Lesson 4 Demo 02
Create AI Agents for Trip Planning

Objective: To design AI agents that work collaboratively to streamline the trip-planning process

Each agent is assigned a specific role, ensuring an organized and efficient approach to itinerary creation.
To manage the complexities of travel planning, responsibilities are distributed among the following specialized AI agents:

1. City selection expert
2. Local tour guide
3. Expert travel agent

Tools required: VSCode

Prerequisites: Complete Lesson 4 Prerequisite Demo and Lesson 4 Demo 1

Steps to be followed:

1. Create an agents.py file
2. Import dependencies
3. Define the TravelAgents class
4. Create an expert travel agent
5. Create a city selection expert
6. Create a local tour guide
7. Save the file

Step 1: Create an agents.py file

Note: Refer to Lesson 4 Demo 1 to create a *.py* file

Step 2: Import dependencies

```
from crewai import Agent
from textwrap import dedent
from langchain_openai import ChatOpenAI, AzureChatOpenAI

from tools.search_tools import SearchTools
from tools.calculator_tools import CalculatorTools
```

Step 3: Define the TravelAgents class

3.1 It is a class that initializes two OpenAI GPT models (GPT-3.5 and GPT-4) and defines methods to create travel-related agents.

```python
class TravelAgents:
    def __init__(self):
        """
        Initializes the TravelAgents class by setting up two OpenAI GPT models:
        - GPT-3.5 model (`gpt-3.5-turbo`) for generating text-based responses.
        - Both models are initialized with a temperature of 0.7 to control randomness in responses.

        The models will be used by the different agents (travel planner, city selection expert, and local to
        on their specific goals and tasks."""

        self.OpenAIGPT35 = AzureChatOpenAI(
            azure_endpoint="https://openai-api-management-gw.azure-api.net/deployments/gpt-4o-mini/chat/comp
            api_version="2023-12-01-preview",
            api_key="2ABecnfxzhRg4M5D6pBKiqxXVhmGB2WvQ0aYKkbTCPsj0JLKsZPfJQQJ99BDAC77bzfXJ3w3AAABACOGi3sC",
            azure_deployment="gpt-4o-mini",
            temperature=0.7
        )

        self.OpenAIGPT4 = AzureChatOpenAI(
            azure_endpoint="https://openai-api-management-gw.azure-api.net/deployments/gpt-4o-mini/chat/comp
            api_version="2023-12-01-preview",
            api_key="2ABecnfxzhRg4M5D6pBKiqxXVhmGB2WvQ0aYKkbTCPsj0JLKsZPfJQQJ99BDAC77bzfXJ3w3AAABACOGi3sC",
            azure_deployment="gpt-4o-mini",
            temperature=0.7
        )
```

Step 4: Create an expert travel agent

4.1 Set up the expert travel agent to help plan a 7-day trip
This agent can:

a) Build a detailed 7-day itinerary for the trip
b) Suggest a budget for the entire journey
c) Offer packing tips
d) Share important safety advice

4.2 The agent uses the following tools (as defined in the previous demo):

a) `SearchTools.search_internet`: To look up useful travel info online
b) `CalculatorTools.calculate`: To handle any necessary calculations (like costs)

4.3 The agent works with the GPT-3.5 model (`self.OpenAIGPT35`) to create responses and complete tasks.

4.4 This will return a fully configured expert travel agent.

```python
def expert_travel_agent(self):

    return Agent(
                role="Expert Travel Agent",
                backstory= f"""I'm a seasoned expert in travel planning and logistics. With decades of expe
                goal=f"""Plan a 7-day trip with day-by-day details, including a budget, packing tips, and s
                tools=[
                    SearchTools.search_internet,
                    CalculatorTools.calculate
                    ],
                verbose=True,
                llm=self.OpenAIGPT35,
            )
```

Step 5: Create a city selection expert

5.1 Create a city selection expert agent to help choose the best travel destinations

5.2 This agent can:

a) Analyze travel data to recommend cities based on various factors

b) Consider weather, season, prices, and the preferences of the traveler

5.3 The agent uses the tool:

a) `SearchTools.search_internet`: To gather city-related information online

5.4 The agent uses the GPT-3.5 model (`self.OpenAIGPT35`) to generate recommendations and respond to queries.

5.5 This will return a fully set-up city selection expert.

```
def city_selection_expert(self):

    return Agent(
                role="City Selection Expert",
                backstory=dedent(f"""A pro at analyzing travel data to pick the best destinations based on a
                                    variety of factors.""" ),
                goal=dedent(f"""
                    Find the top cities to visit, considering weather, seasons, budget, and what the travele
                ),
                tools=[SearchTools.search_internet],
                verbose=True,
                llm=self.OpenAIGPT35,
            )
```

Step 6: Create a local tour guide

6.1 Set up a local tour guide agent to provide detailed information about a city

6.2 This agent can:

a) Share insights about the city's top attractions, customs, and local highlights

b) Offer the best tips for experiencing the city like a local

6.3 The agent uses the tool:

a) `SearchTools.search_internet`: To find local information and details about the city

6.4 The agent uses the GPT-3.5 model (`` `self.OpenAIGPT35` ``) to provide responses and recommendations.

6.5 This will return a fully set-up local tour guide.

```
def local_tour_guide(self):

    return Agent(
            role="Local Tour Guide",
                backstory=dedent(f"""
                                A knowledgeable local guide who knows everything about the city's attracti
                                culture, and hidden gems."""),
                goal=dedent(f"""
                            Give the best insights and recommendations about the selected city for an unfor
                            experience."""),
                tools=[SearchTools.search_internet],
                verbose=True,
                llm=self.OpenAIGPT35,
            )
```

Step 7: Save the file