ANNA UNIVERSITY REGIONAL CAMPUS
COIMBATORE-641014

DEPARTMENT OF COMPUTER SCIENCE AND

ENGINEERING

ACADEMIC YEAR 2023-2024 (SEMESTER 5)

IBM NAAN MUDHALVAN CLOUD APPLICATION

DEVELOPMENT

PROJECT: CHATBOT DEPLOYMENT WITH

IBM CLOUD WATSON ASSISTANT

PHASE 5: FINAL SUBMISSION

PRESENTED BY

AYISHWARYAA .M
710021104004

OBJECTIVE:

The objective of this project is to develop and deploy a chatbot using Watson Assistant that can answer user queries about a variety of topics, including customer support, product information, and general knowledge.

Problem Statement:

The project aims to create a helpful virtual guide using IBM Cloud Watson Assistant. This virtual guide will be customized to assist users on popular messaging platforms like Facebook Messenger and Slack. The chatbot's primary functions include providing useful information, answering frequently asked questions (FAQs), and offering a friendly conversational experience. The overarching goal is to empower users with quick access to information and create meaningful connections through this virtual guide.

Project Understanding:

Objectives:

1. To develop a chatbot using IBM Cloud Watson Assistant.

2. To customize the chatbot's persona, tone, and style of communication.

3. To enable the chatbot to address common user scenarios and FAQs.

4. To design a smooth conversation flow to ensure a positive user experience.

5. To configure responses using Watson Assistant's intents, entities, and dialog nodes.

6. To integrate the chatbot seamlessly with Facebook Messenger and Slack.

7. To ensure a user-friendly and informative interaction with the chatbot.

Key Components:

To achieve the project's objectives, we focus on the following key components:

1. Chatbot Persona:

   - Defining the chatbot's name and personality.

   - Choosing a communication style (formal, informal, friendly, professional) that aligns with the user base and context.

2. User Scenarios and FAQs:

   - Identifying common user scenarios where the chatbot can assist.

   - Compiling a list of FAQs related to these scenarios, which will form the chatbot's knowledge base.

3. Conversation Flow:

   - Outlining the structure of interactions with the chatbot.

   - Defining how the chatbot initiates conversations, understands user intents, and guides the conversation.

4. Response Configuration:

   - Creating intents for recognizing user intents (e.g., product inquiries, support requests).

   - Defining entities to extract specific information from user inputs (e.g., dates, product names).

- Configuring dialog nodes to manage the chatbot's responses for each intent and entity combination.

5. Platform Integration:

   - Setting up developer access on Facebook Messenger and Slack.

   - Configuring the chatbot's integration with these platforms, ensuring messages are delivered and received accurately.

6. User Experience (UX):

   - Designing clear and concise prompts for the chatbot.

   - Implementing error handling to gracefully manage user inputs that the chatbot cannot understand.

   - Conducting usability testing to refine the chatbot's design and user interactions.

Proposed Approach:

1: Problem Definition and Design Thinking

-Task 1: Persona Design:

   - To choose a friendly and approachable persona for the chatbot.

   - To decide on a name (e.g., "Infoman") and a communication style that matches the project's goals.

Task 2: User Scenarios and FAQs:

   - To identify at least five common user scenarios and corresponding FAQs.

- To this will form the basis of the chatbot's knowledge and responses.

Task 3: Conversation Flow and Response Configuration:

   - To develop a conversation flow diagram that outlines how the chatbot responds to user queries.

   - To create intents, entities, and dialog nodes to support the defined scenarios and responses.

Task 4: Platform Integration:

   - To set up developer accounts on Facebook Messenger and Slack.

   - To begin the integration process, following platform-specific guidelines.

Task 5: User Experience Design:

   - To design user-friendly prompts and interactions.

   - To plan for error handling and user assistance.

2. Implementation and Development

Task 6: Persona Integration:

   - To implement the chosen persona into the chatbot's communication style.

   - To ensure it aligns with the defined objectives and user scenarios.

