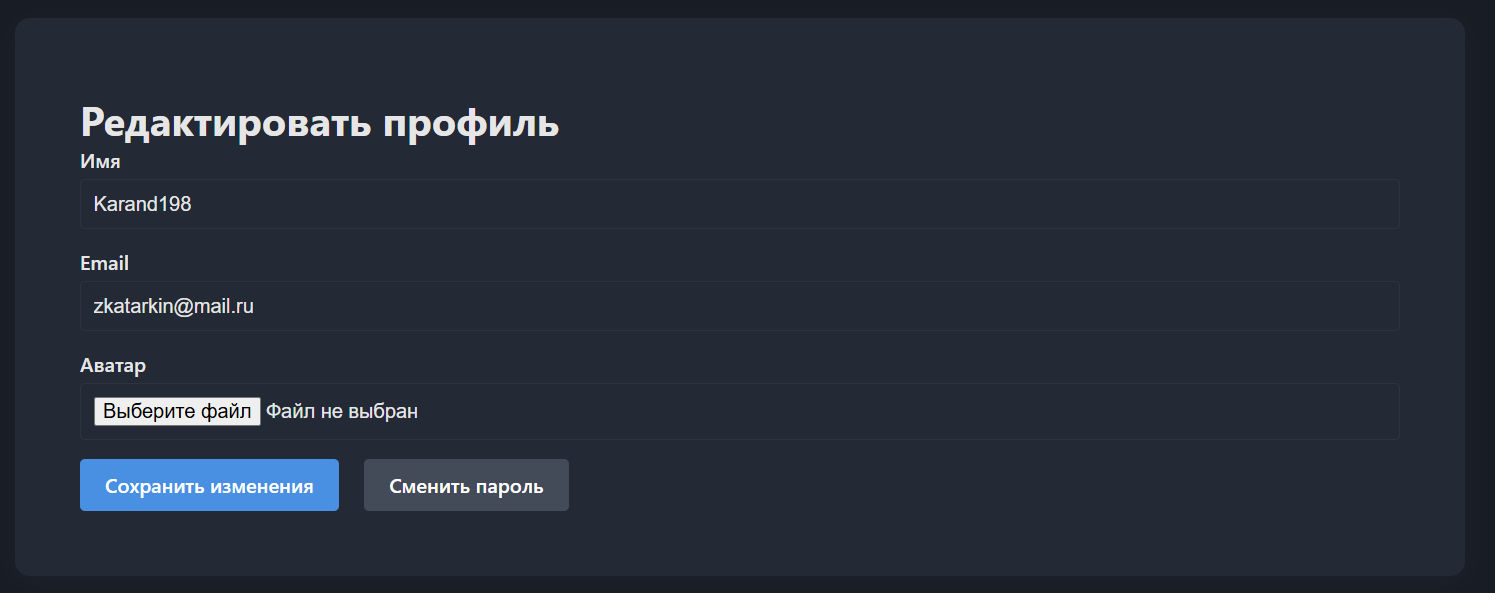


Рисунок 38. Редактирование профиля страница

**Смена пароля**

**Как попасть:**

* Из страницы редактирования профиля — по кнопке «Сменить пароль».

**Что можно делать:**

* Ввести старый и новый пароль;
* Подтвердить изменение;
* Сменить пароль.

