Roll no – 2300290100151

Name–Meghna Chaudhary

Branch – Computer Science

and Engineering

Section – C

Year – 2nd

Domain – Generative AI

PROJECT- **Create a comprehensive NLP solution**:
(CUSTOMER SERVICE AI BOT)

Description - Use Azure AI Services to build and deploy an NLP solution for a real-world application.

To design and develop a web-based customer service chatbot that can understand customer queries, identify the intent behind them, and provide relevant responses. The bot uses Microsoft Azure's Conversational Language Understanding (CLU) model to recognize user intents and serve as an

automated support assistant.

1. Introduction

In the age of digital transformation, AI-powered chatbots have become essential for customer support, providing instant responses to common queries. This project introduces a customer service chatbot that leverages Natural Language Processing (NLP) capabilities via Microsoft Azure's Conversational Language Understanding (CLU) to recognize customer intent and respond accordingly. By integrating this AI model into a web-based interface, we created a user-friendly solution that offers 24/7 support for customer inquiries.

---

2. Key Features

- **Intent Recognition**: Identifies the purpose behind customer queries, such as "order status" or "refund policy," using Microsoft Azure's CLU model.
- **Automated Response**: Provides relevant responses to the identified intent, creating a seamless user experience.
- **Web-Based Interface**: Designed with HTML, CSS, and JavaScript, providing a simple yet effective frontend for users to interact with the bot.
- **Scalable**: Built on Azure's cloud infrastructure, which ensures scalability and reliability.

---

3. Tech Stack

Frontend

- **HTML**: Used to structure the web pages and form the layout of the chatbot interface.
- **CSS**: Styled the user interface, providing a visually appealing look to the chatbox, input fields, and responses.
- **JavaScript**: Handled client-side logic, including capturing user input, sending requests to the CLU model, and displaying responses.

3.2 Backend and AI Integration

- **Microsoft Azure Language Studio**: Managed the Conversational Language Understanding (CLU) project. The Language Studio allows training, testing, and deployment of the CLU model.
- **Azure Portal**: Configured resources such as the Language Studio project, API keys, and endpoints to facilitate communication between the chatbot and CLU model.
- **Conversational Language Understanding (CLU)**: Used for intent recognition, allowing the bot to process natural language queries and categorize them into predefined intents.

---

4. Workflow

1. **User Input**: A user types a query into the chatbox (e.g., "What is the status of my order?").
2. **Intent Recognition**: The chatbot captures the input and sends it to the Azure CLU model.
3. **CLU Model Processing**:
    - The CLU model, trained to recognize specific intents such as "order_status," "refund_order," and "complaint," processes the input.
    - It identifies the intent with the highest confidence score and returns it to the client-side JavaScript code.
4. **Response Generation**: The chatbot uses the identified intent to generate a response. A predefined set of responses is mapped to each intent.

5. **User Interaction**: The response is displayed in the chat interface, providing an answer or further instructions based on the query.

---

5. Project Setup and Configuration

Azure Language Studio and CLU Project

- **Project Creation**: Created a CLU project named “meghna7” in Language Studio, defining intents relevant to customer service (e.g., "order_status", "refund_order").
- **Intent Definition and Training**: Added example phrases for each intent to improve the model’s understanding.
- **Deployment**: Deployed the project under the name “product” to make it accessible through the Azure API.

Azure Resource Setup

- **Language Resource**: Configured a resource named "botmeghnaai07" in Azure Portal with Text Analytics API.
- **API Endpoint and Key**: Retrieved the endpoint and API key from the Azure Portal, required to authenticate API requests from the frontend.

---

6. Code Structure and Integration

1. **HTML Structure**: The web page contains an input field and a display area for the chat conversation.
2. **CSS Styling**: Styles applied to give a user-friendly and professional look to the chat interface.
3. **JavaScript Logic**:
    - Captures user input and adds it to the chat display.

- Makes API calls to the Azure CLU model, sending user input as the request body.
- Processes the CLU model’s response to extract the top intent.
- Matches the intent with a predefined response using a dictionary of possible replies.