DAY – 99

23 December 2025

Pseudo Code

Chatbot_Web_Application ()

```
BEGIN

# System Initialization
    Initialize Flask Server
    Initialize Frontend_Chat_Interface
    Initialize Session_Manager
    Initialize ML_Model (Gemma Model)

    Session_Timeout = 1 Hour
    Model_Loaded = FALSE
    Display Website_Homepage
    Display Floating_Chat_Button


# Session Handling
    IF User_Visits_Website THEN
        Create New_Session
        Assign Session_ID
    END IF


# User Interaction
    WHILE Application_Is_Running DO
        IF Chat_Button_Clicked THEN
            Open Chat_Popup
            Display Welcome_Message
```

```
      Display Language_Selection (English / Punjabi)
    END IF

    Read User_Message
    Read Selected_Language

    IF User_Message = NULL OR User_Message = "" THEN
      Display "Please enter a message"
      CONTINUE
    END IF
    Display User_Message_On_Chat_Screen
    Show Typing_Indicator


# Backend Processing
    IF Model_Loaded = FALSE THEN
      Load PDF_Files
      Load Text_Files
      Extract Text_From_PDF
      Clean And_Preprocess_Data
      Load ML_Model (Gemma)
      Model_Loaded = TRUE
    END IF

    Detect Input_Language
    Normalize User_Message
    Tokenize Normalized_Input


# Response Generation
    Generate_Response_Using_Gemma_Model (Processed_Input, Loaded_Data)
```

```
# Response Formatting
    IF Selected_Language = "Punjabi" THEN
      Translate_Response_To_Punjabi
    END IF



# Response to User
    Stop Typing_Indicator
    Display Bot_Response_With_Stream_Effect

# Session Expiry Check
    IF Session_Timeout_Reached OR User_Closes_Browser THEN
      End Session
      Clear Session_Data
    END IF
  END WHILE

END
```

4.3.2 Algorithm

Step 1: Start the application.

Step 2: Initialize the system components:

- Initialize Flask server
- Initialize frontend chat interface
- Initialize session manager
- Initialize knowledge base

Step 3: Set session timeout to **1 hour** and mark the knowledge base as **not loaded**.

Step 4: Display the website homepage and floating chat button.

Step 5: When a user visits the website, create a new session and assign a unique session ID.

Step 6: Continuously run the application while it is active.

Step 7: If the user clicks the chat button:

- Open the chat popup
- Display a welcome message
- Ask the user to select a language (English or Punjabi)

Step 8: Read the user message and selected language.

Step 9: If the user message is empty or null:

- Display "Please enter a message"
- Go back to Step 8

Step 10: Display the user message on the chat screen and show a typing indicator.

Step 11: If the knowledge base is not loaded:

- Load dataset file
- Load PDF file
- Extract text from PDF
- Clean and pre-process the extracted text
- Parse questions and answers
- Mark the knowledge base as loaded

Step 12: Detect the input language of the user message.

Step 13: Normalize the user input and perform tokenization.

Step 14: Apply rule-based matching using the tokens and knowledge base to find the best answer.

Step 15: If no answer is found:

- Search for relevant information from the PDF content

Step 16: If the answer is still not found:

- Set the response as *"Sorry, I do not have an answer for this query."*

Step 17: If the selected language is Punjabi:

- Translate the response into Punjabi

Step 18: Stop the typing indicator and display the chatbot response with a streaming effect.

Step 19: Check for session expiry or browser closure:

- End the session
- Clear session data if timeout is reached

Step 20: Repeat Steps 6 to 19 until the application stops.

Step 21: End the algorithm.