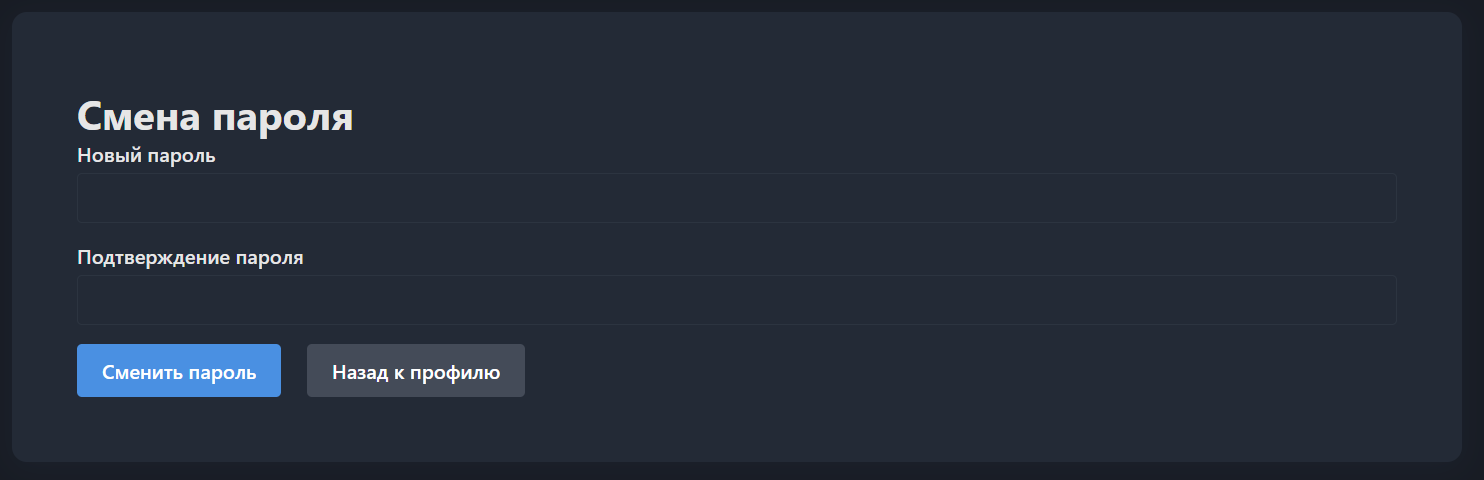


Рисунок 39. Смена пароля страница

**Категории (Админ)**

**Как попасть:**

* Через шапку на странице, появляется только если роль пользователя администратор.

**Что можно делать:**

* Добавлять новые категории;
* Просматривать существующие.

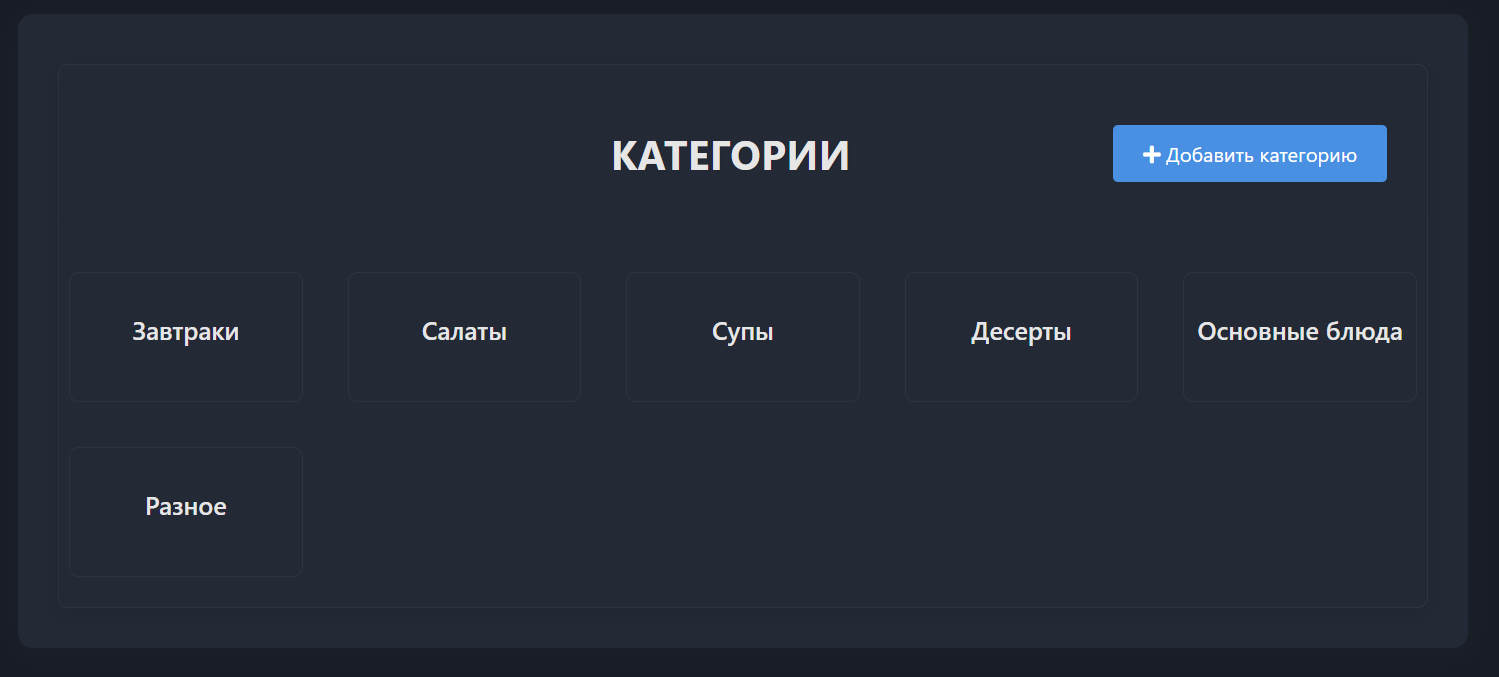


Рисунок 40. Категории страница

**Определённая категория**

**Как попасть:**

* Через клик на категорию (на главной или на странице рецептов).

