CUREMATE - AI POWERED CHATBOT REPORT SUBMITTED TO

K. R. MANGALAM UNIVERSITY

BACHELOR OF
COMPUTER APPLICATIONS
(AI & DATA SCIENCE)

SUBMITTED BY : G231
YASH RAJ JHA (2401201143)
DIPANSHU OHLYAN(2401201062)
PAKHI PAL (2401201088)
KHYATI KALRA (2401201152)

SCHOOL OF ENGINEERING & TECHNOLOGY

K. R. MANGALAM UNIVERSITY

SOHNA , HARYANA 122103 , INDIA

INTRODUCTION:-

The purpose of the application **“CureMate”** is to provide users with an easy-to-use platform for managing healthcare services, appointments, and medical information. Designed to bridge the gap between patients and healthcare providers, CureMate acts as a centralized system where users can schedule appointments, view medical records, and communicate with doctors. It offers various features such as doctor listing, user registration and login, appointment booking, and a contact form for queries and feedback.

The frontend of the website was developed using HTML, CSS, and JavaScript. HTML was used to structure the content, CSS added styling to enhance visual appeal, and JavaScript brought interactivity to the platform. Responsive design was also implemented, ensuring that the website adapts to various screen sizes, including smartphones and tablets. JavaScript was further utilized to dynamically update appointment slots, show available doctors, and handle login popups and form validations.

The backend of the application is powered by **Flask**, a lightweight Python web framework ideal for small to medium-sized applications. API endpoints were created to manage appointments, handle user authentication (in a basic, dummy format), submit contact forms, and fetch doctor information. Data was temporarily stored in Python data structures like lists to simulate database behavior. CORS (Cross-Origin Resource Sharing) was enabled to allow seamless interaction between the backend and frontend hosted separately.

Key features include:

- A registration and login system with basic validation.
- An appointment booking form with real-time input handling.
- Dynamic display of doctors and available time slots.
- A “Contact Us” form for patient support and inquiries.
- Mobile-responsive layout for accessible healthcare management

on-the-go.

Currently, CureMate uses temporary in-memory storage and basic login functionality. Future improvements could involve integrating a real database such as PostgreSQL or MongoDB, implementing secure and encrypted user authentication, supporting digital prescriptions and medical record uploads, and providing email or SMS notifications for appointments.

In conclusion, CureMate serves as a foundational web application that efficiently demonstrates core features of a healthcare management system, setting the stage for future expansion into a comprehensive digital healthcare platform.

Objectives:-

The main goal of “CureMate” is to create an efficient, centralized, and user-friendly healthcare management system that bridges the gap between patients and healthcare providers. Traditional healthcare access often involves long queues, paperwork, fragmented records, and inconsistent communication. CureMate aims to digitize this process, offering a streamlined platform that makes appointment booking, medical consultation, and health service access simple and accessible to all users.

The platform empowers patients to view available doctors, their specializations, and available time slots. They can then register, log in, and book appointments with ease—all within a few clicks. Doctors or healthcare providers, in turn, can manage incoming appointment requests and maintain better communication with patients through built-in forms and feedback tools.

A core objective of CureMate is to enhance communication between patients and medical staff. The integrated “Contact Us” feature allows users to send questions, suggestions, or concerns directly to the team—promoting better responsiveness, care quality, and user satisfaction.

Another key aim is transparency and trust-building. The system introduces users to available doctors and healthcare representatives with brief profiles and availability information. This helps users make informed decisions and builds credibility in the healthcare process.

CureMate is also designed with scalability and automation in mind. The current version supports basic login functionality and dynamic content updates. In the future, the platform can be expanded with database integration (e.g., MySQL or MongoDB), secure authentication systems, electronic medical record (EMR) support, appointment reminders, and email/SMS notifications.

In essence, the primary objective is to simplify appointment scheduling, enhance communication, boost accessibility to healthcare, and deliver a modern digital healthcare experience for both patients and providers.

- Patients can view doctors and book appointments.
- Doctors can manage their schedules and availability.
- The administration can monitor appointment activity.
- Contact and profile information is accessible and transparent.

Technology Used:-

Technology	Purpose
HTML	Structuring Web Pages
CSS	Styling Web Pages
JavaScript	Frontend Interactivity
Flask (Python)	Backend Server Framework
JSON	Data Exchange Format

System Requirements:-

- Python 3.8+
- Flask (pip install flask flask-cors)
- Web Browser (Chrome, Firefox, MS Edge)
- Text Editor (Visual Studio Code or similar)

System Components:-

Frontend

- **HTML (index.html, doctors.html, appointment.html)**
  - o Defines the layout of core pages like the homepage, doctor listings, appointment booking, login/registration forms, and contact page.
  - o Incorporates sections for user input, profiles, and navigation.
- **CSS (style.css)**
  - o Applies consistent and responsive styling across pages.
  - o Designs form layouts, buttons, doctor cards, contact forms, and adaptive mobile views.
- **JavaScript (script.js)**
  - o Handles UI interactivity and dynamic content rendering:

♣ Show/hide modals for login and appointment forms.

♣ Populate doctor listings and appointment slots dynamically.

♣ Validate form inputs and manage user interaction effects.

♣ Implement mobile navigation and smooth scrolling.

Backend

- **Python Flask Application (app.py)**
  - o Serves static files including HTML, CSS, and JS.
  - o Provides REST API endpoints to:
    - ♣ Fetch available doctors and schedules.
    - ♣ Create and store appointment requests.
    - ♣ Handle user login/registration (dummy implementation).
    - ♣ Receive and store contact form queries.
  - o Uses in-memory data structures (Python lists) to simulate a backend database.

Features Implemented:-

Feature	Description
Login System	Users can attempt to log in (dummy implementation without secure authentication).
Doctor Listing	Users can view available doctors along with their specialization and timings.
Appointment Booking	Patients can book appointments through a modal form.
Contact Us Form	Allows users to send inquiries or feedback to the CureMate team.
Mobile Responsive Design	Layout adjusts to various screen sizes, including smartphones and tablets.
Dynamic Content	Doctor profiles and appointments are dynamically rendered using JavaScript.

Flow of the Application:-

1. The homepage loads with a clean interface and navigation links.
2. User can:
   - Browse available doctors and their schedules.
   - Book appointments using a simple form.
   - Log in or register via modal popup (dummy logic).
   - Contact support through the contact form.
3. Backend APIs handle form submissions for appointments, contact messages, and login attempts.

- The application is designed to run locally using Flask.
- To run:

1. Install dependencies (Flask, Flask-CORS).
2. Start the server by running python app.py.
3. Visit http://localhost:8080/ to access the CureMate portal.

Working Process :-

1. User visits the homepage and gets an overview of CureMate's features.
2. Clicking "Login" shows a popup to simulate user login.
3. Clicking "Book Appointment" displays a form to select a doctor and time slot.
4. The doctor list is displayed dynamically with relevant details.
5. The user fills in the form and submits an appointment request.
6. Visitors can send inquiries using the contact form.
7. The Flask backend processes and stores the submitted data using in-memory lists.

Limitations :-

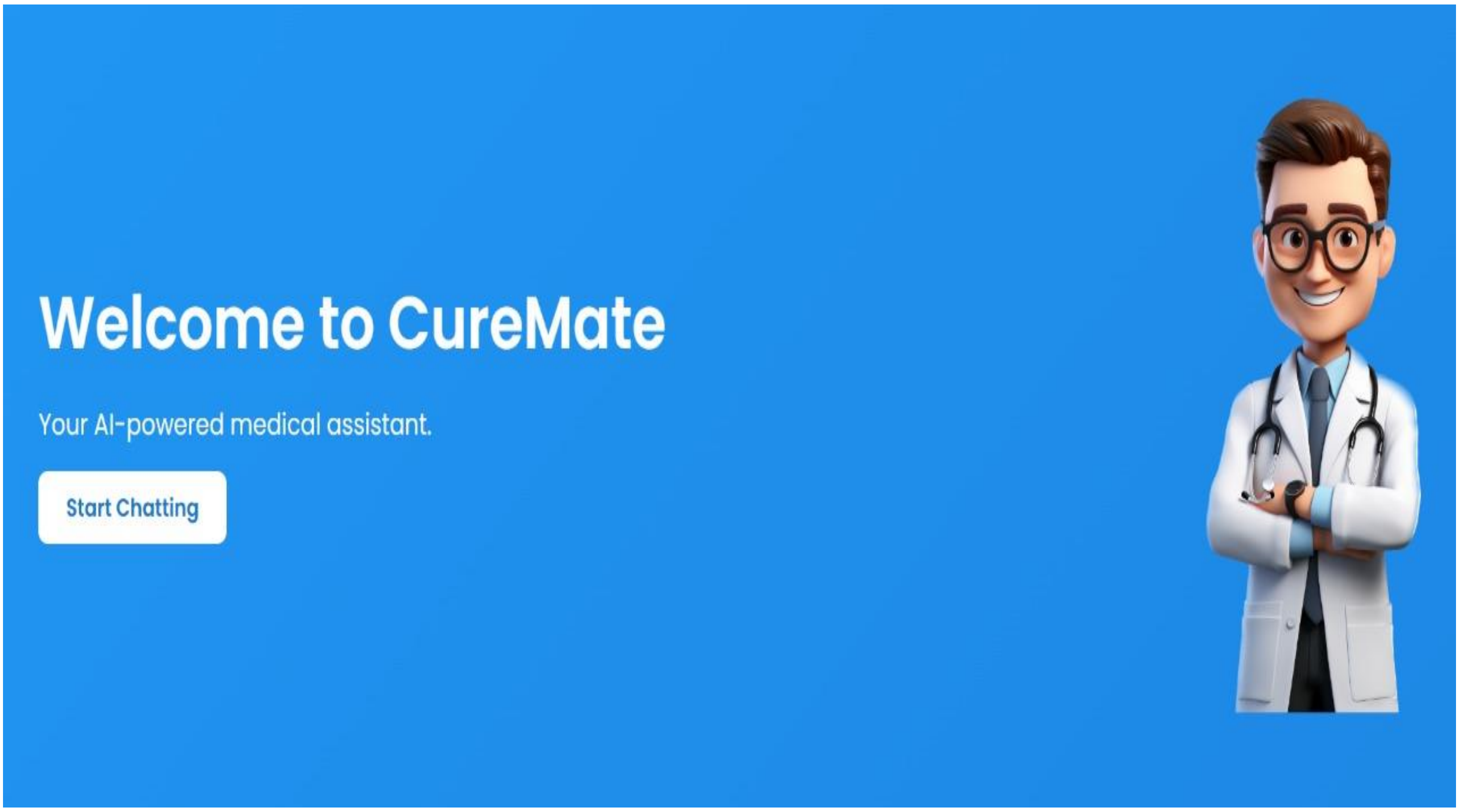
- No persistent database; uses temporary in-memory data.
- No real user authentication or session management.
- No patient history or medical record storage.
- Image uploads for doctor profiles not supported.
- No admin dashboard for managing appointments.

