**1. Setup and Data Preparation**

**Install Required Libraries**

pip install pandas numpy scikit-learn flask

**Create courses.csv Dataset**

Create a CSV file courses.csv with the following sample data:

title,description

"Python for Beginners","Learn Python from scratch with hands-on examples."

"Machine Learning Basics","Introduction to ML concepts and algorithms."

"Advanced Python","Deep dive into Python features and libraries."

"Data Science with Python","Learn data analysis, visualization, and ML in Python."

"AI Fundamentals","Understanding the basics of AI and deep learning."

**2. Build Recommendation Model**

**Load Dataset and Preprocess Data**

import pandas as pd

import numpy as np

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

# Load course data

courses = pd.read\_csv("courses.csv")

# Convert text data to TF-IDF matrix

tfidf = TfidfVectorizer(stop\_words='english')

tfidf\_matrix = tfidf.fit\_transform(courses['description'])

# Compute cosine similarity matrix

cosine\_sim = cosine\_similarity(tfidf\_matrix)

**Define Recommendation Function**

def recommend(course\_title, num\_recommendations=5):

# Find index of the given course title

idx = courses[courses['title'] == course\_title].index[0]

# Get similarity scores with all courses

scores = list(enumerate(cosine\_sim[idx]))

# Sort courses based on similarity scores

scores = sorted(scores, key=lambda x: x[1], reverse=True)

# Select top recommended courses

recommended\_indices = [i[0] for i in scores[1:num\_recommendations+1]]

return courses.iloc[recommended\_indices]

# Test the function

print(recommend("Python for Beginners"))

**3. Develop Web API using Flask**

**Install Flask**

pip install flask

**Create app.py**

from flask import Flask, request, jsonify

import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

# Load dataset

courses = pd.read\_csv("courses.csv")

# Process text data

tfidf = TfidfVectorizer(stop\_words='english')

tfidf\_matrix = tfidf.fit\_transform(courses['description'])

cosine\_sim = cosine\_similarity(tfidf\_matrix)

# Initialize Flask app

app = Flask(\_\_name\_\_)

# Recommendation API

@app.route('/recommend', methods=['GET'])

def get\_recommendations():

course\_title = request.args.get('title')

if course\_title not in courses['title'].values:

return jsonify({"error": "Course not found"})

recommendations = recommend(course\_title).to\_dict(orient='records')

return jsonify(recommendations)

# Run server

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**4. Deploy the Application**

**Run Flask API Locally**

python app.py

**Test API using Browser or Postman**

http://127.0.0.1:5000/recommend?title=Python%20for%20Beginners

**Deploy on Cloud (AWS, Heroku, or Google Cloud)**

1. Initialize a Git repository:

git init

1. Create a requirements.txt file:

pip freeze > requirements.txt

1. Deploy using a cloud platform following their respective documentation.