Step 1: Setting Up the Development Environment

1) *Install Python*

We need to ensure that Python 3.7 or above is installed on our machines. We can download it from python.org.

To verify the installation, we run:

python --version

pip –version

2) *Set Up a Virtual Environment*

To isolate our project dependencies, we will create and activate a virtual environment:

python -m venv venv

To Activate it:

venv\Scripts\activate

3) *Install Project Dependencies*

Using the requirements.txt file provided in the project, we install all necessary libraries:

pip install -r requirements.txt

4) *Install Visual Studio Code (VS Code)*

We will use VS Code as our code editor for this project.

Step 2: Understanding the Project Structure

Before we dive into editing or running the project, here is the folder and file structure:

Static Folder:

- Contains all front-end resources:
    - chatbot.css, signup.css, styling.css: These files manage the design and layout.
    - script.js, gradient.js: These files add interactivity to the pages.
    - user_data.json: Contains sample user input data.

Templates Folder:

- Holds the HTML files for the web pages:
    - index.html: Main home page.
    - chatbot.html: Chatbot interface.
    - signup.html: Signup page.

Instance Folder:

- Contains database files:
    - app.db: These store chatbot and user data.

Core Backend Files:

- app.py: The main Flask application that serves the project.
- database.py and signup.py: Handle database operations.
- training.py: Retrains the chatbot model.

Model and Data Files:

- intents.json: Defines chatbot intents and responses.
- model.h5, labels.pkl, texts.pkl: Pre-trained model and label files.

Step 3: Editing and Enhancing the Code

To ensure everything works, here are the main codes we need to execute:

1. **Backend Code**:
    - Execute app.py to start the Flask server.
2. **Model Training (if required)**:
    - Run training.py to retrain the chatbot model with new intents.
3. **Database Management (if needed)**:
    - Use database.py or signup.py to manage or reset the database.

Once we are done editing and enhancing, we proceed to execute the application.

Step 4: Running the Application

Follow these steps to execute the project:

1. **Activate the Virtual Environment**:
    - Open the terminal in the project folder and activate the virtual environment:

      venv\Scripts\activate

- **Run the Flask Application**:

- Start the backend server by executing app.py:

  bash

  Copy code

  python app.py

  - **Access the Application**:

- Open a web browser and go to:

  http://127.0.0.1:8000/

- Visit the following pages:
  - **Home Page**: index.html
  - **Chatbot Page**: chatbot.html
  - **Signup Page**: signup.html