Future Improvements :-

- Integrate a real database like SQLite or PostgreSQL.
- Implement secure login with password hashing and session control.
- Add roles (Admin, Doctor, Patient) with access control.
- Enable file uploads (e.g., profile pictures, prescriptions).

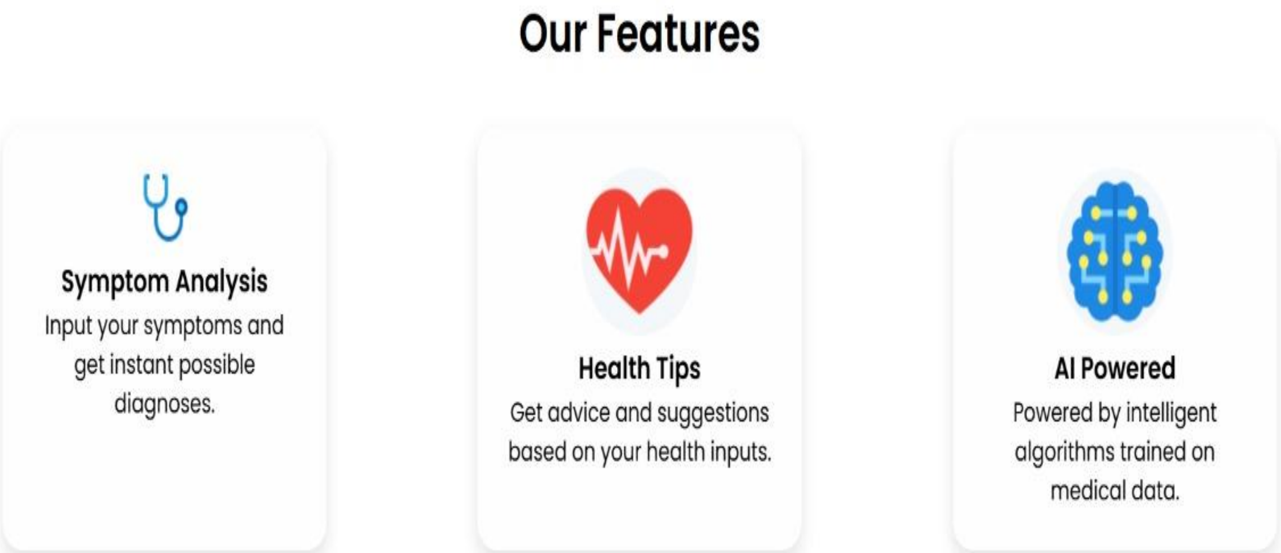
- Track appointment history and allow cancellations/rescheduling.
- Send email/SMS notifications for appointment reminders and confirmations.

Screenshots:-



Welcome to CureMate

Your AI-powered medical assistant.



