DAYANANDA SAGAR UNIVERSITY

Devarakaggalahalli, Harohalli
Kanakapura Road, Ramanagara - 562112, Karnataka, India

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

MEDSCORE ASSIST

Batch: 50

By

Adarsh Jaiswal- ENG21CS0010
Ananya Vithal Yergolkar- ENG21CS0041
Ankita S Hegde- ENG21CS0045
Archana B Biradar- ENG21CS0051

Under the supervision of
Prof. Gaurav Kumar
Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY

(2024-2025)

DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Devarakaggalahalli, Harohalli, Kanakapura Road, Ramanagara - 562112
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled **"MEDSCORE ASSIST"** is carried out by **Adarsh Jaiswal (ENG21CS0010), Ananya Vithal Yergolkar (ENG21CS0041), Ankita S Hegde (ENG21CS0045), Archana B Biradar (ENG21CS0051),** bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2024-2025**.

Prof. Gaurav Kumar	**Dr. Girisha G S**	**Dr. Udaya Kumar Reddy K R**
Assistant Professor Dept. of CS&E, School of Engineering Dayananda Sagar University	Chairman CSE School of Engineering Dayananda Sagar University	Dean School of Engineering Dayananda Sagar University
Date:	Date:	Date:

Name of the Examiner	**Signature of Examiner**
1.	
2.	

DECLARATION

We, **Adarsh Jaiswal (ENG21CS0010), Ananya Vithal Yergolkar (ENG21CS0041), Ankita S Hegde (ENG21CS0045), Archana B Biradar (ENG21CS0051),** are students of eighth semester B. Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled **"Medscore Assist"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2024-2025**.

Student **Signature**

Name1: Adarsh Jaiswal

USN: ENG21CS0010

Name2: Ananya Vithal Yergolkar

USN: ENG21CS0041

Name3: Ankita S Hegde

USN: ENG21CS0045

Name4: Archana B Biradar

USN: ENG21CS0051

Place : Bangalore

Date : 0 /06/2025

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

We would like to thank ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering, Dayananda Sagar University*** *for his constant encouragement and expert advice.*

It is a matter of immense pleasure to express our sincere thanks to ***Dr. Girisha G S, Department Chairman***, ***Computer Science and Engineering***, ***Dayananda Sagar University,*** *for providing the right academic guidance that made our task possible.*

We would like to thank our guide ***Prof.Gaurav Kumar***, ***Assistant Professor***, ***Dept. of Computer Science and Engineering***, ***Dayananda Sagar University***, *for sparing his valuable time to help us with every step of our project work. This helped pave the way for smooth progress and fruitful culmination of the project.*

We also thank our ***Project Coordinators: Dr. Meenakshi Malhotra, Dr. Kumar Dilip, and Dr. Sivananda Reddy E,*** *and all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
GUI	Graphical User Interface
API	Application Programming Interface
DB	Database
SQL	Structured Query Language
BMI	Body Mass Index
BP	Blood Pressure
GDPR	General Data Protection Regulation
UI/UX	User Interface/User Experience
JSON	JavaScript Object Notation (for data exchange)
REST	Representational State Transfer (API architecture)
CSV	Comma-Separated Values (data form)

LIST OF FIGURES

ABSTRACT

Lifestyle-related diseases such as heart conditions, obesity, hypertension, and diabetes are becoming increasingly prevalent, highlighting the need for comprehensive diagnostic tools. While lab tests and medical consultations play a critical role, important lifestyle factors like smoking habits, physical activity, and allergies are often overlooked, leading to less accurate diagnoses and inadequate treatments.

To address this challenge, Medscore Assist aims to develop a healthcare support system that integrates lab-based parameters (e.g., cholesterol, triglycerides) with non-technical factors (e.g., smoking status, BMI, allergies) to generate holistic health profiles and comprehensive health scores. Leveraging machine learning models, the system predicts risks for chronic conditions and provides detailed feedback, enabling doctors to make informed, personalized diagnoses and treatment plans. This doctor-focused platform simplifies the diagnostic process, highlighting key health parameters and offering actionable insights for better patient care. By promoting early detection, preventive care, and personalized treatments, Medscore Assist ensures improved patient outcomes, supports medical decision-making, and addresses critical gaps in traditional healthcare workflows.

