NATIONAL INSTITUTIONAL RANKING FRAMEWORK
NIRF-2024
Engineering Rank Band (151-200)
Pharmacy Rank - 77
Innovation Rank Band (11-50)

www.kiet.edu
Delhi-NCR, Ghaziabad

KIET GROUP OF INSTITUTIONS
Connecting Life with Learning

CELEBRATING 25 YEARS OF EXCELLENCE

Accredited by
NAAC with Grade "A+"

IQAC

Mini Project (KCA353)

Odd Semester

Session 2024-25

<MovieGpt>

<Archi Goel> 2300290140036>

Project Supervisor:

Amit Kumar Gupta

Professor

Content

- Introduction (1 slide)
- Literature Review (2 slides)
- Objective of the Project (1 slide)
- Technology

- Hardware Requirements (Development Environment, Server requirement (if required), Client requirement (if required).
- Software Requirements (Language and Platforms like Frameworks, VS code, Android Studio and Jupyter notebook etc.)

- Modules (2-3 slides)
- Workflow (1 slide)
- Reports (For Example: Project : Student Monitoring System, so reports like: Student Marks, Subjects, companies visit, and student appears in placement etc.)
- References (1 slide)

Introduction

- **MovieGPT** is an innovative platform designed to redefine the way users discover and explore movies. Powered by advanced AI and Natural Language Processing (NLP), the platform offers a seamless movie exploration experience by allowing users to search for films based on personalized queries or prompts, such as "comedy movies with a twist" or "inspirational dramas."
- The **homepage** serves as an interactive hub, showcasing trailers of trending movies, a curated list of upcoming releases, and a display of top-rated films. On the **GPT Search page**, users can interact with the platform by typing customized prompts to find movies that fit their unique preferences, eliminating the need for endless scrolling through generic lists.
- This project leverages **React** for a responsive frontend, **Firebase** for real-time backend services, and **APIs** to fetch movie data, delivering a smooth and engaging user experience. Whether you're looking for a light-hearted comedy or a gripping thriller, MovieGPT is designed to guide users to their next favorite movie effortlessly.

Literature Review

- **1. Research on Existing Movie Databases or Platforms:**
  Several existing platforms such as **IMDb**, **Netflix**, and **Rotten Tomatoes** dominate the movie database and recommendation space. These platforms provide:
- Basic search functionality based on predefined categories such as genre, rating, or popularity.
- Personalized recommendations based on watch history or user ratings.
- Aggregated ratings and reviews from users and critics.

2. **How MovieGPT Improves Upon Existing Solutions:**
   MovieGPT addresses the limitations of traditional platforms by integrating **AI-driven conversational search** and user-centric features:

- **Conversational Search:** Users can type natural language prompts such as "movies about time travel with a romantic subplot" for highly specific results.
- **Dynamic Recommendations:** Combines user input with trending data to provide personalized and current suggestions.
- **Intuitive Interface:** A modern, visually engaging homepage with trailers and categorized sections for upcoming and top movies enhances user engagement.
- **Real-Time Data Updates:** Using Firebase Firestore ensures that movie data and user interactions are always up-to-date

Literature Review (Contd.)

Feature	**Existing Platforms** (e.g., IMDb, Netflix)	**Limitations**
Basic Search Options	Genre, actor, director-based searches	Unable to process custom queries like "comedy movies with a twist."
Personalized Recommendations	Watch history-based recommendations	Restricted to past behavior, lacking adaptability for dynamic user preferences.
User Engagement	Ratings, reviews, and watchlists	Limited interactivity and no conversational search options.
Data Presentation	Static lists and ratings	Overwhelming with generic lists; no tailored user experience.

Objective of the Project

- The **MovieGPT** project aims to revolutionize how users discover and explore movies by combining cutting-edge AI with an intuitive and user-centric platform. The specific objectives of the project are:

- **1. Enhanced User Experience:**
- Develop an intuitive and visually appealing interface where users can effortlessly browse movies.
- Implement features such as **trailers**, **upcoming movies**, and **top-rated movies** on the homepage for a seamless experience.
- **2. Conversational Movie Search**
- Enable users to interact with the system through **natural language prompts**, such as:
  - "Find comedy movies with unexpected twists."
  - "Suggest action thrillers similar to Mission Impossible."
- Provide personalized search results that cater to unique user preferences.

3. Dynamic Recommendations:

- Utilize AI and Firebase Firestore to offer recommendations based on:
  - User input.
  - Popular trends and real-time data.
- Ensure adaptability to user needs and current movie industry trends.

Technology (Hardware Requirements)

- **Minimum Requirements:**
- **Processor**: Intel i5 (8th Gen) or AMD Ryzen 3
- **RAM**: 8 GB
- **Storage**: 256 GB SSD
- **Internet Speed**: 10 Mbps for efficient data synchronization and app testing
- **Recommended Requirements:**
- **Processor**: Intel i7 (10th Gen or higher) or AMD Ryzen 5/7
- **RAM**: 16 GB or higher for optimal performance
- **Storage**: 512 GB SSD or higher for faster load times and data management
- **Internet Speed**: 20 Mbps or higher for fast development and cloud-based services
- These configurations ensure smooth development, testing, and performance for the Cinema Plus app.

