

Health Assist AI

Bhagyesh Manohar Neware¹ and Prof. Ujwala Khartad²

Student, Department of CSE(Data Science)¹

Professor, Department of CSE(Data Science)²

Tulsiramji Gaikwad Patil College of Engineering and Technology, Nagpur, Maharashtra, India

bhagyeshneware0123@gmail.com

Abstract: *This project is a personal AI health assistant built with Flask and MySQL. Users can log in to track daily health with tools for BMI, calories, and sleep. Its main features include AI-powered diagnostics for skin disease using a TensorFlow model and mental health prediction. The app also has an intelligent chatbot using the Open Router API and a premium, monetized diabetes risk checker powered by Razor pay.*

Keywords: *AI health assistant*

I. INTRODUCTION

This project is "Health Assist AI", a personal health assistant web app built with Python and Flask. It allows users to log in and track their wellness using tools like a BMI calculator and calorie tracker. Its key features are AI-driven diagnostics, including a TensorFlow model to identify skin diseases from images and a model to predict mental health status. It also features an AI chatbot and a premium, monetized diabetes risk checker.

II. METHODOLOGY

This project's methodology involves building a Flask web application that connects to a MySQL database to store user and health data. It integrates pre-trained machine learning models, using TensorFlow for image-based skin disease detection and Scikit-learn for form-based mental health prediction. The app also uses external APIs, connecting to Open Router for an AI chatbot and Razor pay to process payments for premium features. The user interface is created with HTML and Bootstrap templates.

Steps:

1. Foundation: Set up a Flask web application and connect it to a MySQL database.
2. User System: Created user registration and login pages, using Flask sessions to keep users logged in.
3. Load AI Models: Loaded a pre-trained TensorFlow model for skin disease detection and a Scikit-learn model for mental health.
4. Build Core Features: Developed routes for the AI diagnostics (image upload for skin, form for mental health) and health trackers (like BMI and calorie logging).
5. Integrate APIs: Added an AI chatbot using the Open Router API and set up the Razor pay API to handle payments for a premium diabetes-checking feature.

III. RESULTS AND DISCUSSION

The project successfully built "Health Assist AI," a working Flask web app. It correctly integrates two AI models for skin disease and mental health. The app also features a live chatbot using the Open Router API and a complete payment system for premium features using Razor pay. All user health data (calories, chat history) is successfully saved to a MySQL database specific to each user. The project's main strength is its excellent integration of different technologies: a Flask backend, AI models, and external APIs for chat and payments. It also has a smart business model by charging for premium features.



A. Figures

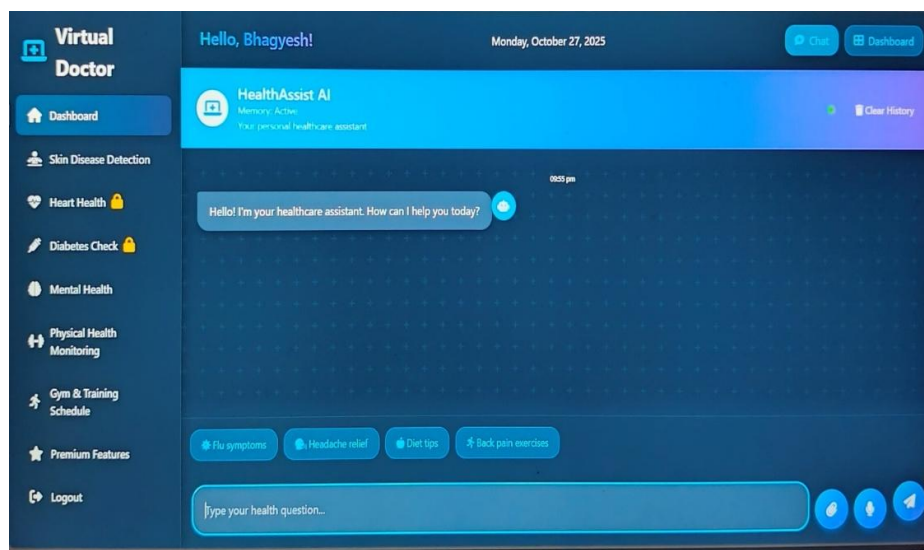


Fig. 1. Disease Analysis AI Chatbot

This is a smart chatbot powered by the Open Router API (gpt-4o-mini). It's not a simple, pre-programmed bot. It saves your entire conversation to the database, so it remembers what you last talked about, providing a continuous and personalized chat experience.

