

# Требования к проекту

## 1. Введение

### 1.1. Название продукта

Медиа-менеджер (Media Manager) — это веб-приложение, предназначенное для каталогизации, отслеживания и обсуждения медиаконтента (фильмов, сериалов, книг, игр).

### 1.2. Контекст и описание продукта

Пользователи смогут создавать персональные коллекции, отслеживать статус потребления (например, "Запланировано", "В процессе", "Завершено"), выставять оценки, оставлять комментарии и просматривать медиа, добавленные другими пользователями и администраторами.

### 1.3. Границы проекта

1 Продукт будет предоставлять REST API для backend-логики и веб-интерфейс на Node.js для frontend.

2 Продукт будет поддерживать аутентификацию и авторизацию пользователей.

3 Продукт будет включать социальные функции: комментирование и публичные рейтинги.

4 Продукт будет иметь роль администратора для добавления и управления медиа в общедоступном каталоге.

5 Продукт не будет предоставлять возможность потоковой передачи медиаконтента (просмотр видео/чтение книг напрямую в приложении).

6 Продукт не будет являться пиринговым трекером для скачивания медиафайлов.

## 2. Требования пользователя

### 2.1. Программные интерфейсы

Backend: Java 17, Spring Boot (для REST API), Hibernate/JPA (для ORM), PostgreSQL (как основная СУБД).

Frontend: Node.js (с фреймворком, например, Express.js или Next.js) для рендеринга веб-интерфейса.

Взаимодействие: Frontend будет взаимодействовать с Backend через REST API.

Документация API: Swagger/OpenAPI для описания и тестирования эндпоинтов.

## **2.2. Интерфейс пользователя**

Интерфейс будет реализован как веб-приложение. Основные экраны включают:

- 1 Экран аутентификации: Вход и регистрация пользователя.
- 2 Главная страница: Лента с новыми и популярными медиа, рекомендациями.
- 3 Каталог медиа: Страница с фильтрацией и поиском по всем медиа (по названию, жанру, году).
- 4 Страница медиа: Детальная информация о конкретном медиа (описание, жанр, год), средний рейтинг, пользовательские оценки, секция комментариев. Здесь же пользователь может добавить медиа в свою коллекцию, изменить его статус, поставить оценку и написать комментарий.
- 5 Личный кабинет / Профиль пользователя: Просмотр и управление личной коллекцией (с фильтрацией по статусу), настройки учетной записи.
- 6 Панель администратора: Интерфейс для добавления новых медиа в общий каталог, редактирования и удаления существующих.

## **2.3. Характеристики пользователей**

### **2.3.1 Обычный пользователь**

Цель: организация личной медиатеки, отслеживание прогресса, общение с единомышленниками.

Опыт: ожидается базовое умение работать с веб-браузером. Опыт использования подобных сайтов (например, MyAnimeList, Shikimori, Goodreads) будет преимуществом, но не обязателен.

Техническая грамотность: средняя.

### **2.3.2 Администратор**

Цель: пополнение и модерация каталога медиа.

Опыт: опыт работы с системами управления контентом (CMS). Понимание основных принципов модерации.

Техническая грамотность: выше средней.

## **2.4. Предположения и зависимости**

- 1 Доступность и стабильность сервера базы данных PostgreSQL.
- 2 Наличие у пользователей современного веб-браузера с поддержкой JavaScript.
- 3 Разработка фронтенда на Node.js является приоритетной задачей после стабилизации backend API.
- 4 Проект зависит от выбранных фронтенд-фреймворков и библиотек (которые будут определены на этапе разработки UI/UX).

### **3. Системные требования**

#### **3.1. Функциональные требования**

1 Система должна позволять пользователям регистрироваться и входить в учетную запись.

2 Система должна предоставлять роль администратора с расширенными правами.

3 Система должна позволять администраторам добавлять, редактировать и удалять медиа (фильмы, сериалы, книги, игры) в общедоступном каталоге.

4 Система должна позволять пользователям добавлять медиа из общедоступного каталога в свою личную коллекцию.

5 Система должна позволять пользователям изменять статус медиа в их коллекции ("Запланировано", "В процессе", "Завершено").

6 Система должна позволять пользователям выставлять личную оценку (от 1 до 10) медиа в их коллекции.

7 Система должна рассчитывать и отображать средний рейтинг для каждого медиа на основе всех пользовательских оценок.

8 Система должна позволять пользователям оставлять текстовые комментарии к медиа.

9 Система должна предоставлять возможность поиска и фильтрации медиа в каталоге по названию, жанру, году выпуска и типу.

10 Система должна предоставлять пользователю возможность просматривать и фильтровать свою личную коллекцию по статусу и типу медиа.

#### **3.2. Нефункциональные требования**

##### **3.2.1. Атрибуты качества**

1 Надежность (Reliability): Система должна быть доступна 99.9% времени в течение рабочего дня. Ошибки сервера (5xx) должны возникать не более чем в 0.1% запросов. Важно для обеспечения непрерывного доступа пользователей к их коллекциям.

2 Масштабируемость (Scalability): Архитектура должна позволять горизонтальное масштабирование для обработки растущего числа пользователей и медиаобъектов. Важно для долгосрочного роста проекта.

3 Безопасность (Security): Все пароли должны храниться в хешированном виде. API эндпоинты, изменяющие данные, должны быть защищены от неавторизованного доступа.

4 Удобство использования (Usability): Веб-интерфейс должен быть интуитивно понятным, соответствовать современным стандартам UI/UX и корректно отображаться на основных desktop браузерах. Важно для привлечения и удержания пользователей.