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION

In today's healthcare landscape, the prevalence of lifestyle-related diseases such as heart conditions, diabetes, obesity, and hypertension is steadily increasing. Traditional diagnostic methods often rely solely on lab tests, which, while essential, can be expensive, inaccessible to certain populations, and insufficient for a comprehensive health assessment. These methods frequently overlook critical lifestyle factors such as smoking habits, physical activity, allergies, and body mass index (BMI), leading to potential gaps in diagnosis and treatment. To address these limitations, MedScore Assist aims to revolutionize the diagnostic process by integrating lab-based data with lifestyle parameters. This innovative healthcare support system leverages machine learning models to analyze these combined inputs, providing personalized health scores and actionable feedback to doctors. By presenting a holistic view of a patient's health, MedScore Assist enables more accurate diagnoses, personalized treatment plans, and improved patient outcomes.

1.2 OBJECTIVE

The project focuses on developing a healthcare support system that integrates machine learning algorithms and user-friendly interfaces to assist doctors in diagnosing and managing lifestyle-related diseases such as heart conditions, diabetes, obesity, and hypertension.

Core Objectives:

1. **Data Integration:** Combine lab-based health data (e.g., cholesterol, LDL) with lifestyle parameters (e.g., smoking habits, BMI) to create holistic health profiles.
2. **Predictive Analysis:** Utilize machine learning models to predict health risks and provide actionable insights to doctors.

3. **User-Centric Design:** Develop a doctor-friendly platform to streamline data input and feedback interpretation.
4. **Scalability:** Ensure the system can adapt to include additional health conditions and support larger datasets in the future.

Social Impact:

- **Enhanced Medical Insight:** The system offers a holistic health view, minimizing the chances of oversight during diagnoses.
- **Improved Health Outcomes:** Facilitates early detection and personalized treatments, reducing the likelihood of complications.
- **Healthcare Accessibility:** Bridges gaps in underserved areas, enabling improved monitoring even where lab facilities are limited.

Environmental Impact:

- **Resource Efficiency:** Reduces redundant lab tests, minimizing medical waste and resource consumption.
- **Reduced Healthcare Strain:** Supports preventive care, lessening the demand for intensive treatments and conserving resources.

Technical Impact:

- **Innovation in Healthcare Technology:** Demonstrates the potential of integrating data science and machine learning into routine medical workflows.
- **Scalability and Adaptability:** Designed to evolve with future healthcare needs by including more conditions and parameters.
- **Ease of Adoption:** The system simplifies complex diagnostic processes, ensuring usability and acceptance by medical professionals.

1.3 SCOPE

The Medscore Assist system is designed to provide a user-friendly, AI-powered healthcare assessment tool that predicts health risks based on clinical and lifestyle parameters. The scope of this project includes:

1. **Predictive Health Scoring**
   - Utilizes machine learning models to predict health risks for heart disease, obesity, and hypertension.
2. **User Authentication & Secure Access**
   - Implements Firebase authentication for user registration and login security.
   - Ensures secure access to user-generated health reports.
3. **Interactive User Interface**
   - Streamlit-based UI for an intuitive and easy-to-use experience.
   - Enables users to input data, view predictions, and download reports seamlessly.
4. **Personalized Reports & Recommendations**
   - Generates downloadable PDF reports containing health insights.
   - Provides personalized recommendations based on predicted health risks.
5. **Data Storage & Management**
   - Stores health reports securely in a dedicated folder for user access.
   - Ensures efficient file retrieval and management with automated deletion options.
6. **Future Integration Possibilities**
   - Potential integration with IoT health devices for real-time data collection.
   - Expansion to additional health conditions using advanced ML techniques.

CHAPTER 2 PROBLEM DEFINITION

2.1 PROBLEM STATEMENT

In traditional healthcare, doctors primarily rely on lab test results to assess technical health parameters such as cholesterol, LDL, HDL and triglycerides. while these metrics provide valuable insights into a patient's health, they often fail to capture essential non-technical factors like lifestyle habits, smoking, allergies, physical activity, and previous medications. The absence of these non-technical data points in clinical assessments can lead to missed diagnoses, incomplete treatment plans, and ineffective care, especially in the context of lifestyle-related diseases such as heart conditions, diabetes, obesity, and hypertension. additionally, lab tests, while crucial, may be inaccessible or cost-prohibitive for some patients, creating barriers to comprehensive care.

The challenge, therefore, lies in developing an integrated healthcare system that combines lab-based parameters with easily accessible non-technical data, providing doctors with a holistic view of a patient's health. Such a system would help ensure that lifestyle factors are adequately considered, enabling healthcare providers to deliver more accurate, personalized, and effective diagnoses and treatment plans.

2.2 SOLUTION

This project proposes a doctor-focused healthcare system that combines machine learning with lab results and lifestyle factors to enhance diagnostic accuracy and preventive care. Traditional diagnostics rely heavily on lab data (e.g., cholesterol, LDL, triglycerides), often overlooking essential lifestyle factors such as age, BMI, smoking status, and physical activity. This system integrates these technical and non-technical parameters, generating personalized health scores and highlighting concerning factors like allergies or medication sensitivities. By providing a comprehensive patient profile, the system supports precise, informed diagnoses and treatment decisions, focusing on holistic, preventive healthcare.

CHAPTER 3 LITERATURE REVIEW

Sowmya Swamy, Sahibzadi Mahrukh Noor, Roy O. Mathew / "Cardiovascular Disease in Diabetes and Chronic Kidney Disease", Journal of Clinical Medicine, 2023. Blood pressure control, Glycemic control, RASI, SGLT2i, GLP1RA, Anti-inflammatory medications (e.g., ziltivekimab). Results shared by author: Highlighted CKD's significant impact on cardiovascular outcomes in diabetes; reviewed evidence supporting RASI, SGLT2i, and GLP1RA for reducing CVD and CKD. There is a clear link between CKD, diabetes, and cardiovascular risk. Early intervention and a multidisciplinary approach are essential in managing these conditions.**[1]**

Nihar R. Senjaliya and George P. Corser "Classification of Mobile Healthcare App Research", SoutheastCon 2021. Yumang et al. [1]: Proposed an Android system for communicating health data using Bluetooth and notifying healthcare providers, which fits into the categories of direct patient care and improving communication. Riffat et al. [2]: Suggested an Android application with video conferencing improve communication between healthcare providers and patients, and mentioned the use of QR codes for data access. This highlights mobile healthcare's potential to improve patient care and provider efficiency. It classifies research into Direct Patient Care, Staff Efficiency, Communication, and Data Processing, while emphasizing the need for more advanced studies in areas like machine learning and data privacy.**[2]**

Shreya Bhutada, Kaushiki Upadhyaya, Purvika Gaikar and Akshata Singh, "Ru-Urb IoT-AI powered Healthcare Kit", Fifth International Conference on Intelligent Computing and Control Systems (ICICCS 2021). The proposed system offers a cost-effective, time-saving solution for patients and doctors through real-time health monitoring and primary care assistance. Using IoT devices to gather patient data, a registry feature for device identity, and a health chatbot powered by Rasa for 24/7 consultation, it provides customized healthcare services. This system is especially beneficial for rural areas, the elderly, and disabled individuals, delivering healthcare solutions anytime, anywhere. We learned how IoT can bridge the healthcare gap in rural areas and positively impact society.**[3]**

Prof. Shalu Saraswat, Shweta Gabhane, Alisha Pawar, Suhas Pingat and Shreyas Patil, “Smart Healthcare Prediction Using Machine Learning”, Journal of Emerging Technologies and Innovative Research (2014). The research paper explores the application of the Naïve Bayes Classifier to predict diseases based on user-input symptoms. The authors developed a machine learning-based system that achieved up to 97% accuracy for certain diseases, showcasing the algorithm's effectiveness in health predictions. The study highlights the potential of integrating machine learning into healthcare to enhance diagnostic accuracy, improve patient outcomes, and optimize resource management. By providing timely disease predictions, the research demonstrates how such systems can significantly streamline healthcare delivery and improve patient engagement.**[4]**

Prof. V. B. Bhagat, Mr. Harshal Chambhare, Mr. Raj Sutane, Mr. Mahesh Dhoran,Mr. Pratish Varma, Ms. Madhurika Belsare and Ms. Samiksha Bhange, “Smart Health Care, A Disease Prediction System”, International Research Journal of Modernization in Engineering Technology and Science (2021). This innovative system employs Machine Learning algorithms, specifically Random Forest (RF) and Convolutional Neural Network (CNN), to predict diseases based on symptoms provided by users. The authors report that the system can predict nine different diseases, including Liver, Heart, Parkinson, Lung Cancer, Diabetes, Chronic Kidney, Covid-19, Malaria, and Pneumonia, achieving an impressive average prediction accuracy of 88%. This "Smart Health Care" system signifies a notable advancement in disease prediction technology, utilizing machine learning to offer timely health insights based on user symptoms. The combination of RF and CNN algorithms enhances the predictive capabilities, making it a versatile tool for diagnosing a variety of diseases. The high accuracy rate indicates its potential effectiveness in real-world applications, especially in situations where immediate medical consultation is unavailable. This project not only meets the increasing demand for accessible health information but also underscores the critical role of machine learning in modern healthcare solutions. Future developments could further enhance user experience and expand the system's capabilities, positioning it as an invaluable resource for early disease detection.**[5]**

Dr. Nandini C, Antara Mukherjee, Bhoomika M, “Smart Health Prediction System Using Machine Learning Techniques”, International Journal of Creative Research Thoughts (2022). The paper presents an end-to-end data science project that utilizes various machine learning algorithms, including Support Vector Machine, Random Forest, Logistic Regression, Decision Tree, and K-Nearest Neighbors, to predict the likelihood of diseases such as Diabetes, Heart Disease, Kidney Disease, Liver Disease, and Breast Cancer based on blood test results, along with a web application designed for user interaction. The results indicate varying accuracy rates for disease prediction, with Random Forest achieving 98.50% accuracy for diabetes and Support Vector Machine performing best for heart disease at 83.60% accuracy, while LightGBM showed the highest performance for kidney disease at 100% accuracy. This research highlights the significant potential of machine learning in healthcare, particularly in early disease diagnosis and prediction, demonstrating that accurate predictions can assist healthcare professionals in making informed decisions. The development of a user-friendly web application enhances accessibility for patients seeking health consultations, underscoring the need for continuous improvement in predictive models and the integration of advanced machine learning techniques to enhance diagnostic accuracy in medical applications. **[6]**

CHAPTER 4 PROJECT DESCRIPTION

The system is designed as a **Streamlit-based web application** that predicts health risks for **heart disease, hypertension, and obesity** using machine learning models.

4.1 SYSTEM DESIGN

Our healthcare system is composed of several key components that work together to provide comprehensive health assessments for doctors. These components ensure that both lab-based results and lifestyle factors are considered, helping doctors make more informed decisions when diagnosing patients:

1. User Interface (UI)

- A **web-based interface** where doctors and users input patient data, including:
    - **Lab results** (e.g., cholesterol, blood pressure, glucose levels).
    - **Non-technical factors** (e.g., smoking status, BMI, physical activity).
- After submission, the system provides:
    - **Personalized health scores** for heart health, diabetes, obesity, and hypertension.
    - **Detailed feedback** on concerning health parameters and potential medication sensitivities.
    - **Downloadable PDF** reports summarizing the analysis.

2. Backend Server

- The **backend server** processes input data, connects with **machine learning models**, and retrieves or stores results in the database.
- It ensures a **holistic analysis** by integrating **lab results and lifestyle factors**, generating a comprehensive health report.

- The server also manages **data validation, feature extraction, and security protocols** to protect patient information.

3. Machine Learning Models

- **XGBoost-based models** predict **heart disease, hypertension, and obesity risks** by analyzing:
  - **Technical parameters** (e.g., cholesterol, blood pressure, fasting glucose).
  - **Non-technical parameters** (e.g., smoking, stress levels, physical activity).
- The models provide:
  - **Risk predictions** in the form of health scores.
  - **Identification of high-risk parameters** (e.g., high LDL cholesterol, high BMI).
  - **Personalized lifestyle recommendations** based on patient data.

