System Design Architecture for Movie Recommendation System

The Movie Recommendation System deployed on an AWS EC2 instance can be broken down into several key components, each responsible for specific tasks in the overall workflow.Overview of the system design architecture:

1. AWS EC2 Instance

- **EC2 Instance**: The core of the deployment, running a Linux operating system, where the application is hosted.

2. Data Storage and Processing

- **CSV Files**: Store movie metadata (e.g., movie descriptions, genres, cast, crew,overview,keywords,popularity score).
- **Pickle Files**: Store pre-computed similarity scores for efficient retrieval.

3. Application Backend

- **Python**: The primary programming language used for data processing and application logic.
- **Pandas**: Used for data manipulation and preprocessing.
- **NLP Techniques**: Advanced text processing techniques to extract insights from movie descriptions and reviews.
- **Machine Learning**: Content-based recommendation system using cosine similarity for suggesting movies.

4. Application Frontend

- **Streamlit**: Used to create an interactive web application for user interface.
- **Customised CSS**: Enhances visual appeal and usability of the UI components.

5. User Interaction

- **Interactive UI Components**: Allow users to explore recommendations based on different features like genres, cast members, and crew roles.

6. Version Control

- **GitHub**: Used for version control and hosting the project repository.

7. Workflow

1. **Data Ingestion**: Movie metadata is read from CSV files.
2. **Data Processing**: Text processing and similarity score calculation using NLP techniques.

3. **Similarity Computation**: Cosine similarity is used to compute similarity scores between movies.
4. **Recommendation Engine**: Based on user preferences, the system fetches similarity scores from pickle files and recommends movies.
5. **UI Rendering**: Streamlit renders the UI, allowing users to interact with the recommendation system.

8. Deployment Architecture Diagram

Below is a high-level architecture diagram: