Project Documentation

MovieMate

Submitted by:
Vaishnavi V Sirsat

1. Project Overview

MovieMate is a web-based application that allows users to manage and track their personal collection of movies and TV shows. It helps users organize their watchlist, track progress for TV shows, rate and review content, and explore personalized recommendations.

2. Objective

The primary objective of MovieMate is to provide a centralized platform for users to manage their entertainment content efficiently. The system enhances user experience by offering search, filter, and AI-driven recommendation features.

3. System Architecture

The system follows a client-server architecture. The frontend, built using ReactJS, communicates with the backend Django REST API over HTTP. The backend handles data storage, authentication, and business logic, while the database stores movie, TV show, and review data.

4. Tech Stack Used

- Frontend: ReactJS (Vite)
- Backend: Django with Django REST Framework
- Database: SQLite (for development) or MySQL/PostgreSQL (for production)
- API Testing: Postman
- Deployment: Vercel (Frontend), Railway (Backend)

5. Frontend Module (React)

The frontend is developed using ReactJS with Vite for faster builds. It includes routing, form handling, and dynamic API integration. The main pages are Dashboard, Add Media, and Media Details. Components such as MediaList, MediaCard, ReviewForm, and ProgressTracker enhance interactivity.

6. Backend Module (Django REST Framework)

The backend exposes RESTful APIs using Django REST Framework. It includes viewsets for Media and Reviews, a recommendation endpoint, and CRUD operations. The backend ensures data integrity, validation, and communication with the frontend via JSON responses.

7. Database Design

The database contains the following main entities:

- Media – title, director, genre, platform, status, progress
- Review – rating, notes, media reference
- User – optional (for personalization and recommendations)

8. API Endpoints

- /api/media/ – List or create media items
- /api/media/{id}/ – Retrieve, update, or delete media
- /api/reviews/ – Add or fetch reviews
- /api/recommend/ – Generate AI-based recommendations

9. Key Features

1. Add and manage movies/TV shows
2. Track progress for TV shows
3. Rate and review watched content
4. Filter by genre, platform, or status
5. AI-based recommendation system
6. Responsive UI with live data updates

10. Setup & Deployment Instructions

Backend Setup:

- Install dependencies: pip install -r requirements.txt
- Run migrations: python manage.py migrate
- Start server: python manage.py runserver

Frontend Setup:

- Navigate to Frontend/moviemate-frontend/
- Install dependencies: npm install
- Start app: npm run dev

Deployment:

- Frontend hosted on Vercel
- Backend deployed on Railway or Render

11. Future Enhancements

- Add authentication and user profiles
- Enable social media sharing for reviews
- Integration with TMDB API for automatic metadata fetching
- Watch party planner and notifications
- Analytics dashboard for viewing time trends

12. Conclusion

MovieMate demonstrates a complete full-stack implementation combining ReactJS and Django REST Framework. It showcases essential CRUD operations, clean UI, and scalable backend logic suitable for real-world media management and recommendation systems.