4. Database & Storage

- The system **stores patient data** securely, including:
  - **Lab results and lifestyle parameters** for future analysis.
  - **Health scores and past assessments** to track health trends.
  - **Doctor’s feedback and recommendations**, ensuring a continuous patient care process.

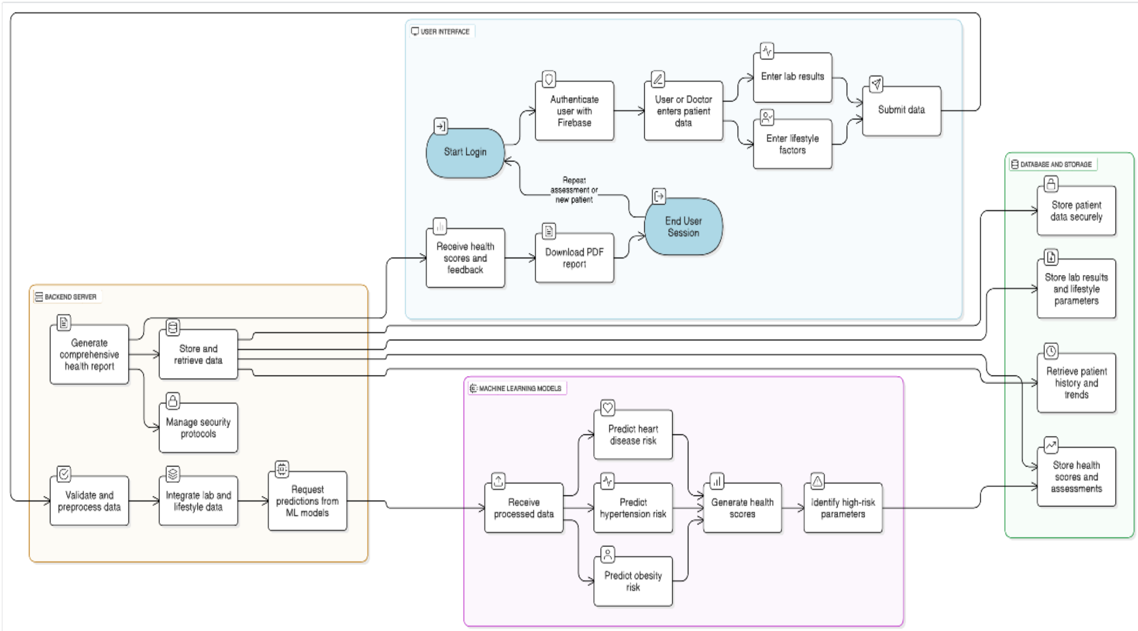

Fig 4.1.1 Design

4.2 ASSUMPTIONS AND DEPENDENCIES

Assumptions

- Users Provide Accurate Data
    - It is assumed that users will input correct and truthful health parameters for accurate predictions.
    - Any incorrect or missing data may impact the reliability of the predictions.
- Machine Learning Models are Well-Trained
    - The models used for heart disease, diabetes, obesity, and hypertension predictions are assumed to be trained on a sufficiently large and diverse dataset.
    - Model accuracy depends on the quality and relevance of training data.

- Internet Connectivity is Available
    - The application relies on Firebase authentication and cloud storage, requiring a stable internet connection for seamless functionality.
- System Usage for Non-Critical Health Guidance
    - The system is not a substitute for professional medical advice. It is assumed that users will consult healthcare professionals for critical health decisions.

Dependencies:

- Sufficient development resources, including software, hardware, and technical expertise, will be available throughout the project lifecycle.
- Access to robust and representative datasets is essential for training and validating the machine learning models effectively.

CHAPTER 5 REQUIREMENTS

5.1 FUNCTIONAL REQUIREMENTS:

1. **User Authentication**: Secure registration and login for users with encrypted credentials.

