CodSoft Internship Project Documentation

Project: Task 1- Rule-Based Chatbot with Professional Web UI
Intern: Vishal Baibhav Panda
Company: CodSoft

📘 Project Overview

This project is a rule-based chatbot built using Flask (Python backend) and a modern web frontend. It was developed as part of the CodSoft Internship to demonstrate natural language processing basics, rule-based responses, and professional chatbot UI design.

The chatbot responds to user inputs based on predefined rules (rules.json) and provides a professional interface similar to ChatGPT-style applications.

✨ Features

- • Rule-based responses using if-else / pattern matching
- • Predefined intents (greetings, name setting, project help, fallback, etc.)
- • Fuzzy matching for handling near matches
- • Context memory (remembers user's name in the session)
- • Dark modern theme (blue gradients, white text, sky blue accents)
- • Fixed chat panel size (scroll inside chat, panel doesn't grow infinitely)
- • Typing indicator (three bouncing dots while bot 'thinks')
- • Welcome center (ChatGPT-style greeting screen)
- • User login screen (name + optional email before entering chat)
- • Local storage chat history (persists until cleared)
- • Export Chat → Download conversation as .json
- • Clear History → Reset chat
- • Sign Out → Logout and return to login page

📂 Project Structure

```
codsoft_chatbot/
├─ .gitignore
├─ README.md
├─ backend/
│  ├─ requirements.txt
│  ├─ app.py
```

```
│  ├─ rules.json
│  ├─ chat_transcript.log
│  ├─ templates/
│  │  ├─ index.html
│  │  └─ login.html
│  └─ static/
│     ├─ css/styles.css
│     └─ js/
│        ├─ app.js
│        └─ login.js
```

⚙️ Installation & Setup

- ➡️ Clone the repository: git clone <repo-url>
- ➡️ Navigate to backend/: cd codsoft_chatbot/backend
- ➡️ Create virtual environment & install dependencies:
  python3 -m venv venv
  source venv/bin/activate
  pip install -r requirements.txt
- ➡️ Run the Flask server: python app.py
- ➡️ Open in browser: http://127.0.0.1:5000/login

👩‍💻 Usage Flow

1. Login Page → Enter your name (and optional email).
2. Chat Window → Start conversation by typing 'hi', 'my name is ...', or any predefined query.
3. Bot Responses → Bot matches against rules and responds.
4. Chat Controls:
   - Send → send message
   - Clear History → reset chat
   - Download Chat → export as JSON
   - Sign Out → logout

📝 Rules Configuration Example

```
{
  "intents": [
    {
      "name": "greeting",
      "patterns": ["hi", "hello", "hey"],
      "responses": ["Hello! What would you like to do?", "Hey there 👋 How can I help you today?"]
```

```
  },
  {
    "name": "set_name",
    "patterns": ["my name is (.*)", "i am (.*)"],
    "responses": ["Nice to meet you, {name}!", "Got it — I'll remember that you're {name}."]
  },
  {
    "name": "fallback",
    "patterns": [],
    "responses": ["I didn't quite catch that. Could you rephrase?", "Sorry, I don't know that
yet."]
  }
 ]
}
```

📊 Technologies Used

- ✔️ Python 3
- ✔️ Flask
- ✔️ HTML, CSS, JavaScript
- ✔️ LocalStorage (browser-based storage)

📌 Example Demo

You: hi
Bot: Hey there 👋 How can I help you today?

You: my name is Vishal
Bot: Got it — I'll remember that you're Vishal.

You: what is my name?
Bot: Yes — you're Vishal.

✅ Final Notes

This chatbot is:
- Rule-based (not AI/ML powered, as per internship task)
- Professional UI (dark theme, responsive, modern look)
- Easy to extend (just add new intents in rules.json)

📌Confidential