**ANNA UNIVERSITY REGIONAL CAMPUS COIMBATORE**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**PROJECT:CREATE A CHATBOT DEPLOYEMENT USING IBM WATSON ASSISTANT**

**SUBMITTED BY:**

**S.ABINAYASARASWATHI**

**710021106316**

**PHASE\_5**

**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.

**Objectives:**

1. Develop a chatbot using IBM Cloud Watson Assistant.

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

3. Enable the chatbot to address common user scenarios and FAQs.

4. Design a smooth conversation flow to ensure a positive user experience.

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

6. Integrate the chatbot seamlessly with Facebook Messenger and Slack.

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

**Key Components:**

1. **Chatbot Personal:**

Define the chatbot's name and personality.Choose a communication style (formal, informal, friendly, professional) that aligns with the user base and context.

2. **User Scenarios and FAQs:**

Identify common user scenarios where the chatbot can assist. Compile a list of FAQs related to these scenarios, which will form the chatbot's knowledge base.

**3. Conversation Flow:**

Outline the structure of interactions with the chatbot. Define how the chatbot initiates conversations, understands user intents, and guides the conversation.

**4. Response Configuration:**

Create intents for recognizing user intents (e.g., product inquiries, support requests). Define entities to extract specific information from user inputs (e.g., dates, product names).Configure dialog nodes to manage the chatbot's responses for each intent and entity combination.

**5. Platform** **Integration:**

Set up developer access on Facebook Messenger and Slac. Configure the chatbot's integration with these platforms, ensuring messages are delivered and received accurately.

**6. User Experience (UX):**

Design clear and concise prompts for the chatbot. Implement error handling to gracefully manage user inputs that the chatbot cannot understand. Conduct usability testing to refine the chatbot's design and user interactions.

**Proposed Approach:**

**Design Thinking**

-Task 1: Persona Design:

- Choose a friendly and approachable persona for the chatbot.

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

Task 2: User Scenarios and FAQs:

- Identify at least five common user scenarios and corresponding FAQs.

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

Task 3: Conversation Flow and Response Configuration:

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

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

Task 4: Platform Integration:

- Set up developer accounts on Facebook Messenger and Slack.

- Begin the integration process, following platform-specific guidelines.

Task 5: User Experience Design:

- Design user-friendly prompts and interactions.

- Plan for error handling and user assistance.

**2. Implementation and Development**

Task 6: Persona Integration:

- Implement the chosen persona into the chatbot's communication style.

- Ensure it aligns with the defined objectives and user scenarios.

Task 7: Response Configuration:

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

Task 8: Platform Integration:

- Complete the integration with Facebook Messenger and Slack.

- Test the integration to ensure messages are transmitted correctly.

Task 9: User Experience Implementation:

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

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

**3: Testing and Optimization**

Task 10: Testing:

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

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

Task 11: Feedback Gathering:

- Collect user feedback through surveys and in-chat prompts.

- Analyze feedback to identify areas for improvement.

Task 12: Optimization:

- Use feedback and analytics to continuously improve the chatbot.

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

**4: Deployment and Maintenance**

Task 13: Deployment:

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

- Monitor its performance and user interactions.

Task 14: Maintenance:

- Regularly update the chatbot's knowledge base and responses.

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

**INNOVATION PHASE:**

Certainly, let's break down the steps for creating a chatbot using IBM Cloud Watson Assistant and explain each one in detail:

**STEP 1: Define Your Use Case:**

Begin by clearly defining the purpose of your chatbot. What problem will it solve, or what tasks will it assist with? Understanding the specific use case is fundamental because it guides the design and development of your chatbot.

**STEP 2:IBM Watson Assistant Setup:**

Setting up Watson Assistant is the first technical step in building your chatbot. Sign up for an IBM Cloud account if you don't have one. IBM Cloud provides the infrastructure and services, including Watson Assistant, for your chatbot. Create an instance of IBM Watson Assistant through the IBM Cloud console. You can select the region (data center location) and pricing plan that suits your needs.