Our Features

Symptom Analysis

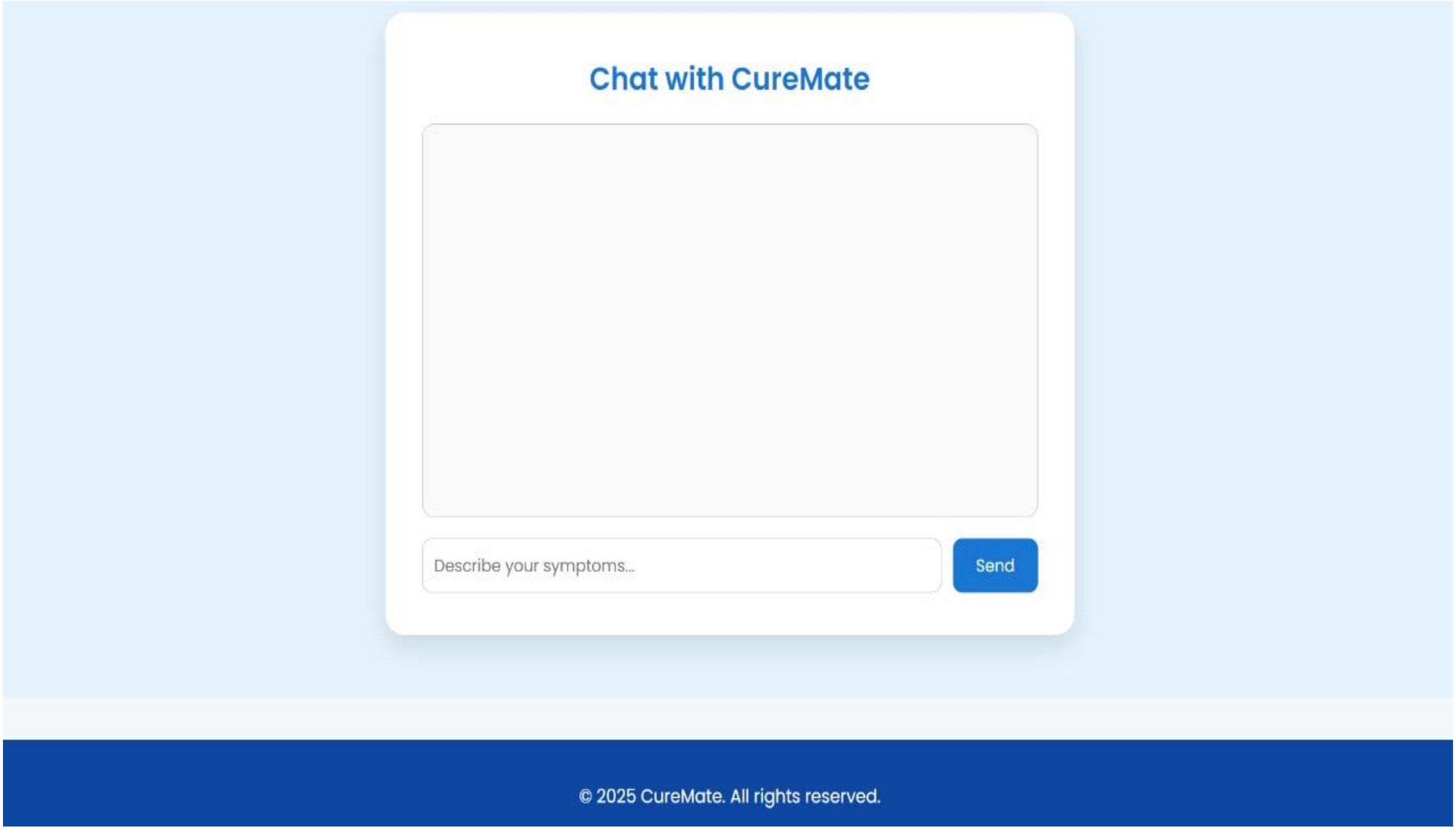
Input your symptoms and get instant possible diagnoses.

Health Tips

Get advice and suggestions based on your health inputs.

AI Powered

Powered by intelligent algorithms trained on medical data.



Chat with CureMate

© 2025 CureMate. All rights reserved.

Conclusion:-

The **"CureMate – AI Medical Chatbot"** project presents a functional prototype that leverages digital technologies to make preliminary healthcare guidance more accessible and interactive. By integrating a responsive frontend interface with a scalable Flask backend, the project delivers key features such as symptom-based diagnosis suggestions, voice input support, multilingual capabilities, and a clean user interface for natural user interaction.

The platform significantly enhances the user experience by providing an easy-to-use, informative tool that assists individuals in understanding their health concerns without the need for immediate medical intervention. Although it currently uses static symptom matching and simple in-browser logic, the project is well-positioned for future development, including backend database integration, AI/ML-enhanced diagnosis, real-time chatbot connectivity, and integration with telemedicine platforms.

This web project not only digitizes the process of preliminary health checkups but also promotes awareness and early detection through an engaging AI interface. With continued development, CureMate holds the potential to evolve into a full-scale virtual healthcare assistant that could be integrated into actual clinical support systems. It fulfills its goal of using technology to bridge the gap between healthcare and accessibility.