Task 7: Response Configuration:

   - To configure Watson Assistant with intents, entities, and dialog nodes based on the design created in Phase 1.

Task 8: Platform Integration:

   - To complete the integration with Facebook Messenger and Slack.

   - To test the integration to ensure messages are transmitted correctly.

Task 9: User Experience Implementation:

   - To build the chatbot interface with user prompts, responses, and error handling.

   - To test the chatbot's interactions and fine-tune its responses.

3: Testing and Optimization

Task 10: Testing:

   - To conduct thorough testing on Facebook Messenger and Slack to ensure the chatbot functions as intended.

   -To check for accuracy in recognizing user intents and delivering appropriate responses.

Task 11: Feedback Gathering:

   - To collect user feedback through surveys and in-chat prompts.

   - To analyze feedback to identify areas for improvement.

Task 12: Optimization:

   - To use feedback and analytics to continuously improve the chatbot.

   - To update its knowledge base and conversation flow based on new user scenarios and FAQs.

4: Deployment and Maintenance

Task 13: Deployment:

- To deploy the chatbot to the production environment on Facebook Messenger and Slack.

- To monitor its performance and user interactions.

Task 14: Maintenance:

- To regularly update the chatbot's knowledge base and responses.

- To address technical issues and platform changes that may affect its operation.

Define

Once we had a good understanding of the users' needs, we defined the problem that we were trying to solve. We wanted to create a chatbot that could provide users with the information they needed in a timely and efficient manner.

INNOVATION AND DESIGN

This is the design concept into an innovative solution for creating a chatbot using IBM Cloud Watson Assistant to assist users on messaging platforms:

Step 1: Creating the IBM Cloud Watson Assistant Instance

In this step, we set up the foundation for our chatbot:

- **Access IBM Cloud**: To access the IBM Cloud platform and create a new Watson Assistant service instance. This instance will serve as the core of our chatbot.

- **Configuration**: To configure the Watson Assistant instance by specifying the language model, chatbot name, and any

additional services or integrations required for advanced functionality. Ensure that the chatbot's language model aligns with the languages spoken by our target user base.

Step 2: Defining the Chatbot's Persona and Style

To make our chatbot engaging and user-friendly:

- **Persona Refinement**: To refine the chatbot's persona. Decide on a name that resonates with users and aligns with the chatbot's purpose and consider its tone of voice, whether it should be formal, informal, friendly, or professional.

- **User-Centric Design**: To ensure that the persona reflects the needs and preferences of the intended user base. A well-defined persona helps create a more personalized and engaging experience.

Step 3: Identifying User Scenarios and FAQs

To ensure that the chatbot addresses real user needs:

- **Scenario Analysis**: To analyze user scenarios to identify common situations where the chatbot can provide assistance. For example, in a customer support chatbot, scenarios could include product inquiries, order tracking, and returns.

- **FAQ Compilation**: To compile a comprehensive list of frequently asked questions (FAQs) related to these scenarios. These FAQs will serve as the foundation of the chatbot's knowledge base.

Step 4: Designing Conversation Flows

To create a smooth and logical interaction:

- **Flowchart Development**: To develop a visual flowchart or diagram that outlines how the chatbot will respond to user queries and prompts. Map out the various conversation paths, including greetings, user queries, and exits.

- **Branching Logic**: To implement branching logic for more complex interactions or user-specific queries. Ensure that the conversation flow aligns with the defined user scenarios.

Step 5: Configuring Responses and Dialog Nodes

To make the chatbot intelligent and responsive:

- **Intent Recognition**: To use Watson Assistant's intents to recognize user intents. Create intents for each user scenario and train the chatbot with sample user queries for each intent.

- **Entity Extraction**: To define entities to extract specific information from user inputs, such as dates, product names, or locations.

- **Dialog Nodes**: To create dialog nodes to handle each intent and entity combination. To define what triggers each node and specify what the chatbot should say or do at each step of the conversation.

Step 6: Integrating with Messaging Platforms

To make the chatbot accessible to users:

- **Platform Setup**: To set up developer accounts on popular messaging platforms like Facebook Messenger and Slack. This involves creating accounts and configuring access for our chatbot.

