Phase 3: Development Part 1

Chatbot Persona:

- **Persona Name: "TravelBuddy"**

- **Persona Description: A friendly and knowledgeable travel assistant that helps users plan their vacations.**

Conversation Flow:

1. **Greeting**

   - **Welcomes the user and asks how it can assist.**

2. **User Intent Detection**

   - **Utilize intents to identify the user's primary goal, e.g., "Plan a trip," "Find a hotel," "Flight booking."**

3. **Handle User Queries**

   - **For each identified intent, create specific dialog nodes to handle related queries. For instance:**

   - **Intent: "Plan a trip"**

   - **Ask about destination, dates, and preferences.**

   - **Provide recommendations based on user input.**

   - **Intent: "Find a hotel"**

   - **Inquire about the city and dates.**

   - **Offer hotel options and their details.**

   - **Intent: "Flight booking"**

   - **Ask for departure and arrival cities, travel dates, and class preferences.**

- Display flight options and prices.

4. Entity Recognition

   - Use entities to extract specific details from user input, such as dates, locations, or travel class.

5. Confirmation and Feedback

   - After providing information or options, ask the user if they need anything else or if they want to proceed with a booking.

6. Handling User's Decision

      - If the user chooses to proceed, initiate the appropriate booking process.

      - If the user has more questions, return to the relevant intent-based dialog node.

7. Handling "Small Talk"

   - Create nodes for handling casual conversations or user comments that don't fit a specific intent.

8. Closing

      - Thank the user for using the chatbot's services and offer assistance for future inquiries.

Dialog Nodes:

      - Each dialog node corresponds to a specific step in the conversation flow. They trigger based on intents, entities, or specific user input.

- You can set conditions for when a node should trigger, define responses, and provide suggestions to guide the conversation.

Entities:

- Define entities for extracting specific information, e.g., @location, @date, @travel_class.

- Configure them to identify relevant values from user input.

Intents:

- Define intents to recognize the user's primary goals, e.g., "Book a flight," "Find a hotel," "Ask for travel advice."

- Train the chatbot with sample user queries for each intent to improve recognition.

Step 1: Create a Watson Assistant Service*

1. Log in to your IBM Cloud account or create one if you don't have an account.

2. Navigate to the IBM Cloud dashboard and create a new Watson Assistant service.

Step 2: Define the Chatbot's Persona

1.   Once you've created the Watson Assistant service, click on it to open the Watson Assistant tool.

2.   In the Watson Assistant tool, you can define the chatbot's persona. This includes its name, description, and the tone or style it should use in interactions.

For example:

 - **Persona Name: "TravelBuddy"**

 - **Persona Description: A friendly and knowledgeable travel assistant that helps users plan their vacations.**

 - **Tone: Helpful and conversational.**

Step 3: Design the Conversation Flow

1.   In Watson Assistant, use the "Skills" tab to create a new skill or edit an existing one.

2.   Create a dialog flow that represents the conversation between the chatbot and the user. Design the flow using a tree structure where each node corresponds to a specific part of the conversation. For example:

 - **Node 1: Greeting**

 - **Respond with a friendly greeting and ask how you can assist.**

 - **Node 2: User Intent Detection**

 - **Utilize intents to identify the user's primary goal, e.g., "Plan a trip," "Find a hotel," "Flight booking."**

 - **Node 3: Handle User Queries**

 - **Create nodes for each identified intent to handle related queries, including asking for specific information and providing responses.**

 - **Node 4: Entity Recognition**

 - **Use entities to extract specific details from user input, such as dates, locations, or travel class.**

 - **Node 5: Confirmation and Feedback**

- After providing information or options, ask the user if they need anything else or if they want to proceed with a booking.

- Node 6: Handling User's Decision

- If the user chooses to proceed, initiate the appropriate booking process or provide more details.

- Node 7: Handling "Small Talk"

- Create nodes for handling casual conversations or user comments that don't fit a specific intent.

- Node 8: Closing

- Thank the user for using the chatbot's services and offer assistance for future inquiries.

Step 4: Configure Intents, Entities, and Dialog Nodes

1.   In the "Skills" tab, navigate to the "Intents" section to define intents, e.g., "Book a flight," "Find a hotel," "Ask for travel advice." Train the chatbot with sample user queries for each intent to improve recognition.

2.   In the "Entities" section, define entities to extract specific information, e.g., @location, @date, @travel_class. Configure them to identify relevant values from user input.

3.   In the "Dialog" section, create dialog nodes for each part of the conversation flow. Set conditions for when a node should trigger, define responses, and provide suggestions to guide the conversation.   *Step 1: Set Up Your Environment*

Make sure you have the necessary Python libraries installed and have an IBM Cloud API key.

```python
import requests

# Replace with your IBM Cloud API key and Watson Assistant workspace ID api_key =
'YOUR_API_KEY' workspace_id = 'YOUR_WORKSPACE_ID'
```

Step 2: Define Intents

You can define intents and provide examples to train your chatbot.

```python
intents =
[
  {
    "intent": "greeting",
    "examples": ["Hello", "Hi", "Hey"]
  },
  {
    "intent": "book_flight",
    "examples": ["Book a flight", "I want to fly somewhere"]
  },
  # Add more intents and examples as needed
```

]

Step 3: Create Entities

Entities help extract specific information from user input.

python

```python
entities = [
    {
        "entity": "location",
        "values": [
            {
                "value": "New York",
                "type": "synonyms",
                "synonyms": ["NYC", "the Big Apple"]
            },
            {
                "value": "Los Angeles",
                "type": "synonyms",
                "synonyms": ["LA", "City of Angels"]
            }
        ]
    },
```

```
    # Add more entities and values as needed

]
```

Step 4: Define Dialog Nodes

Create dialog nodes to structure the conversation.

```python
python dialog_nodes =

[

  {

    "type": "standard",

    "title": "Greeting",

    "output": {

        "text": "Hello! How can I assist you today?"

    },

    "conditions": "#greeting"

  },

  {

    "type": "standard",

    "title": "Book Flight",

    "output": {

        "text": "Sure, I can help you book a flight. Where do you want to fly?"
```

```
    },

      "conditions": "#book_flight"

  },

  # Add more dialog nodes as needed

]
```

Step 5: Send Configuration to Watson Assistant

Use the API to send the configuration data to Watson Assistant.

```python
headers =
{
    'Content-Type': 'application/json',
    'Authorization': 'Bearer ' + api_key
}

url = f'https://api.us-south.assistant.watson.cloud.ibm.com/instances/YOUR_INSTANCE_ID/workspaces/{workspace_id}/intents?version=2021-09-16'

# Send intents configuration for

intent in intents:
```

```python
    response = requests.post(url, headers=headers, json=intent)    print(f'Intent
"{intent["intent"]}" created. Status code: {response.status_code}')


url = f'https://api.us-
south.assistant.watson.cloud.ibm.com/instances/YOUR_INSTANCE_ID/workspace
s/{workspace_id}/entities?version=2021-09-16'


# Send entities configuration for

entity in entities:

    response = requests.post(url, headers=headers, json=entity)    print(f'Entity
"{entity["entity"]}" created. Status code: {response.status_code}')


url = f'https://api.us-
south.assistant.watson.cloud.ibm.com/instances/YOUR_INSTANCE_ID/workspace
s/{workspace_id}/dialog_nodes?version=2
```