pip install pandas numpy scikit-learn flask

import pandas as pd import numpy as np from sklearn.feature\_extraction.text import TfidfVectorizer from sklearn.metrics.pairwise import cosine\_similarity

def load\_data(): return pd.read\_csv("courses.csv")

def train\_model(): courses = load\_data() tfidf = TfidfVectorizer(stop\_words='english') tfidf\_matrix = tfidf.fit\_transform(courses['description']) cosine\_sim = cosine\_similarity(tfidf\_matrix) return courses, cosine\_sim

def recommend(course\_title, num\_recommendations=5): courses, cosine\_sim = train\_model() idx = courses[courses['title'] == course\_title].index[0] scores = list(enumerate(cosine\_sim[idx])) scores = sorted(scores, key=lambda x: x[1], reverse=True) recommended\_indices = [i[0] for i in scores[1:num\_recommendations+1]] return courses.iloc[recommended\_indices]

if **name** == "**main**": print(recommend("Python for Beginners"))

from flask import Flask, request, jsonify from recommendation import recommend

app = Flask(**name**)

@app.route('/recommend', methods=['GET']) def get\_recommendations(): course\_title = request.args.get('title') recommendations = recommend(course\_title).to\_dict(orient='records') return jsonify(recommendations)

if **name** == '**main**': app.run(debug=True)

python app.py

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

[ {"title": "Advanced Python", "description": "Deep dive into Python with OOP, multithreading, and performance tuning."}, {"title": "Machine Learning", "description": "Understand ML algorithms, data preprocessing, and model evaluation."} ]

flask pandas numpy scikit-learn gunicorn

web: gunicorn app:app

git init git add . git commit -m "Initial Commit" heroku create git push heroku main