DiaBot: A Flask-Based Diabetes Chat Assistant with Admin Panel and Chat History

AGASTYA SREEDHAR

Grade 12,Downingtown STEM Academy, US

Email:agastyasreedharz@gmail.com, umasindhucs@gmail.com

ABSTRACT

DiaBot is a chatbot system created using Flask which is a Python web framework, SQLite for database, and Pandas for data processing.It is designed exclusively for diabetes patients.Chatbot is a well designed healthcare assistant for patients suffering from diabetes and it is also applicable to common users for providing responses for the questions raised by users.It provides quick answers to all the queries regarding diabetes without any delay.The system implements user authentication, personalized chat history, an admin panel for managing knowledge base (Question/Answers), and bulk Q/A import via Excel/CSV. The user interface is implemented using a glassmorphism effect ,thereby providing modern usability. Unlike static FAQ (Frequently asked question) pages, Diabot provides an interactive experience with real-time response retrieval, history tracking, and simple analytics for administrators.

Key words: AI, ML, Diabetic, Diabot, Python, SQLite

1. INTRODUCTION

Due to the rapid advancements in AI [1] and ML [2] technology it has paved way to create answers for complicated medical issues in a many of the domains, including healthcare[3][4]. By giving users with proper responses and by efficiently handling their questions, these chatbots can provide answers just like healthcare experts. The main thing needed for the system to function properly is power and an internet connection. [5] A innovative healthcare system is created which is responsible for integrating chatbot assistance for disease prediction with the help of machine learning. In order to evaluate symptoms and provide patients with therapy alternatives, the approach uses complex algorithm. This project study shows chat bots can provide patients with easy access to reliable medical information by reducing the workload of medical personnel. [6]

Diabot provides these advantages as follows :

- Patients can directly ask questions in natural language.
- Responses are provided based upon the stored knowledge in a database.
- Administrators such as doctors, nurses, healthcare managers can make changes like update, add, or import knowledge at any time.
- The chatbot is capable of tracking chat history for personalization and analytics.

The project uses Flask as the backend as it is lightweight, scalable, and easy to integrate with a relational database. SQLite is used as the database since it is file-based and it needs no separate server.This makes DiaBot a low-cost, flexible, and extendable chatbot solution for small clinics, student projects, and academic demonstrations.

2. SYSTEM METHODOLOGY

2.1 Architecture

The overall architecture of DiaBot is organized into three distinct layers to maintain modularity and scalability. The **frontend (User interface)** is well developed using HTML templates rendered through Flask and also it features a glassmorphism design style that gives the chatbot a clear, refined and clear layout. The **backend (server logic)** is modern, professional, and engaging appearance by Flask, where different routes handle essential operations such as user login, registration, chat submissions, and admin functionalities. Atlast the **database layer (SQLite)** manages persistent storage by maintaining three categories of data: user accounts, the knowledge base (Q/A pairs), and chat history. This segmented structure allows the chatbot's conversational logic to be improved independently, without requiring modifications to authentication mechanisms or database management, thereby providing a clean separation of concerns and easier future upgrades.

2.2 Database Design

The database consists of three main tables:

1. **Users**
    - User table includes its fields as id, username, and password.
    - The main purpose is to store login credentials. Passwords are stored as **values** (in hashed format) using generate_password_hash thereby ensuring strong security.
    - A **default admin account** is created automatically if not present ,where we include username and password in the code itself
2. **Q/A (Knowledge Base)**
    - Knowledge base includes its fields as id, question, and answer.
    - Question/Answer (Knowledge Base) represents the chatbot's stored responses.
    - Administrator of diabot can **add, edit, delete, or import** Q/A pairs in large numbers.
3. **Chat History**
    - Chat History includes its fields as **id, username, question, answer, timestamp**.
    - Makes use of logs to record as user's interaction with the chatbot.
    - Allows users to review their history and admins to analyze commonly asked questions.

