import requests

from bs4 import BeautifulSoup

YELP\_API\_KEY = "yelpkey"

OPENWEATHER\_API\_KEY = "e244ecc53b14661aff8120fbbc4edcca"

GOOGLE\_MAPS\_API\_KEY = " api\_key"

def get\_restaurant\_details(yelp\_url):

response = requests.get(yelp\_url)

soup = BeautifulSoup(response.content, 'html.parser')

name = soup.find("h1").text.strip()

address = soup.find("address").text.strip() if soup.find("address") else "N/A"

timings = soup.find("table", class\_="hours-table").text.strip() if soup.find("table", class\_="hours-table") else "N/A"

menu = {"Butter Chicken": 15.0, "Naan": 3.0}

return {"name": name, "address": address, "timings": timings, "menu": menu}

def find\_competitors(location, term="indian food", radius=2000):

url = "https://api.yelp.com/v3/businesses/search"

headers = {"Authorization": f"Bearer {YELP\_API\_KEY}"}

params = {"term": term, "location": location, "radius": radius, "sort\_by": "rating", "limit": 5}

response = requests.get(url, headers=headers, params=params)

data = response.json()

competitors = []

for business in data.get("businesses", []):

competitors.append({

"name": business["name"],

"rating": business["rating"],

"menu": {"Butter Chicken": 12.5, "Naan": 2.0}, # Placeholder for menu

})

return competitors

def get\_weather\_data(lat, lon):

"""Fetch current weather data from OpenWeatherMap."""

url = f"http://api.openweathermap.org/data/2.5/weather"

params = {"lat": lat, "lon": lon, "appid": OPENWEATHER\_API\_KEY, "units": "imperial"}

response = requests.get(url, params=params)

if response.status\_code == 200:

return response.json()

return {"error": "Failed to fetch weather data"}

def calculate\_final\_prices(menu, competitors\_menu, weather\_data, busy\_time):

"""Adjust menu prices based on conditions."""

final\_prices = {}

for item, base\_price in menu.items():

competitor\_prices = [comp["menu"].get(item, base\_price) for comp in competitors\_menu if item in comp["menu"]]

lowest\_price = min(competitor\_prices) if competitor\_prices else base\_price

if (

weather\_data["main"]["temp"] < 45 or "rain" in weather\_data["weather"][0]["main"].lower()

) and busy\_time:

final\_prices[item] = round(lowest\_price \* 1.15, 2) # 15% increase

else:

final\_prices[item] = lowest\_price

return final\_prices

def main():

yelp\_url = "https://www.yelp.com/biz/village-the-soul-of-india-hicksville"

lat, lon = 40.755, -73.982 # Replace with actual coordinates

print("Fetching restaurant details...")

restaurant\_data = get\_restaurant\_details(yelp\_url)

print("Finding competitors...")

location = restaurant\_data["address"]

competitors = find\_competitors(location)

print("Fetching weather data...")

weather\_data = get\_weather\_data(lat, lon)

if "error" in weather\_data:

print(weather\_data["error"])

return

busy\_time = True # Replace with live data from Google Maps

print("Calculating final prices...")

final\_prices = calculate\_final\_prices(restaurant\_data["menu"], competitors, weather\_data, busy\_time)

print("\n--- Final Prices ---")

for item, price in final\_prices.items():

print(f"{item}: ${price}")

if \_\_name\_\_ == "\_\_main\_\_":

main()