**Что можно делать:**

* Видеть рецепты из одной категории;
* Переходить к конкретному рецепту.

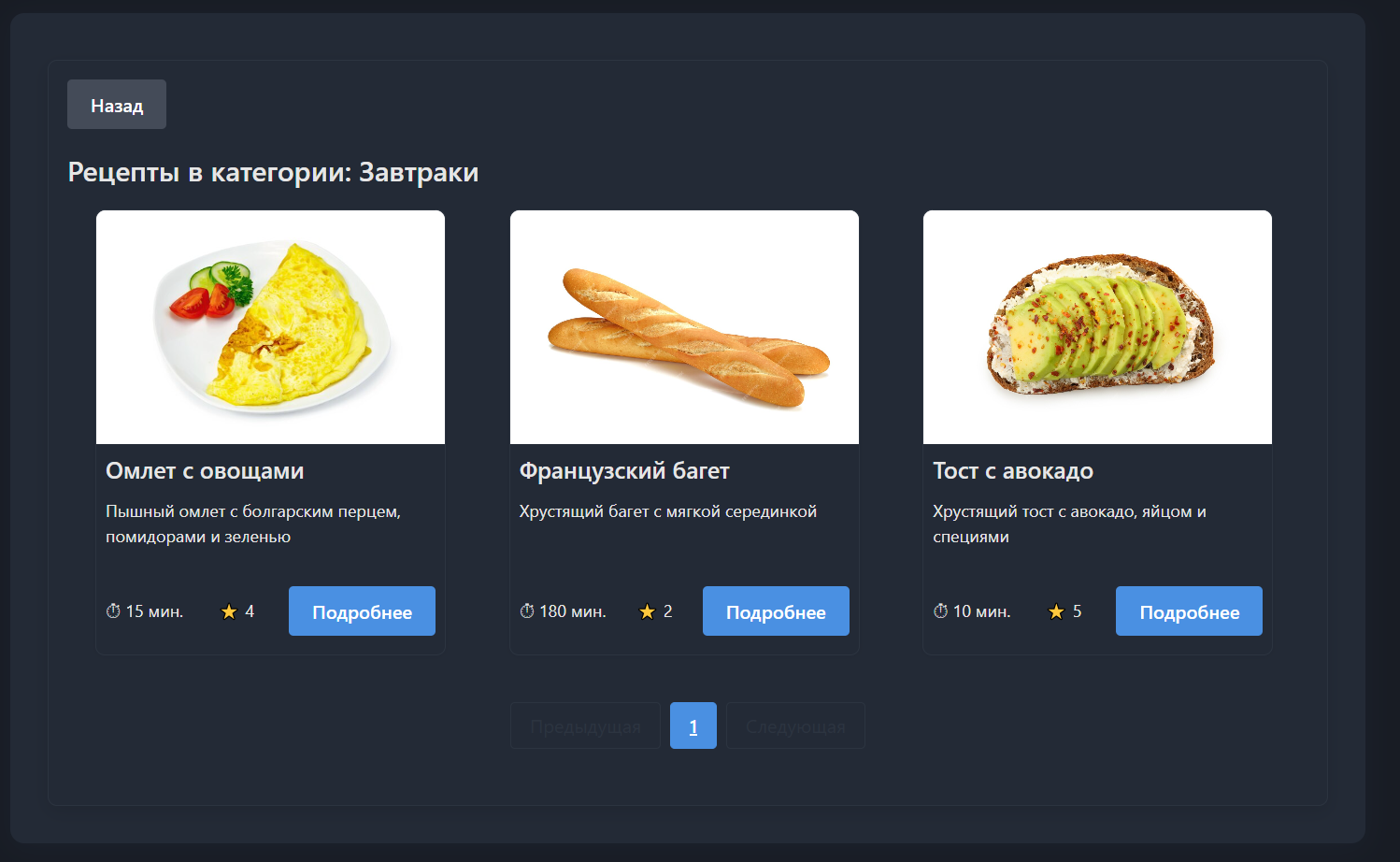


Рисунок 41. Определенная категория страница

**Добавления категории**

**Как попасть:**

* Через клик на кнопку Добавить категорию на странице Категории.