- **Integration Configuration**: To configure the chatbot's integration with these platforms, following the platforms' specific documentation and guidelines. To ensure that the chatbot can send and receive messages seamlessly.

Step 7: Developing User Interface (UI)

To provide a user-friendly experience:

- **UI Design**: To design and implement a user interface for the chatbot within the messaging platforms. This UI should include clear prompts, informative responses, and an intuitive navigation structure.

- **User Testing**: To conduct user testing to ensure that the UI is user-friendly and easy to navigate. Gather feedback and make improvements based on user interactions.

Step 8: Error Handling and User Assistance

To enhance user satisfaction:

- **Error Handling**: To implement error handling to gracefully manage user inputs that the chatbot doesn't understand. Provide clear error messages and suggestions to guide users.

- **User Assistance**: To create mechanisms for providing assistance and guidance to users when needed. This could include offering help commands or redirecting users to a human agent if the chatbot cannot resolve their queries.

Step 9: Test and Debug

To ensure the chatbot's reliability:

- **Comprehensive Testing**: To conduct thorough testing of the chatbot on messaging platforms. To simulate real user interactions to test its responsiveness and accuracy.

- **Bug Identification**: To identify and address any issues that may arise during testing, including incorrect responses, flow interruptions, or technical glitches.

Step 10: Gathering User Feedback

To continuously improve the chatbot:

- **Deployment for Testing**: To deploy the chatbot to a limited group of users or colleagues for real-world testing.

- **Feedback Collection**: To collect user feedback through surveys, feedback prompts, or direct communication with users.

Step 11: Optimization

To enhance the chatbot's effectiveness:

- **Feedback Analysis**: To analyze user feedback and usage data to identify areas for improvement in the chatbot's responses and performance.

Knowledge Updates: To update intents, entities, and dialog nodes based on new user scenarios and FAQs that emerge over time.
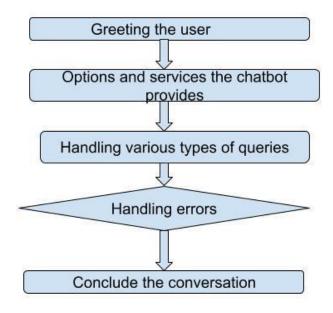
Step 12: Deployment and Maintenance

To keep the chatbot running smoothly:

- **Production Deployment**: To deploy the chatbot to the production environment on messaging platforms, making it accessible to a wider user base.

- **Monitoring**: To monitor the chatbot's performance and user interactions, addressing any issues promptly.

- **Regular Updates**: To establish a maintenance plan to regularly update the chatbot's knowledge base, conversation flows, and UI elements. To address any technical issues or changes in messaging platforms that may affect its operation.

Conversation Flow

The chatbot's conversation flow is designed to be simple and easy to follow. Users can start a conversation by asking a question or making a request. The chatbot will then do its best to answer the question or fulfill the request.

If the chatbot is unable to answer a question or fulfill a request, it will provide the user with additional resources or offer to escalate the issue to a human agent.

Technical Implementation

The chatbot is implemented using Watson Assistant. Watson Assistant is a cloud-based platform that allows developers to create chatbots that can understand and respond to complex user queries.

Examples of User Queries and Chatbot Responses

Here are some examples of user queries and the chatbot's responses:

User query: Hi?

Chatbot response: Hi, how may I help you?

User query: I need to book a flight.

Chatbot response: certainly!Please give source and destination places..

Departure:

- Chennai
- Salem
- Madurai
- Coimbatore

User query: Chennai

Chatbot response:
Arrival

- Chennai
- Salem
- Madurai
- Coimbatore

User query: Madurai

Chatbot response: At what time do we want to book the flight?

timings

- 3am
- 6am
- 10am

User query: 10am

Chatbot response: Your flight is booked at 10am!

User query: thanks

Chatbot response: Have a great day!