2.3 Chatbot Logic

Here the user submits questions in the chat text box.The entered question will be compared to the question stored in question answer database table.This done by using the function difflib.get_close_matches which performs fuzzy matching.The response will be provided even if the entered question partially matches with answers stored previously.If it doesn't find a match ,it simply returns the default message "Sorry,I can't answer that".

This makes DiaBot **flexible** in handling user input while keeping the system lightweight (unlike heavy NLP models).

3. IMPLEMENTATION

3.1 Authentication System

New users should register with a unique username and password.The entered password will be hashed before storing and it will not be stored as plain text. When the registered user logins, the entered password will be compared with the stored hashed password. Flask's session object is responsible for the users to be logged in until they log out.This ensures a well secured authentication thereby preventing unauthorized access.

3.2 Chat Functionality

Logged-in users can enter their question in a text box. The chatbot finds the closest answer and displays it immediately. Each interaction is recorded in chat history table along with a timestamp.

Here it provides a personalized experience, where every user has their own history and answers.

3.3 Admin Features

The **admin panel** is only visible to the admin account. It includes:

1. **Add Q/A:** A form to manually add new Q/A pairs.
2. **Edit/Delete Q/A:** Each entry can be updated or removed.
3. **Bulk Import (Excel/CSV):**
    - Admins can upload a spreadsheet containing multiple Q/A pairs.

- The system automatically detects columns like *Question, Answer, Q, Ans,* etc.
- Invalid rows (blank or duplicates) are skipped.
- A summary is shown: *"X added, Y duplicates skipped, Z blank rows skipped."*

4. **Top 5 Questions Analytics:**
   - Displays the most frequently asked questions based on chat history.
   - Helps admins improve the knowledge base by identifying popular queries.

This ensures the chatbot is **dynamic and easily maintainable**.

3.4 User Interface (UI/UX)

The frontend of the diabot application provides a well designed login form.The interface is well structured with semi-transparent background for different sections which includes login, chat, and the admin panel, which ensures and professional organized layout. It is fully responsive, allowing uninterrepted access across both desktop and mobile devices. In addition, there is a clear distinction between user areas and administrative tools, which enhances usability and prevents interface clutter.

4. FEATURE AND RESULTS

The developed system Diabot thereby provides a complete and well designed chatbot platform for managing healthcare frequently asked questions. The important outcome of this application is secure user authentication, since login and registration are protected using password hashing to ensure data privacy. The platform further includes an interactive chatbot that employs fuzzy string matching to handle partial matching in user queries and provide accurate responses which enhances personalization,and the system maintains a chat history.It enables users to move through their previous interactions at any time.

In administration side, the solution offers a robust admin panel that supports all the operations such as adding, editing, and deleting question answer pairs. In addition to that this system facilitates bulk knowledge import via Excel or CSV files, making it productive and efficient for large healthcare datasets. Beyond knowledge curation, the platform also provides statistical analysis features, such as identifying and displaying the most frequently asked questions, thus offering valuable insights for administrator

When considering with user interface, the application delivers a unique, glass-styled design with a seamless navigation, ensuring both modern look and usability. Hence,these functionalities establish DiaBot as a reliable, scalable, and user-friendly healthcare chatbot solution capable of supporting patients, caretakers, and healthcare providers with quick access to frequently asked information for diabetes.

5. DISCUSSION

The system is designed to be platform independent and adaptable, such that it can be capable of running on any machine that supports Python. The overall setup process will be simple and efficient, and it does not need the installation or configuration of an external database server.

In addition to that, it is capable of supporting to dynamic updates and it allows to include modifications to be applied without the need to restart the server.This feature ensures continuous availability. To improve and enhance security, the system has a robust authentication mechanisms with password hashing which acts as a effective mechanism to safeguard user credentials against unauthorized access.