**Что можно делать:**

* Ввести название категории;
* Добавить категорию нажатием на кнопку Добавить категорию.

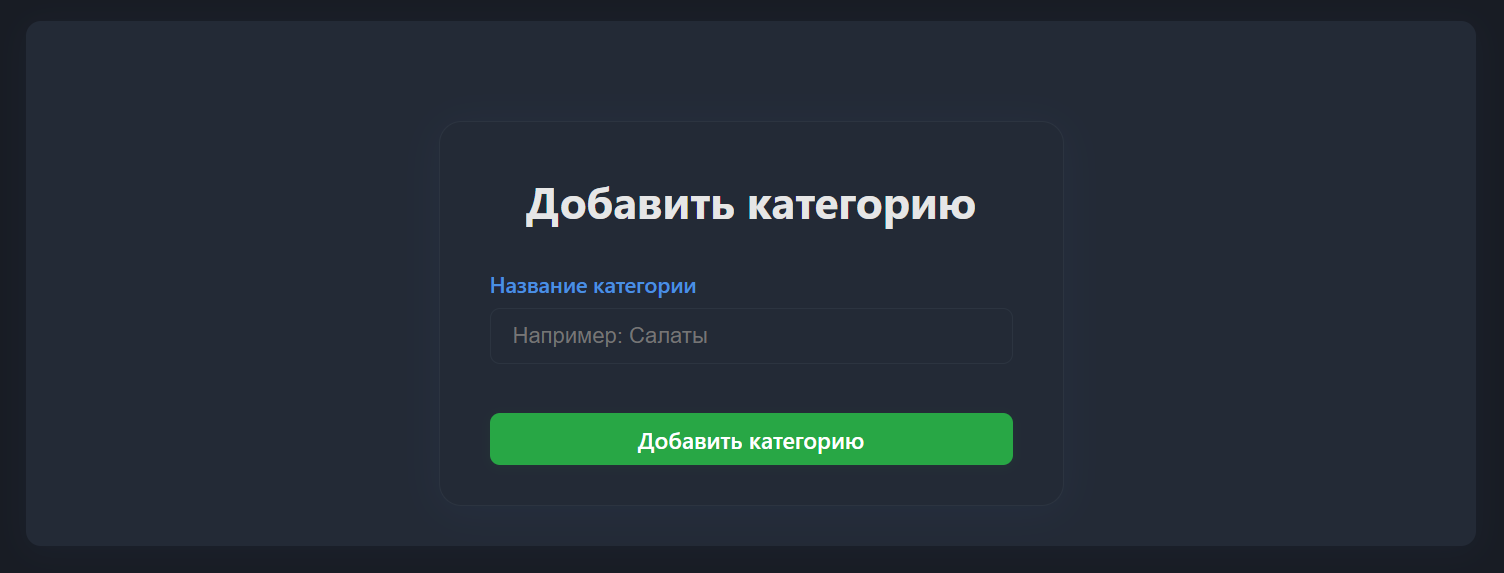


Рисунок 42. Добавление категории страница

**Ингредиенты**

**Как попасть:**

* Через шапку страницы, появляется только если роль пользователя администратор.

**Что можно делать:**

* Ввести название ингредиента;
* Добавить ингредиент нажатием на кнопку Добавить ингредиент.