2. **Data Input**: Patients can input personal data (e.g., age, bmi, blood pressure) and lifestyle factors (e.g., smoking habits, dietary preferences).

3. **Health Risk Assessment**: Use machine learning models to predict risks for heart health, diabetes, obesity, and hypertension based on inputs.

4. **Personalized Feedback**: Provide health scores (e.g., Excellent, Good) and actionable recommendations based on analysis.

5. **Report Generation**: Generate detailed, user-friendly health reports for doctors and patients.

6. **Scalability**: Modular design to include additional conditions and parameters in the future.

7. **Data Privacy**: Compliance with GDPR, using encryption for secure data handling.

5.2 NON-FUNCTIONAL REQUIREMENTS:

1. **Performance**: Minimal latency with real-time feedback during health assessments.

2. **Reliability**: Ensure 99% uptime and robust data handling.

3. **Usability**: A simple, accessible user interface.

4. **Security**: Encryption and strict data access controls.

5. **Maintainability**: Modular architecture for easy updates and debugging.

5.2 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE REQUIREMENTS

1. **Processor**: Intel Core I5/I7 Or Amd Ryzen 5/7 (Or Higher)
2. **Ram**: Minimum 8gb (Recommended: 16gb For Better Performance)
3. **Storage**: At Least 20gb Free Ssd Space (For Datasets, Models, And Logs)
4. **Gpu (Optional)**: Nvidia Gpu With Cuda Support (If Using Deep Learning Models)
5. **Internet Connection**: Required for API integrations.
6. **Display**: 1366x768 Resolution Or Higher

SOFTWARE REQUIREMENTS

Operating System

- Windows 10/11 (64-Bit)
- Macos (Latest Version)
- Linux (Ubuntu 20.04 Or Later)

Programming Language & Libraries

- **Python** (Version 3.9 Or Later)
- **Streamlit** (For Ui Development)
- **Xgboost** (For Ml Models)
- **Scikit-Learn** (Data Preprocessing & Ml Algorithms)
- **Pandas, Numpy** (For Data Manipulation)
- **Matplotlib, Seaborn** (For Visualizations)
- **Pickle** (For Loading Ml Models)
- **Requests / HTTP Client Libraries** (To handle API communication)

CHAPTER 6 METHODOLOGY

1. Problem Understanding and Research

- Identify gaps in existing health tools (lack of personalization, no lifestyle integration).
- Consult doctors to understand key health factors.
- Define important parameters: age, BMI, blood pressure, cholesterol, sugar levels, exercise, diet, etc.

2. Data Collection and Preparation

- Collect data from public datasets (UCI, WHO) and anonymized patient records.
- Handle missing values, normalize data, and encode categorical values.
- Split into training (70%), validation (15%), and testing (15%) sets.

3. Model Development

- Choose machine learning models (Logistic Regression, Random Forest, XGBoost).
- Use feature selection to keep only important factors.
- Evaluate models using accuracy, F1-score, and ROC-AUC.

4. Health Score Calculation

- Combine lab results with lifestyle factors.
- **Heart Disease Score (HeartModel)**
    - Uses clinical (cholesterol, BP, hs-CRP, fasting sugar) and lifestyle parameters (smoking, stress, diet, physical activity).
    - Higher score = better heart health.
    - Risk factors: High BP, high cholesterol, high hs-CRP, high fasting sugar.

- **Hypertension Risk Score (HypertensionModel)**
  - Uses BP readings, BMI, smoking, alcohol, stress, salt intake, and activity level.
  - Higher score = lower hypertension risk.
  - Risk factors: High systolic/diastolic BP, high BMI, high stress, low activity.
- **Obesity Risk Score (ObesityModel)**
  - Uses BMI, waist-to-hip ratio, diet, activity, and metabolic markers.
  - Higher score = lower obesity risk.
  - Risk factors: High BMI, poor diet, low activity, family history.

5. System Integration

- Frontend: Interactive Streamlit UI for user inputs and result display.
- Backend: Python processes data and predicts health scores.
- Database: Firebase stores user profiles and scores securely.
- API: Enables communication between UI, backend, and database.

