Task:3

🔷 Step 1: Create a Java-based chatbot

We'll write the chatbot in Java. It can be console-based or GUI-based (Swing/JavaFX).

🔷 Step 2: Use NLP techniques

Since full NLP libraries (like Stanford NLP, OpenNLP) can be heavy for a class project, we can simulate NLP with keyword matching, tokenization, and intent detection.

🔷 Step 3: Implement ML logic or rule-based answers

For simplicity, start rule-based (hardcoded responses to keywords). Later you can extend with ML (e.g., train from FAQ dataset).

🔷 Step 4: Train the bot

We can store FAQs in a text/CSV file and let the bot learn responses dynamically.

🔷 Step 5: GUI/Web Interface

We’ll use Swing GUI (easier to demonstrate).

✅ Example: Rule-Based AI Chatbot with GUI in Java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;

public class AIChatBotGUI
extends JFrame {
    private JTextArea
chatArea;
    private JTextField
inputField;
    private Map<String,
String> knowledgeBase;
```

```
    public AIChatBotGUI() {
        setTitle("AI Chatbot");
        setSize(500, 500);

setDefaultCloseOperation(E
XIT_ON_CLOSE);
        setLayout(new
BorderLayout());

        // ==== Knowledge
Base (Training Data) ====
        knowledgeBase = new
```

```
HashMap<>();

knowledgeBase.put("hi",
"Hello! How can I help
you?");

knowledgeBase.put("hello",
"Hi there!");

knowledgeBase.put("how
are you", "I’m doing well,
thank you!");
```

```
knowledgeBase.put("what is your name", "I’m a Java AI Chatbot.");

knowledgeBase.put("bye", "Goodbye! Have a great day!");

        // ==== Chat Area ====
        chatArea = new JTextArea();

chatArea.setEditable(false);
```

```
        chatArea.setFont(new
Font("Arial", Font.PLAIN,
14));
        add(new
JScrollPane(chatArea),
BorderLayout.CENTER);

        // ==== Input Field ====
        inputField = new
JTextField();
        add(inputField,
BorderLayout.SOUTH);
```

```
        // ==== Action Listener
====

inputField.addActionListene
r(new ActionListener() {
            public void
actionPerformed(ActionEve
nt e) {
                String userInput =
inputField.getText();

chatArea.append("You: " +
userInput + "\n");
```

```
                String botResponse
= getResponse(userInput);

chatArea.append("Bot: " +
botResponse + "\n\n");

inputField.setText("");
            }
        });
    }

    // ==== NLP-like
Response System ====
```

```
    private String
getResponse(String input) {
        input =
input.toLowerCase().trim();

        // Match against
knowledge base
        for (String key :
knowledgeBase.keySet()) {
            if
(input.contains(key)) {
                return
knowledgeBase.get(key);
```

```
 -> {
            new
AIChatBotGUI().setVisible(tr
ue);
        });
    }
}
```

🚀 Features

Java-based chatbot.

Uses rule-based NLP logic (keyword matching).

GUI-based interaction with JTextArea & JTextField.

Can be extended to load FAQs from a file (training).

Can later be improved with OpenNLP / Stanford NLP for

real NLP.

👉 Do you want me to upgrade this chatbot so it can read FAQs from a text/CSV file (train itself dynamically) instead of hardcoding responses?