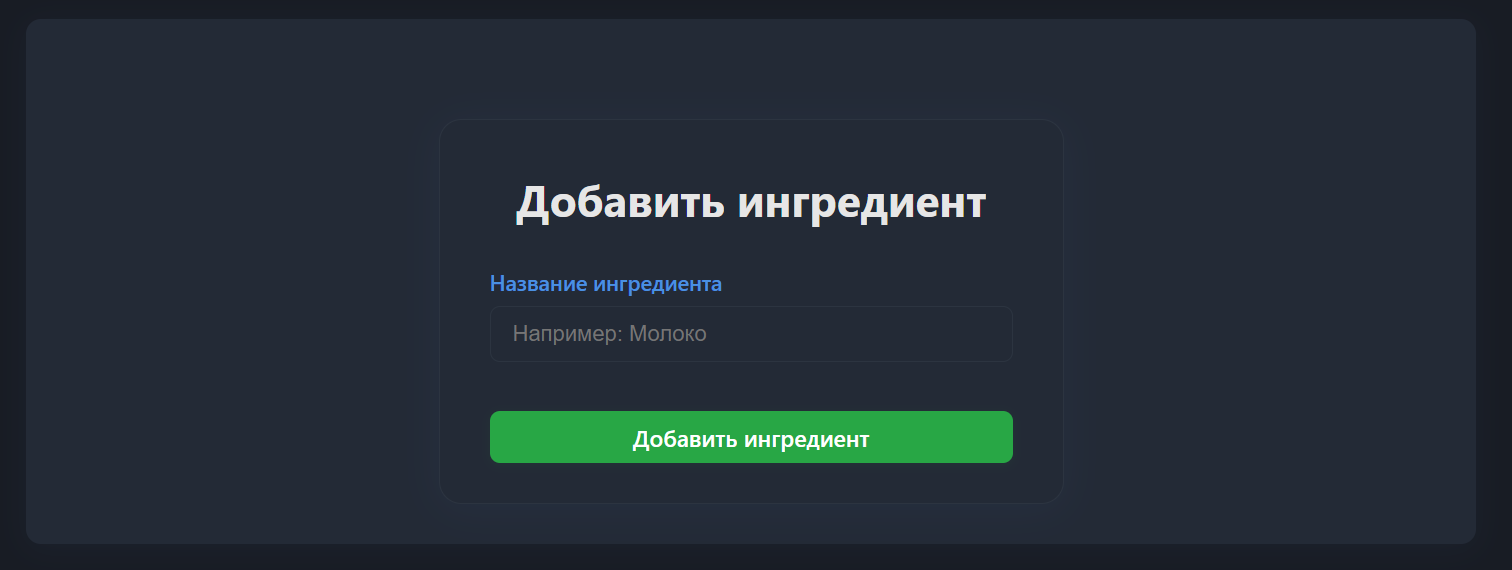


Рисунок 43. Добавление ингредиента страница

**Авторизация**

**Как попасть:**

* Через кнопку «Войти» в шапке сайта (если пользователь не авторизован);
* Автоматически после попытки доступа к защищённой странице без авторизации.

**Что можно делать:**

* Ввести email и пароль;
* Войти в систему;
* Перейти на регистрацию.

