Movie Recommendation Web Application - Project Report

Name: Sanjeet Singh Solanki
Email: sanjeetsinghsolanki11@gmail.com
Project Type: Full Stack Web Application
Technologies Used: React, Node.js, Fastify, SQLite

1. Introduction

The Movie Recommendation Web Application is a full-stack project designed to suggest movies based on user preferences such as genre, mood, or similar movies.
The system integrates a frontend interface, backend API, and database storage to provide recommendations and maintain history.

2. Objective

The main objectives of this project are:
- To build a full-stack web application
- To integrate an external API for recommendations
- To store user queries and results in a database
- To implement proper error handling and fallback mechanisms

3. System Architecture

Frontend:
Developed using React and Vite. Allows users to enter movie preferences and displays recommended movies.

Backend:
Built using Node.js and Fastify. Processes API requests, handles recommendation logic, and saves results to the database.

Database:
SQLite is used for lightweight storage and stores user input, recommendations, and timestamps.

4. Features Implemented

- Movie recommendation based on user input
- Data saved in SQLite database
- History endpoint to view saved recommendations
- Error handling and fallback mechanism when API is unavailable
- Clean and responsive UI

5. Database Design

Table: Recommendations
Fields:
- id (Primary Key)
- user_input
- recommended_movies
- timestamp

This structure allows tracking of all past recommendations.

6. API Endpoints

GET /health
Checks whether backend is running.

POST /recommendations
Returns movie recommendations based on user input.

GET /history
Returns stored recommendation records.

7. Error Handling

The system handles invalid input, API errors, and quota limitations.
If the external API is unavailable, fallback recommendations are returned so that the application continues to function normally.

8. Tools and Technologies Used

Frontend:
React, Vite, CSS

Backend:
Node.js, Fastify

Database:
SQLite

Other Tools:
dotenv, OpenAI API

9. Testing

The application was tested for:
- Input validation
- Backend API responses
- Database insertion
- UI responsiveness

10. How to Run the Project

Backend:

```
cd backend
npm install
npm run dev
```

Frontend:

```
cd frontend
npm install
npm run dev
```

11. Future Improvements

Possible enhancements include:

- Movie posters and ratings
- User authentication
- Cloud deployment
- Improved recommendation logic

12. Conclusion

This project demonstrates the development of a complete full-stack application including frontend, backend, API integration, and database storage.
It showcases practical knowledge of building scalable and user-friendly web applications.