**Title:**

MOVIREC: A Movie Recommendation System

**Objective:**

The objective of the system is to suggest movies to users based on their preferences, mood and viewing history.

**Scope:**

The scope of a Movie Recommendation System Website mini project typically includes the following components:

**User Registration and Authentication:** Implement a user registration system with secure authentication mechanisms to allow users to create accounts and log in securely.

**User Profiles:** Enable users to create profiles where they can set preferences, provide feedback, and view their watch history and recommended movies.

**Movie Database:** Create and maintain a comprehensive database of movies with details such as title, genre, release year, cast, director, synopsis, and user ratings.

**Search Functionality:** Implement a search feature that allows users to search for movies based on various criteria like title, genre, actors, and directors.

**Recommendation Algorithm:** Develop and integrate recommendation algorithms, such as collaborative filtering, content-based filtering, or hybrid methods, to provide personalized movie suggestions to users.

**User Feedback System:** Include a mechanism for users to rate movies, provide reviews, and indicate their preferences. Use this feedback to refine the recommendation algorithms.

**User Interface (UI/UX):** Design an intuitive and user-friendly interface with features like movie categories, trending movies, user recommendations, and easy navigation. Ensure the website is responsive and accessible across different devices and screen sizes.

**Integration with External APIs:** Integrate external APIs to fetch additional movie data, including images, trailers, and reviews, enhancing the user experience.

**Content Management System (CMS):** Implement a CMS to manage and update the movie database, including adding new movies, editing existing entries, and removing outdated content.

**User Interaction Features:** Include features like user comments, movie sharing options, social media integration, and notifications to enhance user engagement and social interactions.

**Admin Panel:** Create an admin panel with functionalities to manage user accounts, monitor user activities, analyse user feedback, and perform necessary actions to maintain the platform's integrity.

**Data Security:** Implement security measures to protect user data, including encryption, secure connections (HTTPS), and secure storage practices.

**Performance Optimization:** Optimize the website's performance to ensure fast loading times, smooth user interactions, and efficient database queries.

**Testing and Debugging:** Conduct thorough testing, including unit testing, integration testing, and user acceptance testing, to identify and fix any issues or bugs in the system.

**Documentation:** Prepare comprehensive documentation covering system architecture, database schema, algorithms used, API integrations, and user guides for both users and administrators.

**Methodologies:**

1.Gather and Prepare Data:

2. Choose the Right Recommendation Algorithm

3. Development with Streamlit

4.User Experience (UX) Design

5. Testing and Iterative Development

6. Deployment on Heroku

7. Continuous Monitoring and Improvement

8.Documentation and Knowledge Sharing

9. Security and Privacy Considerations

**Proposed System for Movie Recommendation:**

**1. Introduction:**

**- \*Project Overview: \*** Create an intelligent movie recommendation system using Stream lit and deploy it on Heroku. The system will provide personalized movie recommendations based on user preferences and movie data obtained from external APIs.

**2. Features and Functionality:** - \*User Authentication: \* Allow users to create accounts, log in, and save their preferences for personalized recommendations.

- \*User Input: \* Implement an intuitive user interface where users can input their movie preferences, such as genres, actors, or ratings.

- \*Movie Recommendations: \* Utilize machine learning algorithms to generate movie recommendations based on user input and historical data.

- \*Real-time Movie Data: \* Fetch real-time movie information from external APIs (e.g., TMDB API) to ensure up-to-date movie details.

- \*User Feedback: \* Allow users to provide feedback on recommended movies to improve the recommendation algorithms.

- \*Responsive Design: \* Create a visually appealing and responsive user interface that adapts to various devices and screen sizes.

\*3**. System Architecture: \***

- \*Front-end: \* Develop the user interface using Streamlit, incorporating interactive components for user input and displaying movie recommendations.

- \*Back-end: \* Implement the recommendation algorithms and handle user data using Python, Flask (optional), and necessary libraries (e.g., Pandas, Scikit-learn).

- \*Database: \* Store user profiles, preferences, and feedback in a secure database (e.g., PostgreSQL) to personalize recommendations.

- \*External APIs: \* Integrate external APIs to fetch real-time movie data, ensuring the latest information for users.

**Features:**

**Implementation Plan:**

1. Project Setup and Planning:

   - \*Define Objectives

   - \*Gather Resources

   - \*Project Timeline

2. Data Collection and Preparation:

   - \*Choose Data Source

   - \*Data Gathering

   - \*Data Preprocessing

3. Streamlit App Development:

   - \*Front-end Interface

   - \*Integration with APIs

   - \*User Authentication

4. Recommendation Algorithm Implementation:

   - \*Choose Algorithms

   - \*Training and Testing

5. Database Setup and Integration:

   - \*Database Creation

   - \*Database Integration

6. Deployment and Hosting:

   - \*Heroku Setup

   - \*Deployment

   - \*Domain Configuration

7. Testing and Quality Assurance:

   - \*Unit Testing

   - \*User Testing

   - \*Bug Fixing

8. Continuous Integration and Deployment (CI/CD):

   - \*Set up CI/CD Pipeline

9. User Feedback and Iterative Improvement:

   - \*Feedback Collection

   - \*Algorithm Enhancement

10. Documentation and Knowledge Sharing:

   - \*Code Documentation

   - \*User Guides

   - \*Knowledge Sharing

11. Security and Compliance:

   - \*Security Audit

   - \*Privacy Compliance

12. Future Enhancements and Scaling:

   - \*Scaling Strategies

   - \*Future Features

**Team Members:**

Deepti Garg

Shivangi Srivastava

Prashant Bhadoria

Priyanshu Kumar

**Resources Required:**

1.Gather and Prepare Data:

2. Choose the Right Recommendation Algorithm

3. Development with Streamlit

4.User Experience (UX) Design

5. Testing and Iterative Development

6. Deployment on Heroku

7. Continuous Monitoring and Improvement

8.Documentation and Knowledge Sharing

9. Security and Privacy Considerations

**References:**

* <https://www.geeksforgeeks.org>
* <https://www.tutorialspoint.com>
* <https://www.javatpoint.com>
* <https://stackoverflow.com>
* <https://stackexchange.com>
* General Paper: Shani, G. & Gunawardana, A. (2011). "Evaluating recommendation systems", in Recommender systems handbook, ed: Springer, 2011, pp. 257-297
* General Paper:  Pazzani, M. J. & Billsus, D. (2007). "Content-based recommendation systems", in the adaptive web, ed: Springer, 2007, pp. 325-341

**Expected Outcomes:**

The Systemgenerates recommendations based on the user preferences, mood and viewing history.

**Project Supervisor:**

Mr. Sandeep Kumar Chhoker

**Conclusion:**

Recommendation systems have become an important part of everyone’s lives. With theonomous number of movies releasing worldwide every year, people often miss out on some amazing work of arts due to the lack of correct suggestion.  Putting machine learning based Recommendation systems into work is thus very important to get the right recommendations. We saw content-based recommendation systems that although may not seem very  effective  on  its  own,  but  when  combined  with  collaborative  techniques  can solve  the   cold   start   problems   that   collaborative   filtering   methods   face   when   run independently.  Similarly, such   systems can   be   improved   further by applying neural network embedding to uplift the quality of recommendations and make them more user

personalized