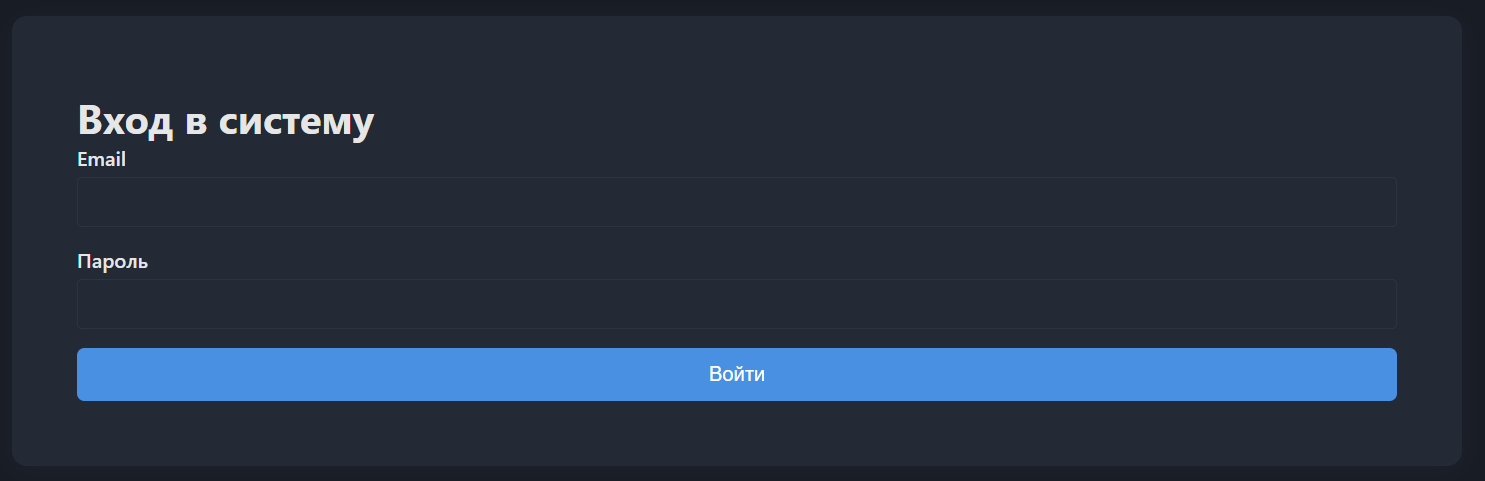


Рисунок 44. Авторизация страница

**Регистрация**

**Как попасть:**

* Из шапки страницы нажав на кнопку регистрация.

**Что можно делать:**

* Ввести данные;
* Загрузить аватар;
* Создать аккаунт.

