

ТЗ "Кулинарный блог"

1. Введение

1.1 Цель проекта

Создать современное веб-приложение для кулинарного сообщества, позволяющее пользователям публиковать рецепты, вести собственные блоги, комментировать и оценивать записи, а администраторам - управлять сайтом, контентом и пользователями.

1.2 Область применения

Сайт ориентирован на:

- Обычных пользователей - чтение, публикация, оценка и комментирование рецептов.
- Авторов блогов - ведение личного кулинарного блога.
- Администраторов - модерация контента.

2. Функциональные требования

2.1 Регистрация и авторизация

- Регистрация по email и паролю, подтверждение через email.
- Авторизация по логину и паролю, поддержка восстановления пароля.
- Опционально - поддержка OAuth-авторизации (Google/Facebook).

2.2 Пользовательская часть

- Главная страница: подборка популярных и новых рецептов, блок топовых авторов.

- Рецепты: просмотр, поиск, фильтрация по категориям, ингредиентам, времени приготовления.
- Блоги: создание, редактирование, удаление блога; просмотр блогов других пользователей.
- Рецепты внутри блога: CRUD-операции (создание, редактирование, удаление, публикация).
- Комментарии: добавление, редактирование, удаление своих комментариев.
- Рейтинг: оценка рецептов по 5-балльной системе, отображение среднего рейтинга.
- Личный кабинет: управление профилем, смена пароля, просмотр статистики.

2.3 Административная часть

- Управление пользователями: добавление, редактирование, удаление, добавление в черный список.
- Управление блогами и рецептами: модерация контента (редактирование, удаление, блокировка пользователя).
- Управление комментариями: удаление комментариев, не соответствующие правилам сайта.
- Статистика: количество пользователей, активность на сайте, популярные рецепты.

3. Архитектура

3.1 Frontend

Технологии:

- HTML5, CSS3, JavaScript.
- Framework: React.

Основные страницы и компоненты:

- Главная страница с топовыми рецептами и поиском.
- Страницы блогов и рецептов.
- Регистрация/авторизация.
- Личный кабинет пользователя.
- Интерфейс администратора.
- Поиск и фильтрация (по категориям, ингредиентам, рейтингу).

3.2 Backend

Технологии:

- Java.
- REST API.

Функционал:

- Авторизация/аутентификация.
- CRUD-операции для пользователей, блогов, рецептов, комментариев, рейтингов.
- API для поиска и фильтрации рецептов.
- API для административных операций.

4. База данных

Таблицы:

- Пользователи.
- Блоги.
- Рецепты.
- Комментарии.
- Оценки.

5. Дополнительные требования

5.1 UX/UI

- Современный адаптивный интерфейс (desktop + mobile).
- Удобная навигация, минималистичный дизайн.
- Опционально - поддержка темной/светлой темы.

5.2 Безопасность

- Хэширование паролей.
- Проверка прав доступа (роль: пользователь, администратор).
- Защита от XSS, CSRF, SQL-инъекций.

5.3 Производительность и масштабируемость

- Оптимизация запросов к БД (индексы).
- Кэширование популярных запросов.

5.4 Тестирование

- Unit-тесты для backend.
- Интеграционные тесты API.
- UI-тестирование (Selenium/Playwright).

Во время выполнения проекта могут вноситься правки об
применении технологии разработки веб-приложения