

AI Chatbot MVP – Beginner's Guide

This document explains the AI Chatbot MVP project in simple terms, with diagrams and screenshot placeholders. It is designed for absolute beginners.

Table of Contents

1. Project Overview
2. How the Chatbot Works (with Diagram)
3. Project Structure & Files (with Diagram)
4. Backend (Server) Explanation
5. Frontend (User Interface) Explanation (with UI Diagram)
6. How Everything Connects (with Flowchart)
7. How to Run the Project
8. Screenshots (Replace Placeholders)

1. Project Overview

This project is a simple web-based chatbot powered by artificial intelligence (AI). It allows users to type questions or messages and get smart replies from an AI assistant. The project is designed to be easy to understand and run, even for beginners.

2. How the Chatbot Works

Step-by-step Flow:

- The user types a message in the chat window on a web page.
- The message is sent to a server (the backend) running on your computer.
- The server asks the AI (from OpenAI) for a reply.
- The server sends the AI's reply back to the web page.
- The reply appears in the chat window for the user to read.

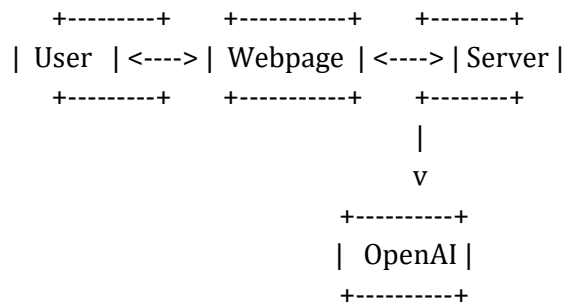


Diagram: Chatbot Flow

3. Project Structure & Files

Folder Structure:

```
SaturnIQ/  
├── backend/  
│   └── app.py  
├── frontend/  
│   └── index.html  
└── Documents/  
    └── AI_Chatbot_MVP_Explanation.docx
```

What Each Folder/File Is For:

- backend/: Contains the server code (Python)
 - - app.py: The main server script that talks to the AI and handles messages.
- frontend/: Contains the web page files
 - - index.html: The main web page you see in your browser.
- Documents/: Contains documentation and explanations
 - - AI_Chatbot_MVP_Explanation.docx: This explanation file.

4. Backend (Server) Explanation

The backend is a Python program that runs on your computer. It uses a tool called Flask to create a simple web server. It listens for messages from the web page (frontend). When it gets a message, it sends it to the AI and waits for a reply. Once the AI replies, the backend sends the answer back to the web page. The backend also handles errors, like if the AI is not available.

Main Parts of the Backend:

- Flask: Lets Python act as a web server.
- CORS: Allows the web page to talk to the backend, even though they run separately.
- OpenAI: Connects to the AI service to get smart replies.
- API Key: A secret code that lets the backend use the AI service.

What Happens When You Send a Message:

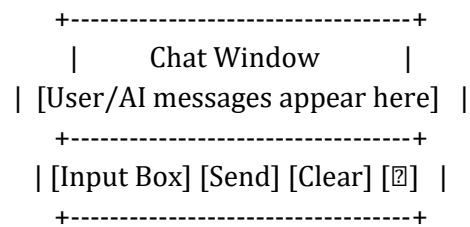
- The backend receives your message from the web page.
- It sends your message to the AI.
- It waits for the AI to reply.
- It sends the AI's reply back to the web page.
- If something goes wrong, it sends an error message instead.

5. Frontend (User Interface) Explanation

The frontend is a web page you open in your browser. It shows a chat window where you can type messages and see replies. It uses HTML for structure, CSS for styling, and JavaScript for making things interactive.

Main Parts of the Frontend:

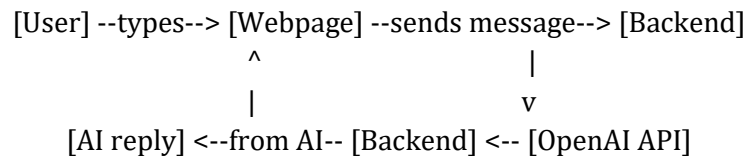
- Chat Window: Shows the conversation between you and the AI.
- Input Box: Where you type your message.
- Send Button: Click to send your message.
- Clear Chat Button: Clears the chat history.
- Theme Toggle: Switches between light and dark mode.
- Loading Spinner: Shows when waiting for the AI to reply.
- Copy Button: Lets you copy a message to your clipboard.
- Toast Notification: Shows small pop-up messages for errors or actions.



UI Layout Diagram

6. How Everything Connects

Flowchart:



The frontend (web page) and backend (server) talk to each other using the internet, but both run on your computer. The frontend sends messages to the backend using a special address (<http://localhost:5000/api/chat>). The backend talks to the AI and sends the reply back to the frontend. The frontend updates the chat window with the new message.

7. How to Run the Project

- Start the backend (server): Open a terminal and run the backend Python script.
- Start the frontend (web page): Open another terminal and run the command to start the web server for the frontend.
- Open your browser and go to <http://localhost:8000> to use the chatbot.

8. Screenshots (Replace Placeholders)

Tip: To take a screenshot, press PrtScn (Print Screen) or use the Snipping Tool on Windows. Paste (Ctrl+V) or insert the image into the document where indicated below.

a) Chat Window (Initial State)

[Insert screenshot here]

b) Sending a Message

[Insert screenshot here]

c) AI Reply

[Insert screenshot here]

d) Error State

[Insert screenshot here]

e) Dark Mode

[Insert screenshot here]

Callouts

Note: The diagrams above are for illustration. You can draw them in Word using shapes for a neater look.

Warning: Never share your API key publicly.

Tip: Replace all screenshot placeholders with your own screenshots for a complete guide.