CHATBOT APPLICATION USING JAVA

INTERN INFORMATION:

Name: Vineesh Pydisetti

ID: COD6919

INTRODUCTION

The Chatbot is a Java application that simulates a conversation with a user through a graphical user interface (GUI). It allows users to interact with a chatbot, answering questions and exchanging messages in a colorful and engaging way.

The project's scope is to demonstrate the use of Java Swing components to create an interactive chatbot application. The chatbot interacts with users, responding to their input with predefined messages. The chatbot's responses are tailored based on the conversation count, providing a structured and conversational experience.

The primary goal of this Java Chatbot Application is to demonstrate:

Natural Language Processing (NLP): Implementing basic rule-based techniques to understand and respond to user queries.

Graphical User Interface (GUI): Using AWT to design an interactive interface for the chatbot, enhancing user engagement and experience.

Event Handling: Employing ActionListener interface to capture and process user interactions (e.g., button clicks, text input).

Scope of the Project: The project aims to create a user-friendly chatbot that can interpret natural language input, respond appropriately, and provide a seamless conversational experience within a GUI environment.

Importance and Advantages

- **User Interaction:** The Colorful Chatbot provides users with a friendly and interactive chat experience.
- **Learning:** This project helps developers understand event handling, GUI design, and basic chatbot logic.
- **Customization**: The chatbot's interface and behavior can be extended and customized for various applications.

Main Advantage of Using AWT in Java

The main advantage of using AWT (Abstract Window Toolkit) in Java for this project is its simplicity and platform independence. AWT provides a basic set of GUI components that can be used to build graphical applications across different operating systems without external dependencies. This makes it ideal for developing lightweight and portable applications like our chatbot.

TECHNICAL SKILLS USED

Programming Language: Java

GUI Library: AWT (Abstract Window Toolkit)

Development Environment: VS Code

Natural Language Processing (NLP): Techniques employed to process and understand user input for generating appropriate responses.

Version Control: Git for managing and tracking project changes

IMPLEMENTATION

Chatbot: Main class representing the chatbot application.

sendMessage (String message): Method to append a message to the chat area.

actionPerformed (ActionEvent e): Event handler for user input.

CODE EXPLAINATION

The chatbot code is structured into:

GUI Setup: Initializes and configures the JFrame layout with JTextArea, JTextField, and JLabel components.

Event Handling: Implements ActionListener to respond to user input events.

Message Sending: Appends user messages to the chat area and triggers chatbot responses based on conversation count.

Chatbot Logic: Uses conversation count to determine the appropriate response at each step of the conversation.

GUI Deployment and Styling

- The chatbot GUI is deployed using AWT components arranged within a Frame and Panels to create a user-friendly interface.
- Styling includes setting font sizes, layout configurations, and interaction components (e.g., buttons, text fields).

FUNCTIONALITY OF THE TECHNOLOGIES

SendMessage: Adds a message to the chat area and scrolls to the bottom for visibility.

ActionPerformed: Processes user input, sends messages to the chat area, and controls the flow of conversation..

USAGE

Adding a Task: Typing queries in the input text field.

Marking a Task as Completed: Clicking the "Send" button to submit queries for processing.

Close the task: Observing chatbot responses displayed in the chat area of the GUI.

CONCLUSION

The Colorful Chatbot project demonstrates the creation of an interactive Java application using Swing components. It showcases how to build a basic chatbot interface and implement conversational logic. The project is a good starting point for learning GUI development and event-driven programming in Java.

EXECUTION AND OUTPUT

Execution:

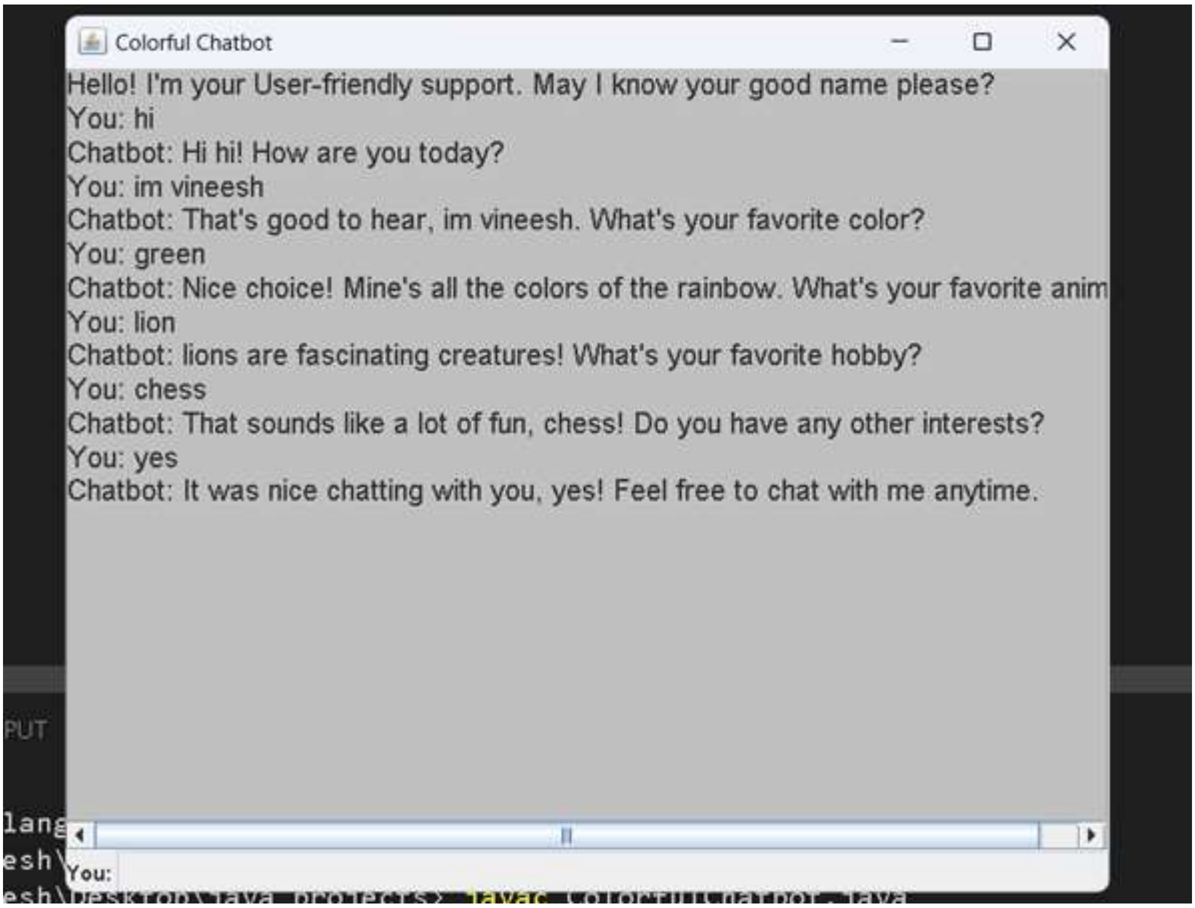
- Compile the Chatbot.java file.
- Run the compiled Chatbot class.

Output:

- The chatbot GUI window will appear.
- Enter your name when prompted.
- Engage in a conversation with the chatbot by responding to its questions.
- See messages displayed in the chat area with different colors for user and chatbot messages.
- After six prompts, further input will be disabled.

Output:

- Capture screenshots during Chatbot usage, showing different input scenarios and corresponding outputs.
- Include screenshots to illustrate the Chatbot interface and user interactions.



```
Hello! I'm your User-friendly support. May I know your good name please?
You: hi
Chatbot: Hi hi! How are you today?
You: im vineesh
Chatbot: That's good to hear, im vineesh. What's your favorite color?
You: green
Chatbot: Nice choice! Mine's all the colors of the rainbow. What's your favorite anim
You: lion
Chatbot: lions are fascinating creatures! What's your favorite hobby?
You: chess
Chatbot: That sounds like a lot of fun, chess! Do you have any other interests?
You: yes
Chatbot: It was nice chatting with you, yes! Feel free to chat with me anytime.
```

- Chatbot.jpg image displayed above
- Screenshot demonstrating the Simple Chatbot with AWT GUI during execution, showcasing user interactions and Chatbot interaction functionality.