Technology (Software Requirements)

Technology	Purpose
React.js	Frontend development; creating a dynamic and interactive user interface.
Firebase Authentication	User authentication and management for secure access.
Firebase Firestore	Real-time database for storing and managing movie data and user preferences.
TMDB API	Fetching movie metadata such as titles, genres, and descriptions.
OpenAI GPT API	Implementing conversational AI for intelligent movie recommendations.
Node.js (optional)	Backend API layer for advanced operations and integrations, if needed.
CSS and Tailwind CSS	Styling and responsive design for user interface components.

Technology (Software Requirements)

Software	Version/Details
Operating System	Windows 10/11, macOS, or Linux (for development and deployment).
Web Browser	Latest version of Google Chrome, Firefox, or Edge.
Code Editor	Visual Studio Code (preferred) or any IDE of choice.
Node.js	Version 16 or above (for running React and npm).
npm (Node Package Manager)	Required for managing JavaScript dependencies.
Firebase CLI	Used for configuring and deploying Firebase services.
Postman	For testing API endpoints during development.

Modules

- 1. Login Page
- Purpose:
  To allow existing users to securely log in to the application.
- Features:
- Firebase Authentication for secure login.
- Login with email and password.
- 2. Registration Page
- Purpose:
  To allow new users to create an account.
- Features:
- User registration with email and password.
- Form validation for input fields (e.g., valid email, strong password).

Modules (Contd.)

- 3. Home Page
- Purpose:
  To serve as the main dashboard for browsing movies and accessing core functionalities.
- Features:
- A trailer of a recent or popular movie playing automatically.
- 4. GPT Search Page
- Purpose:
  To enable users to search for movies using natural language prompts.
- Features:
- Input field for custom prompts (e.g., "Suggest comedy movies with a twist").
- 5. User Profile Page
- Purpose:
  To manage user account details and preferences.
- Features:
- View and update account details (e.g., email, password).

Workflow/Gantt Chart

1. User Authentication:
•Users log in or register using Firebase Authentication.
•Verified users are redirected to the **Home Page**.
2. Home Page:
•Displays upcoming movies, top movies, and a featured movie trailer.
•Users can browse movies and access detailed information.
3. GPT Search Page:
•Users enter natural language queries (e.g., “Suggest thriller movies with a twist”).
•The system fetches relevant movie recommendations using OpenAI GPT and the TMDB API.
4. Movie Details Page:
•Displays detailed movie information, trailers, and options to add movies to "Watchlist" or "Favorites."
5. User Profile:
•Users can view and manage their account, "Watchlist," and "Favorites."
6. Admin Dashboard (Optional):
•Admins manage the movie database and analyze user activity and app performance.
7. Logout:
•Users can log out anytime, ending their session securely.

Reports

•**Introduction**:
MovieGPT is a web-based application designed to provide users with an intelligent platform for exploring movies,
•obtaining recommendations, and accessing detailed information such as genres, ratings, and summaries using
•AI-driven insights.
•**System Design**:
The architecture integrates **React.js** for the frontend, **Firebase** for the backend, and third-party APIs like **TMDB** for
• real-time movie data. OpenAI's GPT API is used for natural language processing.
•**Implementation**:
Focused on seamless user interaction with a responsive frontend, backend services utilizing Firebase Firestore, and
• secure user authentication with Firebase Auth.
•**Testing and Quality Assurance**:
Conducted unit, integration, and UI testing to ensure smooth functionality, along with performance testing to handle
•large datasets and real-time queries effectively.
•**Challenges and Solutions**:
Addressed challenges like API rate limits, real-time data updates, and personalized recommendations by optimizing
•API calls and using advanced caching strategies.
•**Conclusion**:
Successfully delivered MovieGPT with intelligent recommendations and a user-friendly interface, with plans for future
•features like offline capabilities, multi-language support, and social sharing.

References

1. **Books and Articles**
   - *"Learning React: Functional Web Development with React and Redux"* by Alex Banks and Eve Porcello
   - *"Firebase Essentials"* by Charles McKeever
2. **Online Documentation**
   - React Documentation: https://reactjs.org/docs/getting-started.html
   - Firebase Documentation: https://firebase.google.com/docs
   - React Router Documentation: https://reactrouter.com/
   - TMDB API Documentation: https://www.themoviedb.org/documentation/api
   - MDN Web Docs: https://developer.mozilla.org/
3. **Tutorials and Blog Posts**
   - *"How to Set Up Firebase Authentication in React"* - Medium Article
   - *"Build a React App with Firebase Firestore"* - DigitalOcean Tutorial
4. **Tools and Libraries**
   - React.js: https://reactjs.org/
   - Firebase: https://firebase.google.com/
   - Axios: https://axios-http.com/
   - Material UI: https://mui.com/