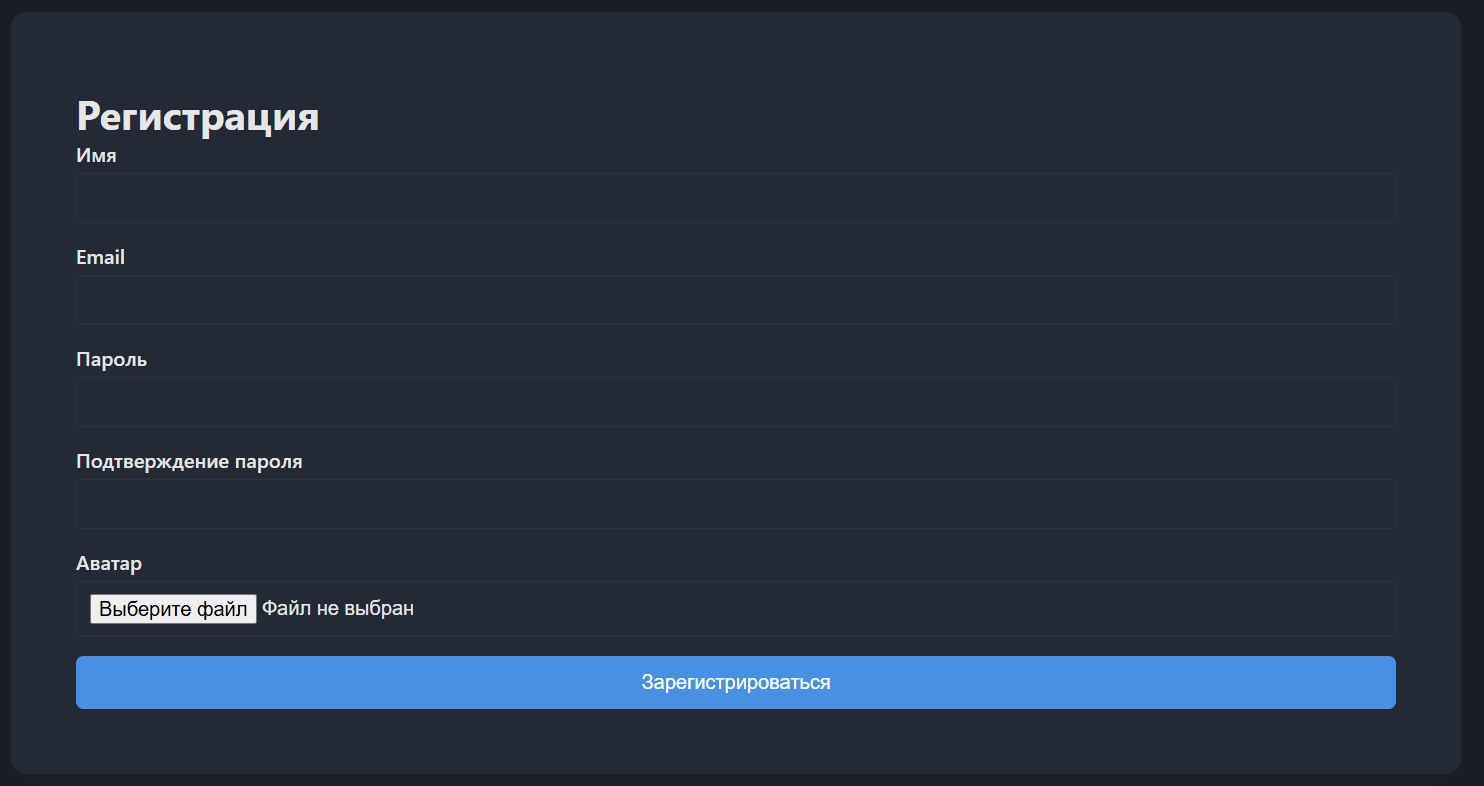


Рисунок 45. Регистрация страница

# **ТЕСТИРОВАНИЕ ВЕБ-САЙТА**

**Тест-кейсы для ручного тестирования веб-приложения**

В ходе разработки веб-приложения было проведено ручное тестирование всех основных пользовательских сценариев с целью проверки корректности работы интерфейса, а также адаптивности и устойчивости к ошибкам.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** | **Сценарий** | **Входные данные** | **Этапы тестирования** | Ожидаемый результат | **Факт. результат** | **Статус** |
| 1 | Регистрация пользователя (валидный) | Имя: user123, Email: user@mail.com, Пароль: Pass1234! | Заполнить форму регистрации, нажать "Зарегистрироваться" | Успешная регистрация, переход в личный кабинет | Успешная регистрация, переход в личный кабинет | Выполнен |
| 2 | Регистрация с пустыми полями | Имя: , Email: , Пароль: `` | Нажать "Зарегистрироваться" | Ошибка: обязательные поля | Ошибка: обязательные поля | Выполнен |
| 3 | Регистрация с невалидным email | Имя: user, Email: usermail.com, Пароль: Pass1234! | Нажать "Зарегистрироваться" | Ошибка: "Неверный формат email" | Ошибка: "Неверный формат email" | Выполнен |
| 4 | Регистрация с коротким паролем | Имя: user, Email: user@mail.com, Пароль: 123 | Нажать "Зарегистрироваться" | Ошибка: "Пароль минимум 7 символов" | Ошибка: "Пароль минимум 7 символов" | Выполнен |
| 5 | Авторизация пользователя (валидный) | Email: user@mail.com, Пароль: Pass1234! | Ввести логин и пароль, нажать "Войти" | Вход в систему, отображение профиля | Вход в систему, отображение профиля | Выполнен |
| 6 | Авторизация с неверным паролем | Email: user@mail.com, Пароль: WrongPass | Нажать "Войти" | Ошибка: "Неверный логин или пароль" | Ошибка: "Неверный логин или пароль" | Выполнен |
| 7 | Просмотр списка рецептов | Не требуется | Перейти на страницу «Рецепты» | Отображается список рецептов с фильтрами | Отображается список рецептов с фильтрами | Выполнен |
| 8 | Просмотр подробного рецепта | Не требуется | Клик по карточке рецепта | Открывается страница с полным описанием | Открывается страница с полным описанием | Выполнен |
| 9 | Добавление рецепта (авторизован) | Название, изображение, шаги, ингредиенты | Заполнить форму, нажать "Сохранить" | Рецепт появляется в общем списке и в профиле | Рецепт появляется в общем списке и в профиле | Выполнен |
| 10 | Добавление рецепта (гость) | Не требуется | Попробовать перейти на форму добавления рецепта | Перенаправление на страницу входа | Перенаправление на страницу входа | Выполнен |
| 11 | Добавление в избранное | Не требуется | Нажать на значок "избранное" на карточке рецепта | Рецепт сохраняется в разделе избранного | Рецепт сохраняется в разделе избранного | Выполнен |
| 12 | Комментирование рецепта | Текст комментария | Написать комментарий, нажать "Отправить" | Комментарий отображается под рецептом | Комментарий отображается под рецептом | Выполнен |
| 13 | Изменение данных профиля | Новое имя, аватар | Зайти в профиль, внести изменения, нажать "Сохранить" | Данные обновлены и отображаются на странице профиля | Данные обновлены и отображаются на странице профиля | Выполнен |

В процессе ручного тестирования были проверены все ключевые пользовательские сценарии.

Результаты:

* Все функции реализованы и работаю корректно.
* Страницы адаптированы под все роли и корректно отображается на всех устройствах.
* Ошибки обрабатываются надежно, ошибок обнаружено не было.

# **РУКОВОДСТВО АДМИНИСТРАТОРА**

Администратор обладает полным доступом к функциональности системы и несёт ответственность за её настройку и модерацию. Ниже приведены ключевые разделы и доступные в них действия.

Основные функции администратора

* Управление категориями:
  + Просмотр категорий.
  + Добавление категорий.
* Управление ингредиентами:
  + Добавление ингредиентов.
* Модерация комментариев:
  + Удаление сомнительных комментариев.