**STEP 3: Configure Watson Assistant:**

After creating your Watson Assistant instance, you need to configure it with basic settings. These settings include: Giving your Watson Assistant instance a name. Specifying the language and time zone. Defining the assistant's default response.Enabling or disabling system settings like small talk (casual conversation) and disambiguation (clarifying user queries when ambiguous).

**STEP 4: Data Collection and Training:**

To make your chatbot understand and respond to user queries effectively, you'll need to train it with data. This involves: Gathering a dataset of example user queries and corresponding responses. These examples will help train your chatbot's natural language understanding (NLU) model. Uploading this dataset to your Watson Assistant instance and using it to train the chatbot's language model.

**STEP 5: Design the Conversation Flow:**

Creating a conversational flowchart is essential to outline how users will interact with your chatbot. Design a user-centric, intuitive, and context-aware flow for the conversation. Define the paths the conversation can take, including user queries and bot responses.

**STEP 6: Integrate with Messaging Platforms:**

To make your chatbot available on messaging platforms like Facebook Messenger, Slack, or WhatsApp, you need to integrate it with these platforms using their APIs. This integration allows users to interact with your chatbot directly from these messaging apps.

**STEP 7: Multi-Channel Integration**:

Depending on your target audience, it might be beneficial to integrate your chatbot with multiple messaging platforms. Each platform may have unique features and user expectations, so consider these differences in your integration strategy.

**STEP 8: Natural Language Understanding (NLU):**

Enhance your chatbot's ability to understand user queries by utilizing Watson Assistant's NLU capabilities. Creating and fine-tuning intents (user intentions), entities (relevant pieces of information), and dialog nodes (chatbot responses) to improve the chatbot's accuracy in recognizing and responding to user queries.

**STEP 9: Personalization and Contextual Understanding:**

Personalization is crucial for providing a great user experience. Implement mechanisms to store and recall user context and preferences throughout the conversation, creating a more engaging and user-friendly experience.

**STEP 10: AI-Powered Recommendations:**

Employ AI and machine learning to offer personalized recommendations or solutions based on user input and behavior. This can enhance the value your chatbot provides to users.

**STEP 11: Analytics and Continuous Improvement:**

Track user interactions and collect data on chatbot performance. Use this data for continuous improvement, allowing you to refine your chatbot's responses and user experience over time.

**STEP 12: Security and Privacy:**

Ensure your chatbot complies with data privacy regulations and security standards. Take steps to protect user data and maintain trust. Security measures may include encryption, access controls, and compliance with data protection laws (e.g., GDPR).

**STEP 13: Feedback Loop:**

Encourage users to provide feedback on your chatbot's performance and usability. This feedback is invaluable for making ongoing improvements, fixing issues, and enhancing the user experience.

**CONVERSATIONAL FLOW:**

A conversational flow, in the context of a chatbot or virtual assistant, refers to the logical sequence of interactions and responses that occur during a conversation between the user and the chatbot. It outlines how the conversation progresses, the paths it can take, and how the chatbot responds to various user inputs. A well-designed conversational flow is crucial for creating a smooth and effective user experience when interacting with a chatbot. Here's an explanation of the key elements and principles of a conversational flow:

1. User-Centric Design: The conversational flow should be designed with the user's needs and goals in mind. It should anticipate user queries, preferences, and intentions to provide a seamless and relevant interaction.

2. Initiation: The conversation typically starts with an initiation message from the chatbot, greeting the user and setting the tone for the interaction. This message can be a welcome message or a prompt to guide the user's first input.

3. User Input: Users provide input through text or voice messages, and this input triggers the chatbot's responses. User input can take various forms, such as questions, commands, or statements.

4. Intent Recognition: Within the conversational flow, the chatbot uses natural language understanding (NLU) techniques to recognize the user's intent, which is the purpose or goal behind their input. NLU helps the chatbot understand what the user wants.

5. Entity Recognition: In addition to intent, the chatbot may also recognize entities within the user's input. Entities are specific pieces of information that are relevant to fulfilling the user's request. For example, in the sentence "Book a flight to New York," "New York" is the entity representing the destination.

6. Dialog Management: Dialog nodes are used to manage the conversation flow. These nodes contain the chatbot's responses, and they are organized in a tree-like structure. The chatbot selects the appropriate dialog node based on the user's intent and entities recognized.

