Project:

Simple Rule-Based Chatbot

**TIMELINE OF THE PROJECT:**

**DAY 1: 20, Aug,2024 (Tuesday)**

**Project Setup and Rule Definition**

**Strategy:**

* Set up your Python environment.
* Create a list of common queries and responses.
* Define rules/patterns for matching user inputs.
* Begin coding the basic chatbot structure.

**DAY 2: 21, Aug,2024 (Wednesday)**

**Implementing Core Functionality**

**Strategy:**

* Implement pattern matching using regex or simple string matching.
* Add functions to handle greetings, farewells, and common questions.
* Set up a loop for user interaction.
* Updated Code:
* The code provided above already covers these functionalities. Ensure that all common patterns are added to the responses dictionary.

**DAY 3: 22, Aug,2024 (Thursday)**

**Enhancing Responses and Handling Small Talk**

**Strategy:**

* Expand the list of responses to handle more varied inputs.
* Implement additional small talk responses.
* Add fallback responses for unrecognized inputs

**DAY 4: 23, Aug,2024 (Friday)**

**Final Testing and Documentation**

**Strategy:**

* Conduct thorough testing with various inputs.
* Optimize code for readability and efficiency.
* Write documentation explaining the chatbot’s functionality and usage.
* Prepare a presentation or demo.

**ROLES OF TEAM:**

Muhammad Ali Shafique(Leader):

**Query Management**:

* Handle and respond to queries effectively, ensuring accurate retrieval and processing of information within the system.

**Presentation and Display**:

* Focus on the clear and user-friendly presentation of information, making sure that outputs are easily understandable and visually accessible.

**Input Handling and Response Mechanisms**:

* Design robust input handling with fall-back mechanisms to manage unexpected or invalid inputs, providing appropriate responses or default actions.

**Pre-defined Rules and Logic**:

* Establish and enforce a set of pre-defined rules to guide the behavior of the system, ensuring consistent and predictable outcomes.

Adan Fatima:

**Scripting and Automation**:

* Develop modular Python scripts for task automation, ensuring efficiency and reducing manual work.

**Project Structure:**

* Design and maintain a clear project structure.

**Testing and Quality Assurance**:

* Implement and execute unit tests and integration tests ensuring code reliability and high-quality output.

**Function Implementation and Matching**:

* Develop and integrate functions to handle specific tasks, such as matching inputs against criteria or patterns, ensuring accurate processing.