# **РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Пользователь — зарегистрированный участник системы, получающий доступ к расширенному функционалу: добавлению рецептов, оценкам, комментариям и управлению профилем.

**Основные функции пользователя**

* Просмотр рецептов:
  + Пользователь может просматривать все рецепты, фильтровать их по категориям и искать по названию или ингредиентам.
* Добавление рецептов:
  + Через личный кабинет пользователь может создать собственный рецепт, указав название, изображение, ингредиенты, шаги приготовления и категорию.
* Оценка рецептов:
  + Возможность поставить оценку от 1 до 5 звёзд. Средний рейтинг отображается на странице рецепта.
* Комментирование:
  + Оставление комментариев к рецептам, возможность обсуждения.
* Избранное:
  + Добавление понравившихся рецептов в избранное для быстрого доступа в будущем.
* Управление профилем:
  + Изменение имени, аватара, пароля и просмотр списка добавленных и сохранённых рецептов.

# **ЗАКЛЮЧЕНИЕ**

Целью дипломного проекта являлось проектирование и разработка онлайн веб-приложения “рецепты блюд”. Разрабатываемое веб-приложение должно было обеспечивать удобный доступ к кулинарным рецептам, реализовать функции оценки, комментариев, избранного и управление контентом с учётом ролей пользователей.

Для достижения поставленной цели были решены следующие задачи:

* Анализ предметной области.
* Проектирование структуры базы данных и логики приложения.
* Разработка backend-части на Laravel (роутинг, контроллеры, модели, Blade-шаблоны).
* Создание frontend-части с использованием JavaScript для динамических элементов (лайки, избранное, тема).
* Реализация системы аутентификации и авторизации.
* Добавление функционала для администратора (управление категориями и ингредиентами).
* Внедрение темной/светлой темы с сохранением выбора в localStorage.
* Тестирование и отладка приложения.

Проект разрабатывался индивидуально, включая все этапы:

* **Backend-разработка (Laravel):**
  + Проектирование базы данных и миграций
  + Реализация CRUD для рецептов, комментариев, избранного
  + Настройка аутентификации и ролей (пользователь/админ)
  + Динамические элементы без перезагрузки страницы (лайки, добавление в избранное)
  + Реализация переключения темы (интеграция с localStorage)
  + Тестирование и деплой
* **Frontend-разработка (Blade + JS):**
  + Верстка страниц (главная, рецепты, профиль, админ-панель)
  + Валидация форм на стороне клиента
  + Настройка темной/светлой темы (localStorage)
  + Оптимизация UX/UI

В результате проделанной работы были успешно достигнуты все поставленные задачи.

Разработанная система может быть использована как основа для публичного кулинарного сервиса. Проект легко масштабируется и может быть дополнен мобильным приложением, рекомендациями по интересам, пошаговыми видео. Также возможна интеграция новых ролей, таких как модератор или редактор.

Веб-приложение «Рецепты блюд» реализует поставленные в дипломе цель и задачи, обеспечивает удобный доступ к кулинарному контенту, а его архитектура позволяет масштабировать систему и расширять функциональность в будущем.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Laravel Documentation — Официальная документация по фреймворку Laravel. URL: <https://laravel.com/docs>
2. Blade Templates — Шаблонизатор Laravel. URL: https://laravel.com/docs/blade
3. PHP Manual — Официальная документация языка PHP. URL: <https://www.php.net/manual/ru/>
4. JavaScript Guide — Руководство по JavaScript от Mozilla. URL: <https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide>
5. HTML Living Standard — Спецификация HTML. URL: <https://html.spec.whatwg.org/>
6. CSS Reference — Справочник по CSS от MDN. URL: <https://developer.mozilla.org/ru/docs/Web/CSS/Reference>
7. MySQL Documentation — Официальная документация по СУБД MySQL. URL: <https://dev.mysql.com/doc/>
8. Web-программирование для всех. — СПб.: БХВ-Петербург, 2008. — 560 с.: ил. (Автор: Вадим Дунаев) (Открытый)
9. Уроки MySQL. Самоучитель. — СПб.: БХВ-Петербург, 2006. — 208 с.: ил. (Автор: Анатолий Мотев) (Открытый)
10. Дэнни Гудман, Майкл Моррисон. JavaScript. Библия пользователя. - СПб: Вильямc, 2006. - 1184 с.