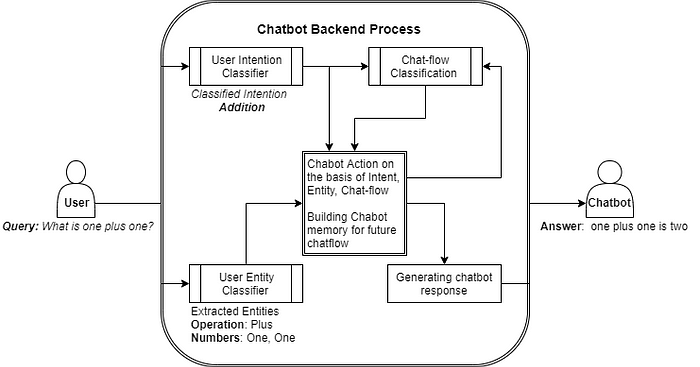
7. Context Management: Maintaining context throughout the conversation is essential for providing a coherent and natural interaction. Context management involves remembering previous interactions, user preferences, and ongoing topics to guide the conversation effectively.

8. Fallback and Error Handling: Chatbots should have mechanisms in place to handle user inputs that they don't understand or when they encounter errors. A fallback strategy involves providing a default response or asking the user to rephrase their input.

9. Prompts and Suggestions: To guide users and prompt them for specific actions or information, chatbots can use prompts and suggestions. These can be in the form of buttons or quick-reply options for the user to select.

10. User Satisfaction and Feedback: After completing the conversation, it's important to assess user satisfaction and gather feedback. This information can help improve the chatbot's conversational flow and overall performance.

**FLOW DIAGRAM OF CHATBOT:**

****

**TECHNICAL IMPLEMENTATION:**

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 can I help you.

User query:I need icecream.

Chatbot response:Select your flavour choice

1)vanilla ice cream

2)chocolate ice cream

3)strawberry ice cream

User query:option 1

Chatbot response:your vanilla ice cream order has been placed.

Bill:

Amount:250

Quantity:300g

User query: Thank you

**README File**

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

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

To interact with the chatbot on messaging platforms, you can simply send it a message. The chatbot will then do its best to answer your question or fulfill your 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 your choice (e.g., Facebook Messenger, Slack, Telegram)

**Steps:**

1. **Create a new chatbot in Watson Assistant.**
   * Go to the Watson Assistant website and log in to your account.
   * Click on the Create Chatbot button.
   * Enter a name for your chatbot and select the messaging platform that you want to deploy it to.
   * Click on the Create button.
2. **Train the chatbot on the dataset of user queries and responses.**
   * Click on the Add Training Data button.
   * Select the Upload a File option and upload the dataset of user queries and responses that are provided in the GitHub repository.
   * Click on the Train button.
3. **Deploy the chatbot to the messaging platform of your choice.**

Facebook Messenger:

1. Go to the Facebook Messenger Developer Console and create a new page for your business.
2. Click on the \*\*Settings\*\* tab and then click on the \*\*Advanced Messenger Features\*\* link.
3. Click on the \*\*Add a Plugin\*\* button and select the \*\*Chatbot\*\* plugin.
4. Click on the \*\*Get Started\*\* button and follow the instructions to create a new chatbot integration.
5. Once you have created the integration, click on the \*\*Deploy\*\* button to deploy the chatbot to your Facebook Messenger page.

**Slack:**

1. Go to the Slack App Directory and search for the \*\*Watson Assistant\*\* app.
2. Click on the \*\*Install\*\* button and follow the instructions to install the app.
3. Once you 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 you have created the integration, click on the \*\*Deploy\*\* button to deploy the chatbot to your Slack workspace.

**How to Interact with the Chatbot on Messaging Platforms**

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

The chatbot will do its best to answer your question or fulfil your request in a helpful and informative way. If the chatbot is unable to answer your question or fulfil your request, it will provide you with additional resources or offer to escalate the issue to a human agent.