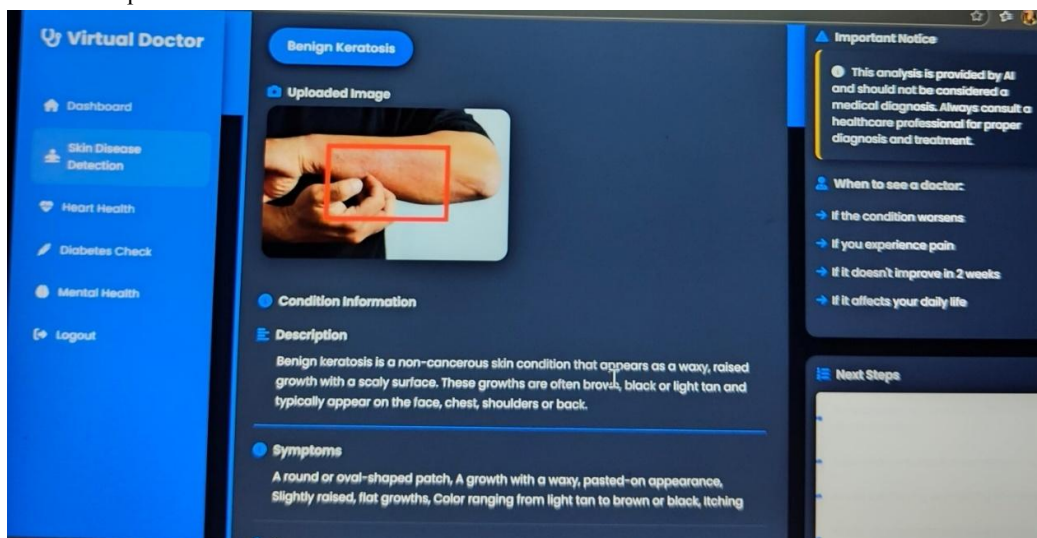


Fig. 2. Skin Disease Analysis

You upload an image of your skin. The app uses a TensorFlow AI model (modelnew.h5) to guess what it is from 7 categories, like "Melanoma" or "Benign Keratosis". It then shows you a description of the predicted condition.



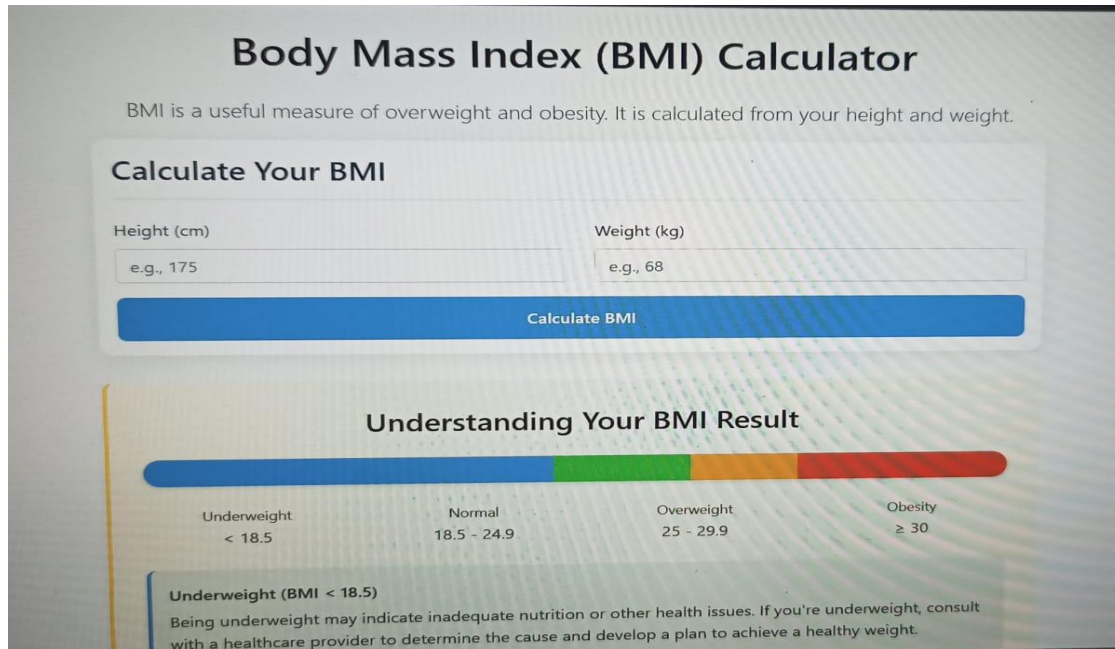


FIG 3. BMI CALCULATOR

IV. CONCLUSION

This project successfully created "Health Assist AI," a comprehensive and functional Flask application. It successfully integrates multiple AI models for skin and mental health diagnostics, a live chatbot using an external API, and a complete Razor pay payment system for premium features. While the project is a strong technical success, it requires a critical security update to hash user passwords, which are currently stored in plain text, before any real-world deployment.

V. ACKNOWLEDGMENT

I would like to express my gratitude for the opportunity to develop "Health Assist AI." This project was a significant learning experience, blending web development using Flask with the power of AI and machine learning for healthcare. Bringing together features like the skin disease analyzer, mental health predictor, and the AI chatbot has been both challenging and rewarding.

REFERENCES

- [1] Python: Official Documentation: The primary reference for the Python programming language, which forms the backend of your application. <https://docs.python.org/3/>
- [2] Flask: Official Documentation: The reference for the Flask web framework, used to build the server, handle routing, and manage requests. <https://flask.palletsprojects.com/>
- [3] MySQL: Official Documentation: The reference for the MySQL database, used for all data storage in the project. <https://dev.mysql.com/doc/>
- [4] TensorFlow: Official Documentation: The reference for the deep learning framework used to load and run the skin disease detection model (.h5). https://www.tensorflow.org/api_docs