6.FUTURE ENHANCEMENTS

- Replacing difflib matching with **NLP models** (e.g., spaCy, transformers).
- Adding **role-based access** (multiple admins, doctors).
- Implementing **email/SMS notifications** for critical queries.
- Integrating **visual charts** for admin analytics.
- Deploying on a **cloud server** with PostgreSQL for scalability.

7.CONCLUSION

This application, DiaBot demonstrates how a **Flask-based chatbot** can be developed with secure authentication, knowledge management, and analytics features. It is **scalable, flexible, and customizable and well** suitable for healthcare Frequently Asked Questions or other domains like education and customer support.

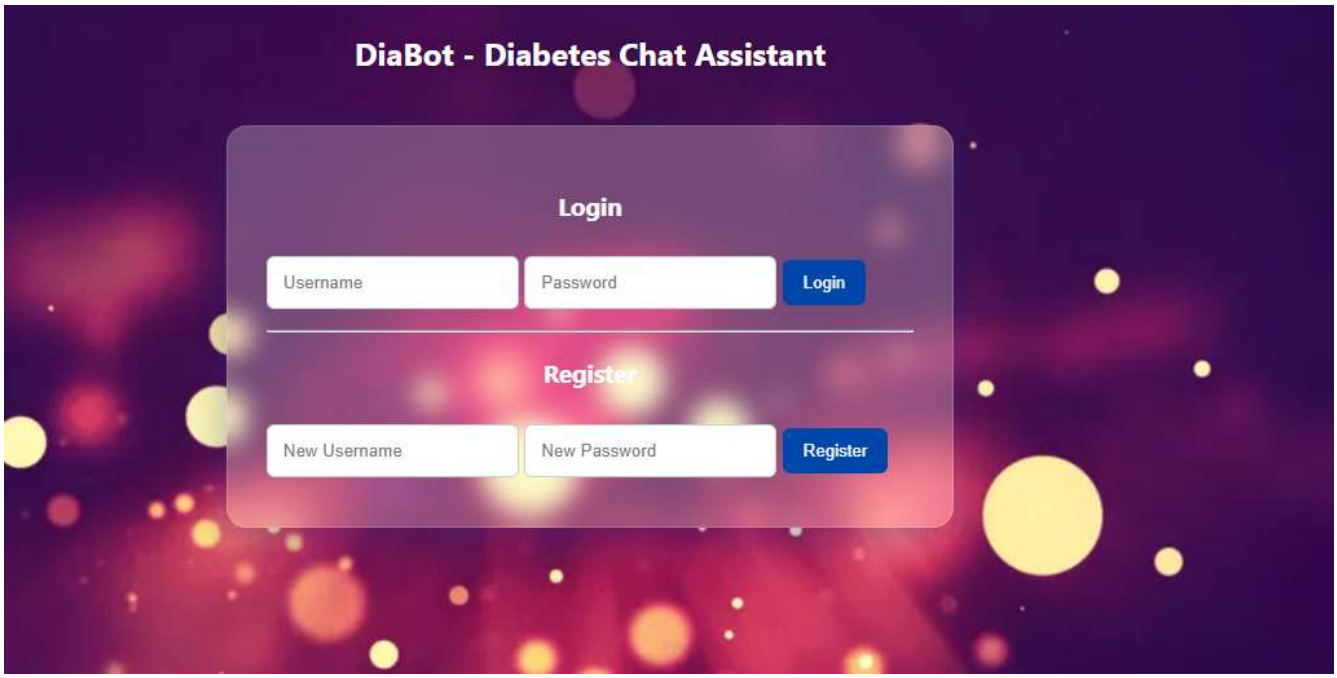
The project successfully combines **database-driven responses**, **user personalization**, and **admin controls** into a single application. Future work include enhancements in NLP and analytics, and making it evolve into a more intelligent and professional healthcare assistant.

