from fastapi import FastAPI, Header, HTTPException

from pydantic import BaseModel

from typing import List, Optional

import random

app = FastAPI()

API\_KEY = "fidara\_secret\_key"

class RecommendationRequest(BaseModel):

user\_id: str

agent\_profile: dict

request\_text: str

top\_k: Optional[int] = 6

class FeedbackRequest(BaseModel):

user\_id: str

recommendation\_id: Optional[str]

item\_index: int

signal: str

comment: Optional[str] = None

agent\_profile: dict

@app.post("/api/recommendations")

def get\_recommendations(data: RecommendationRequest,

x\_api\_key: str = Header(...)):

if x\_api\_key != API\_KEY:

raise HTTPException(status\_code=401, detail="Invalid API key")

sample\_items = [

{"title": "Portable Espresso Maker",

"subtitle": "Compact under $150",

"rationale": "Great fit for travelers who value durability and quality.",

"source\_link": "https://example.com/espresso",

"tags": ["coffee","travel","budget"]},

{"title": "Trail Running Shoes",

"subtitle": "Neutral support, rugged outsole",

"rationale": "Ideal for durable, outdoor-focused users.",

"source\_link": "https://example.com/shoes",

"tags": ["running","outdoors","durable"]},

{"title": "Hi-Fi Starter Amp",

"subtitle": "Clean 40 W per channel",

"rationale": "For craftsmanship and audio lovers.",

"source\_link": "https://example.com/amp",

"tags": ["audio","hi-fi","craftsmanship"]},

{"title": "Steel Drum Essentials",

"subtitle": "Afro-Caribbean playlist",

"rationale": "Matches island & jazz tastes.",

"source\_link": "https://example.com/playlist",

"tags": ["music","island","jazz"]}

]

random.shuffle(sample\_items)

return {"items": sample\_items[:data.top\_k]}

@app.post("/api/feedback")

def feedback(data: FeedbackRequest, x\_api\_key: str = Header(...)):

if x\_api\_key != API\_KEY:

raise HTTPException(status\_code=401, detail="Invalid API key")

# Simulate learning

updated\_profile = data.agent\_profile

updated\_profile["last\_feedback"] = data.signal

return {"agent\_profile": updated\_profile}