

1. Project Overview : Job Recommendation Service

The **Recommendation Service** provides personalized item recommendations to users based on their past interactions, preferences, and contextual data. It uses a combination of machine learning models, such as collaborative filtering and content-based filtering, to generate relevant suggestions. The service is built with a focus on scalability and real-time responsiveness, allowing it to cater to various use cases like e-commerce, streaming services, or personalized content delivery.

2. Technologies Used

1. **Python 3.11**: The main programming language for developing the application.
2. **FastAPI**: A modern, fast (high-performance) web framework used for creating the API.
3. **Uvicorn**: An ASGI server to run the FastAPI application.
4. **Numpy & Pandas**: Libraries for numerical and data manipulation, helping with feature extraction and data preprocessing.
5. **Scikit-learn**: For implementing machine learning algorithms (e.g., collaborative filtering, content-based filtering).
6. **Alembic**: For database migrations and versioning control.
7. **PostgreSQL**: The primary database for storing user data, item metadata, and interaction history.
8. **Psycopg2**: PostgreSQL database adapter for Python, with **psycopg2-binary** for easier installation in containers.
9. **Docker**: Used for containerization, ensuring that the service can be deployed consistently across environments.

3. Service Logic

1. Data Input

- Collects **user interaction data** (views, clicks, purchases, ratings), **preferences**, and **contextual information**.
- Data is processed and transformed into feature vectors that are suitable for model predictions.

2. Recommendation Engine

- **Collaborative Filtering**: Suggests items based on what similar users have liked or interacted with.

- **Content-Based Filtering:** Recommends items based on a user's past behavior and item attributes.
- **Hybrid Model:** Combines both collaborative and content-based methods for improved accuracy.

3. Inference Process

- Scores items based on their relevance to the user and ranks them accordingly.
- Applies business rules such as price and category filters, and then returns the top recommendations.

4. Logic

The recommendation logic works by comparing a user's profile, including their experience level, skills, location preferences, and job type, against available job postings in the database. First, the system retrieves all job postings and checks each one to see if it matches the user's experience level and if at least one of the user's skills aligns with the job's required skills. The job also needs to be located in one of the user's preferred locations and must match the user's preferred job type (e.g., full-time or part-time). If a job posting meets all these conditions, it is added to a list of recommendations. This matching process ensures that only the most relevant jobs are suggested to the user based on their preferences and qualifications. If no matches are found, an error is returned, and if any system errors occur during the process, an internal error is logged.