8.ACKNOWLEDGEMENT

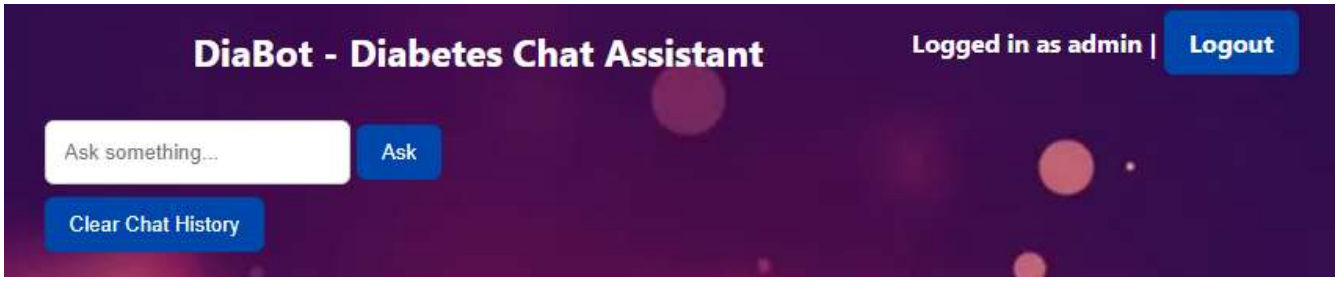
I would like to express my sincere gratitude to my guide **Mrs T.UMAPRIYA M.E (CSE),** Faculty (CS), **POLYGLOT, Jaipur, Rajasthan, India** for her valuable guidance throughout the successful completion of this project.