**How to Navigate the Website**

The website for the chatbot deployment project should be divided into the following sections:

* Home: This section should provide a brief overview of the chatbot deployment project and links to the other sections of the website.
* Chatbot: This section should contain information about the chatbot, such as its persona, conversation flow, and technical implementation.
* Resources: This section should contain links to the GitHub repository, documentation, and other resources related to the chatbot.
* Blog: This section should contain articles about chatbot development, deployment, and best practices.

To navigate the website, simply click on the links in the header menu.

How to Update Content

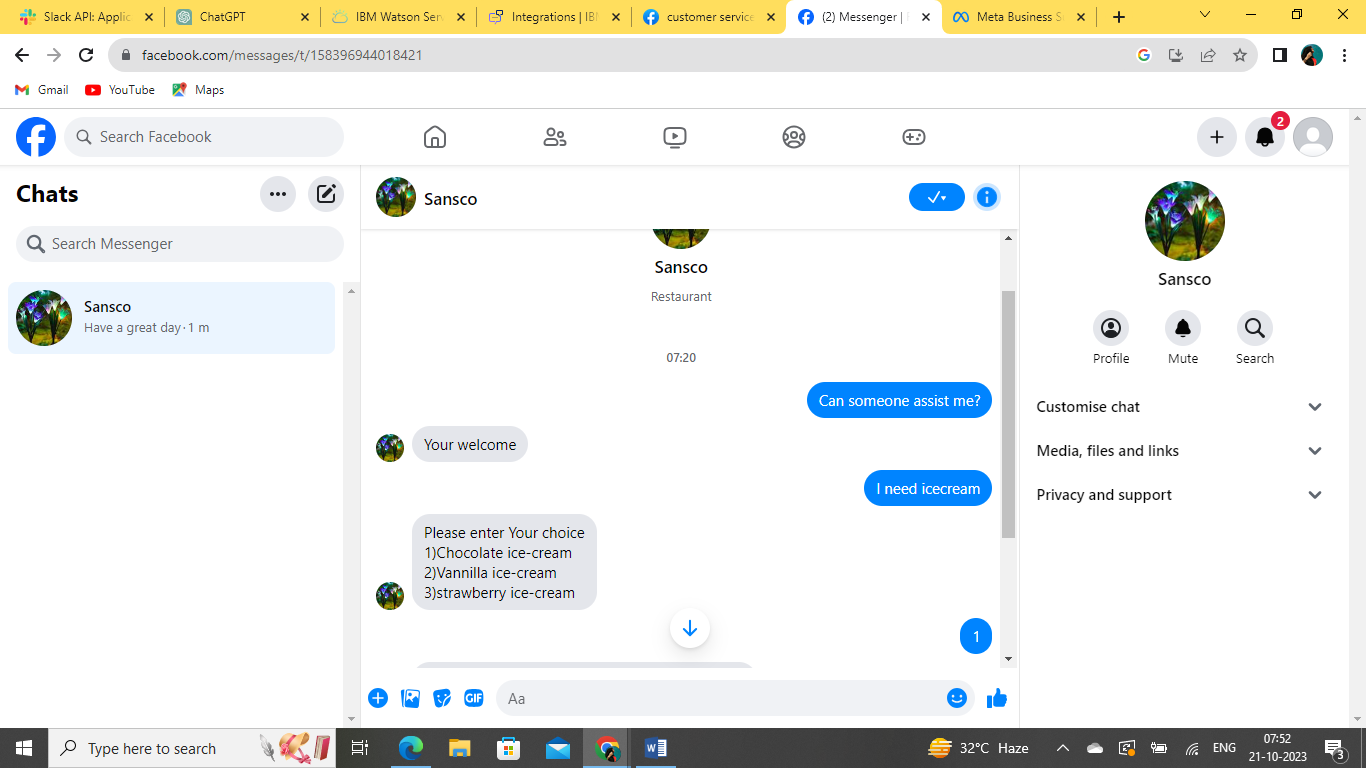
To update the content on the website, you will need to edit the HTML, CSS, and JavaScript files in the GitHub repository. Once you have made your changes, commit them to the repository and deploy them to the website.