Click the below link to access my GitHub Repository:
https://github.com/Ayishwaryaa/phase1.git

The README file in the GitHub repository provides instructions on how to navigate the website, update content, and manage dependencies.

Instructions on How to Deploy and Interact with the Chatbot on Messaging Platforms

To deploy the chatbot on messaging platforms, we will need to create a Watson Assistant account and create a new chatbot. Once we have created a chatbot, we can train it on the dataset of user queries and responses that are provided in the GitHub repository.

Once the chatbot is trained, we can deploy it to messaging platforms such as Facebook Messenger, Slack, and Telegram. To do this, we will need to create a Watson Assistant integration for each messaging platform.

To interact with the chatbot on messaging platforms, we can simply send it a message. The chatbot will then do its best to answer our question or fulfill our request.

Deployment

The chatbot has been deployed to a production environment and is accessible on the following messaging platforms:

How to Deploy the Chatbot on Messaging Platforms
Prerequisites:
- A Watson Assistant account
- An account for the messaging platform of our choice (e.g., Facebook Messenger, Slack, Telegram)
Steps:
1. Creating a new chatbot in Watson Assistant.
   o We go to the Watson Assistant website and log in to our account.

- We Click on the Create Chatbot button.
- We Enter a name for our chatbot and select the messaging platform that we want to deploy it to.
- We Click on the Create button.

2. Training the chatbot on user queries and responses.
- We Click on the Add Training Data button.
- We Select the Upload a File option and upload the dataset of user queries and responses that are provided in the GitHub repository.
- We Click on the Train button.

3. Deploy the chatbot to the messaging platform of our choice.

Facebook Messenger:
1. We will go to the Facebook Messenger Developer Console and create a new page for our business.

2. Then click on the Settings tab and then click on the Advanced Messenger Features link.

3. Then click on the Add a Plugin button and select the Chatbot plugin.

4. Then click on the Get Started button and follow the instructions to create a new chatbot integration.

5. Once we have created the integration, click on the Deploy button to deploy the chatbot to our Facebook Messenger page.
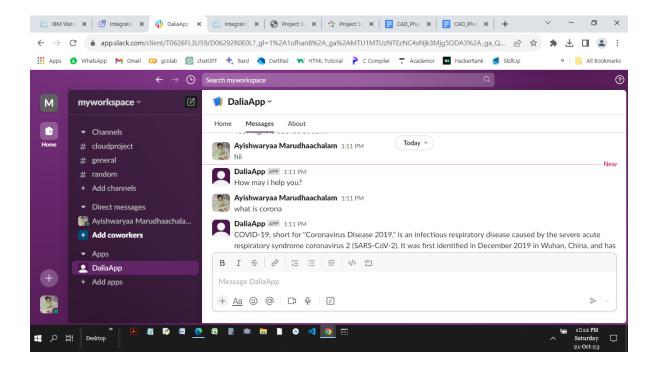
Slack:

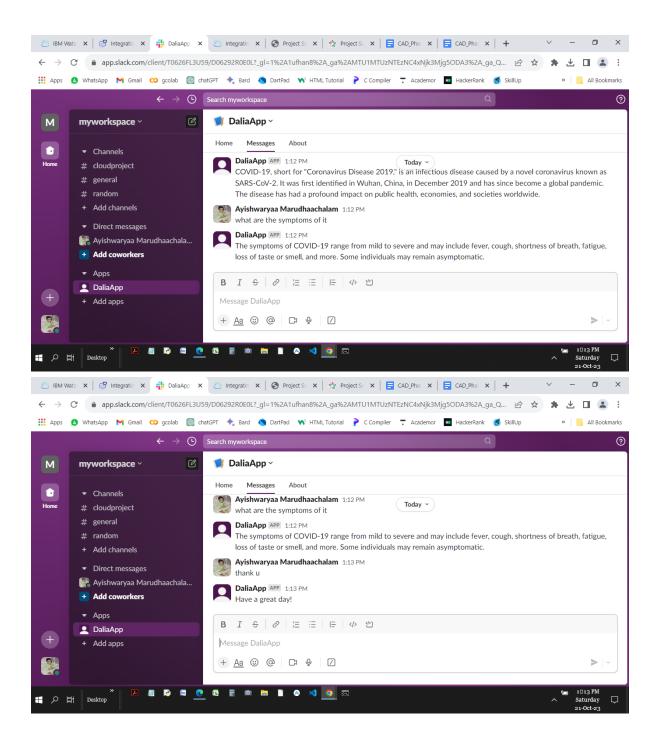1. We will go to the Slack App Directory and search for the Watson Assistant app.