SCREENSHOTS

LOGIN



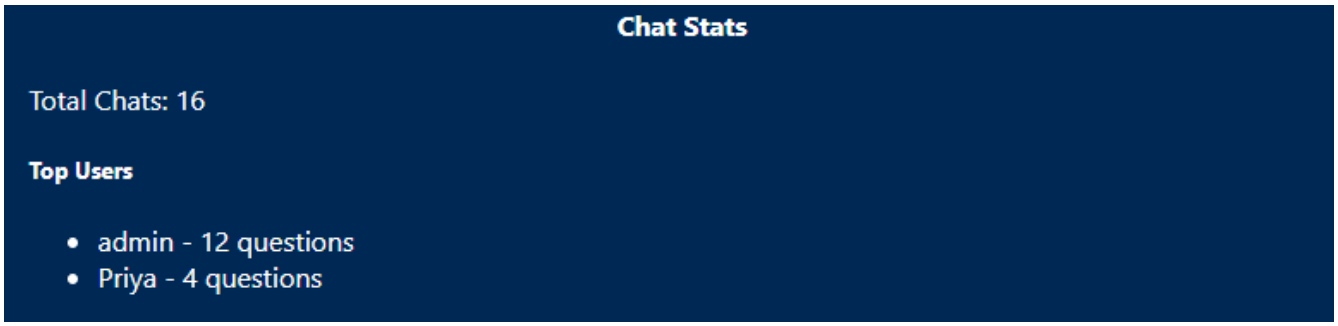
ADMIN PAGE



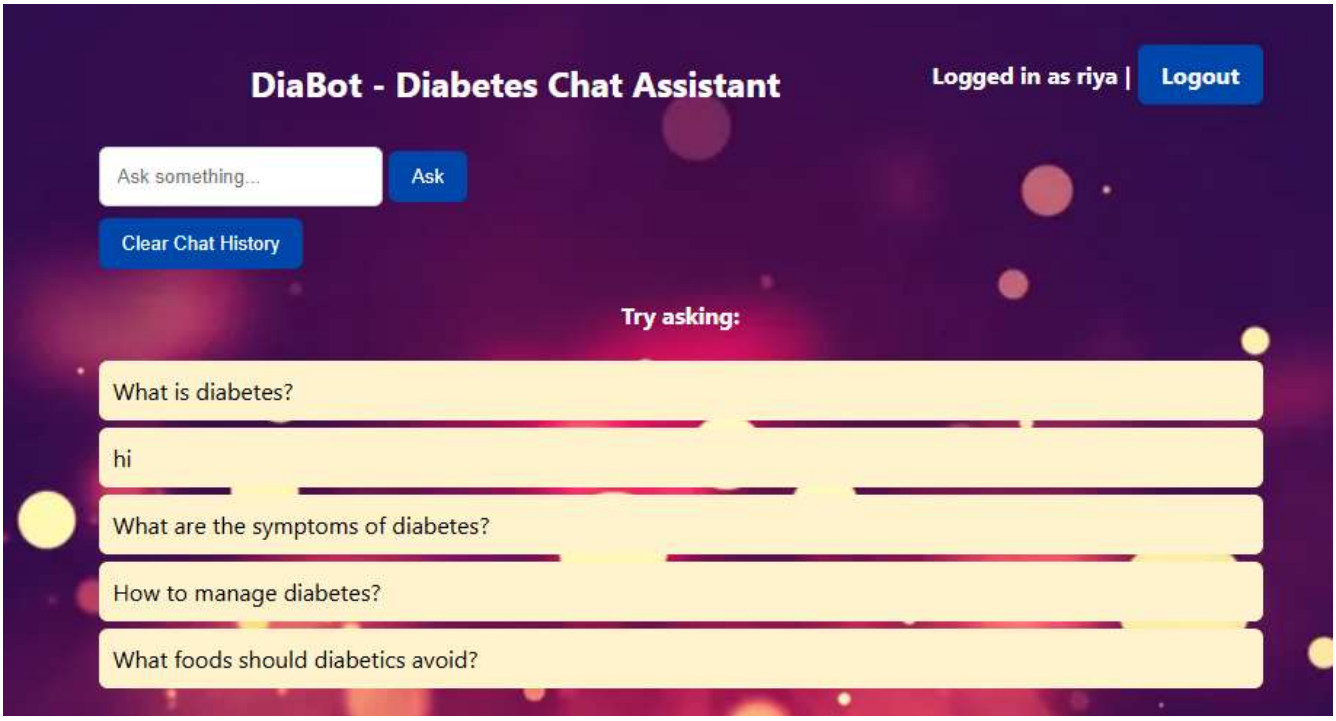
CHAT HISTORY WITH TIME STAMP

You: How to manage diabetes?
DiaBot: Diabetes can be managed with medication, healthy diet, regular exercise, and monitoring blood sugar.
2025-08-12 08:07:07

CHAT STATUS



USER LOGIN



REFERENCES :

[1] Salman Muneer, Taher M. Ghazal, Tahir Alyas, Explainable AI-Driven Chatbot System for Heart Disease Prediction Using Machine Learning, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 15, No. 12, 2024.

[2] Sahithya B, Prasad G, Sahithi B, Devarlla AC, Yashavanth TR. Empowering Healthcare with AI: Advancements in Medical Image Analysis, Electronic Health Records Analysis, and AI-Driven Chatbots. In2024 3rd International Conference for Innovation in Technology (INOCON) 2024 Mar 1 (pp. 1-7). IEEE.

[3] Bagheri M, Bagheritaba M, Alizadeh S, Parizi MS, Matoufinia P, Luo Y. AI-Driven Decision-Making in Healthcare Information Systems: A Comprehensive Review.

[4] Chakraborty C, Bhattacharya M, Pal S, Lee SS. From machine learning to deep learning: An advances of the recent data-driven paradigm shift in medicine and healthcare. Current Research in Biotechnology. 2023 Nov 22:100164.

[5] Biradar, S., Shastri, S.: Medical chatbot: Ai based infectious disease predic- tion model. Journal of Scientific Research and Technology (2024). Available at jsrtjournal.com

[6] Caruccio, L., Cirillo, S., Polese, G., Solimando, G.: Can chatgpt provide intelligent diagnoses? a comparative study between predictive models and chatgpt to define a new medical diagnostic bot. Expert Systems with Applications (2024). Available at sciencedirect.com