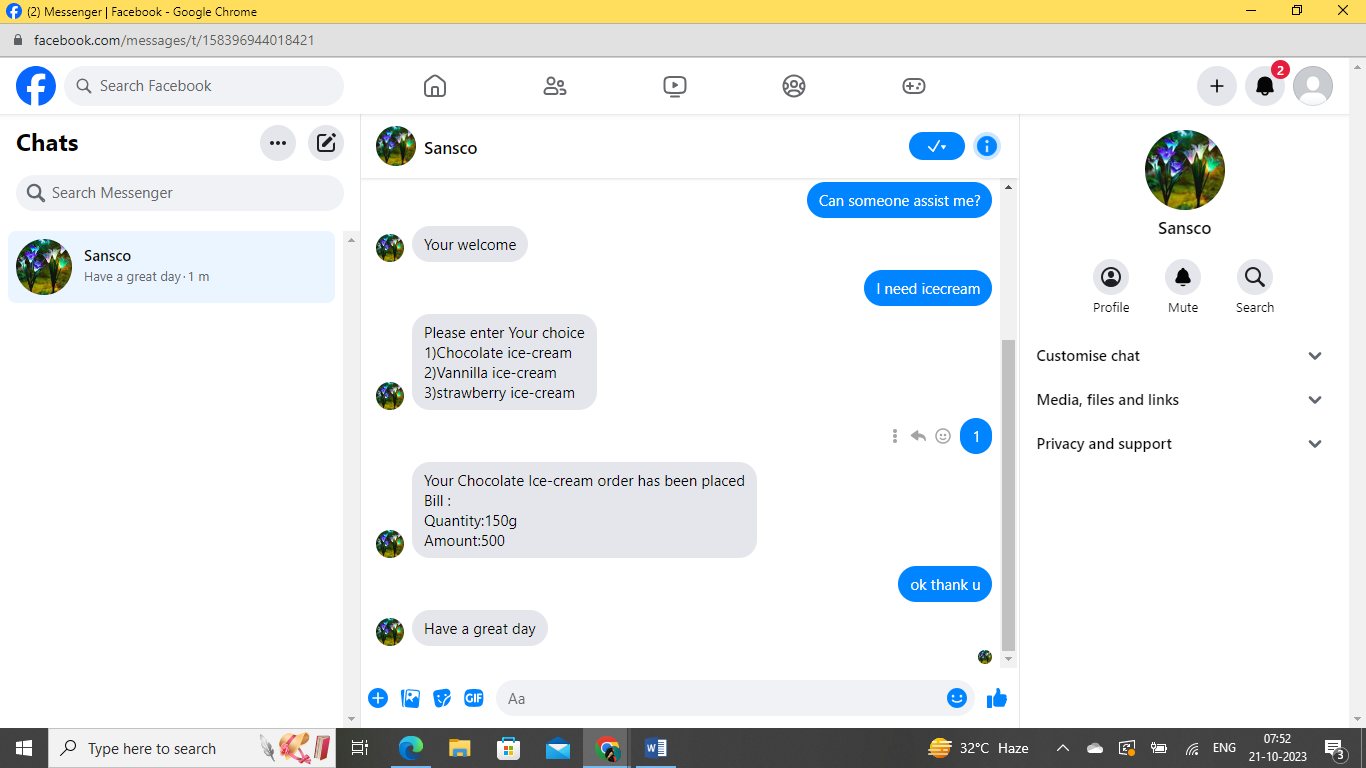
How to Manage Dependencies

The chatbot deployment project relies on the following dependencies:

* Watson Assistant Python SDK

**FACEBOOK MESSENGER:**

****

****

**GITHUB REPOSITORY LINK:**

**https://github.com/Abinayasaraswathis/Chatbot.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. Create a Watson Assistant account and deploy the chatbot to it.
2. Create an integration between Watson Assistant and the messaging platform of your choice (e.g., Facebook Messenger, Slack, ).
3. Configure the integration to use the chatbot that you deployed in step 1.
4. Test the chatbot with users to ensure that it is working as expected.

Once you have completed these steps, the chatbot will be available to users on the messaging platform of your 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 your 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.