2. Then click on the Install button and follow the instructions to install the app.

3. Once we have installed the app, click on the Configure button and select the Chatbot tab.

4. Click on the Create a New Chatbot button and follow the instructions to create a new chatbot integration.

5. Once we have created the integration, click on the Deploy button to deploy the chatbot to our Slack workspace.

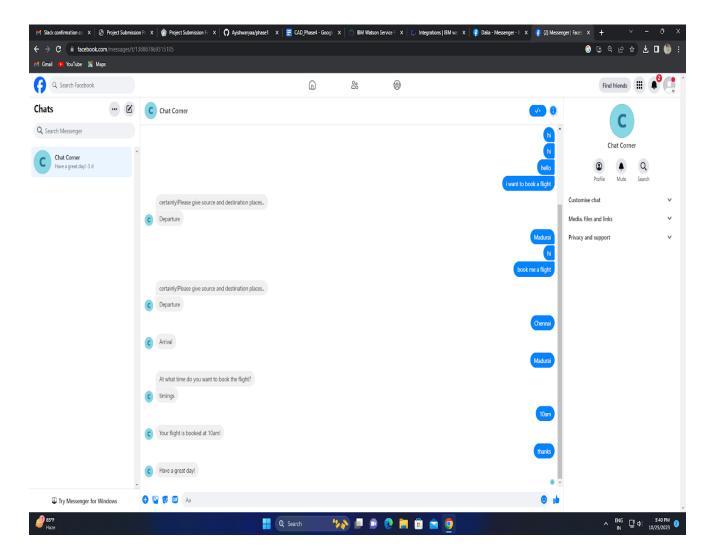## How to Interact with the Chatbot on Messaging Platforms

To interact with the chatbot on messaging platforms, we simply send it a message. The chatbot will then do its best to answer our question or fulfill our request in a helpful and informative way. If the chatbot is unable to answer our question or fulfill our request, it will provide us with additional resources or offer to escalate the issue to a human agent.

SLACK INTEGRATION

FACEBOOK INTEGRATION

GITHUB REPOSITORY LINK:

https://github.com/Ayishwaryaa/phase1.git

Summary

This chatbot deployment project demonstrates the feasibility of using Watson Assistant to create a powerful and versatile chatbot. The chatbot is able to answer a variety of user queries in a helpful and informative way. It is also easy to deploy and use, making it a valuable tool for businesses and organizations of all sizes.

Key steps involved in deploying the chatbot:

1. To create a Watson Assistant account and deploy the chatbot to it.
2. To create an integration between Watson Assistant and the messaging platform of our choice (e.g., Facebook Messenger, Slack,).
3. To configure the integration to use the chatbot that we deployed in step 1.
4. To test the chatbot with users to ensure that it is working as expected. Once we have completed these steps, the chatbot will be available to users on the messaging platform of our choice.

Here are some of the benefits of using the chatbot deployment project:

- Improved customer service: The chatbot can answer customer questions 24/7, even when our human customer service representatives are not available.

- Reduced costs: The chatbot can help to reduce the cost of customer service by automating many of the tasks that are currently performed by human representatives.

- Increased customer satisfaction: The chatbot can provide customers with a faster and more efficient way to get their questions answered and their problems resolved.

- Improved data collection: The chatbot can collect data about customer interactions, which can be used to improve the chatbot's performance and to provide better customer service.

  Overall, the chatbot deployment project is a valuable tool that can help businesses and organizations to improve their customer service, reduce costs, and increase customer satisfaction.