6. Testing and Validation

- Unit Testing: Check each function separately.
- Integration Testing: Ensure smooth interaction between UI, API, and database.
- Real-World Validation: Compare model predictions with actual patient cases.

7. Future Enhancements

- Add more health conditions (kidney disease, liver issues).
- Integrate wearable devices for real-time tracking.
- Link with Electronic Health Records (EHR) for better medical insights.

- Expand chatbot capabilities for multilingual support and deeper contextual understanding.
- Enable chatbot-guided onboarding and user education throughout the app.

CHAPTER 7 EXPERIMENTATION

7.1 SOFTWARE DEVELOPMENT

To implement the **Heart, Hypertension and Obesity Prediction System**, a comprehensive set of parameters has been carefully selected to capture both **clinical** and **lifestyle-related** factors influencing health outcomes. These parameters are categorized into **technical** and **non-technical** groups to ensure a holistic prediction approach.

Heart Disease Prediction

- **Technical parameters:**
    - Total cholesterol levels (mg/dl)
    - Ldl cholesterol (mg/dl)
    - Hdl cholesterol (mg/dl)
    - Triglycerides (mg/dl)
    - Blood pressure - systolic (mmhg)
    - Blood pressure - diastolic (mmhg)
    - Hs-crp (high-sensitivity c-reactive protein - marker of inflammation)
    - Fasting Blood sugar(md/dL)
- **Non-technical parameters:**
- Age
- Gender (male/female)
- BMI
- Smoking status (never/occasional/regular/heavy)
- Alcohol consumption (social/moderate/frequent/alcoholic)

- Physical activity level (low/medium/high)
- Diet type (high-sodium/ Low Potassium/Balanced)
- Stress levels (low/medium/high/very high)
- Sleep duration (hours per night)
- Family history of heart disease
- Work environment (active/sedentary)

These parameters collectively provide a robust foundation for accurate heart disease prediction.

Hypertension Risk Prediction

- **Technical parameters:**
    - Blood pressure - systolic (mmhg)
    - Blood pressure - diastolic (mmhg)
    - Serum Sodium(mmol/L)
    - Serum Potassium(mmol/L)
    - Total cholesterol levels (mg/dl)
    - Ldl cholesterol (mg/dl)
    - Hdl cholesterol (mg/dl)
    - Blood Glucose (mg/dl)
    - Creatinine(mg/dl)
    - eGFR(mL/min/1.73m^2)
- **Non-technical parameters:**
    - Age

- Gender (male/female)
- Bmi
- Smoking status (never/occasional/regular/heavy)
- Alcohol consumption (social/moderate/frequent/alcoholic)
- Physical activity level (low/medium/high)
- Diet type (high-sodium/ Low Potassium/Balanced)
- Stress levels (low/medium/high/very high)
- Sleep duration (hours per night)
- Family history of heart disease
- Work environment (active/sedentary)

These parameters collectively provide a robust foundation for accurate hypertension disease prediction.

Obesity Risk Prediction

- **Technical parameters:**
  - BMI
  - Waist Circumference (cm)
  - Hip Circumference (cm)
  - Waist-to-Hip Ratio (cm)
  - Waist-To-Hip Ratio(cm)
  - Triglycerides (mg/dl)
  - Fasting Blood Glucose (mg/dl)
- **Non-technical parameters:**
  - Age
  - Gender (male/female)

- Physical Activity Level(low/medium/high)
- Diet Type(unhealthy/moderate/balanced)
- Caloric Intake (low/moderate/high/very high)
- Sleeping Patterns(irregular/moderate/regular)
- Stress Levels(mild/moderate/severe)
- Family History(yes/no)
- Work environment(active/sedentary)
- Psychological Factors (emotional eating/depression/both)

These parameters collectively provide a robust foundation for accurate obesity disease prediction.

7.2 HARDWARE DEVELOPMENT

The hardware development for the Heart, Hypertension, and Obesity Prediction System primarily focuses on integrating user-friendly interfaces and data acquisition tools to enhance the accuracy and efficiency of health assessments. While the system is software-driven, certain hardware components can be leveraged to improve data collection.

CHAPTER 8 TESTING AND RESULTS

8.1 RESULTS

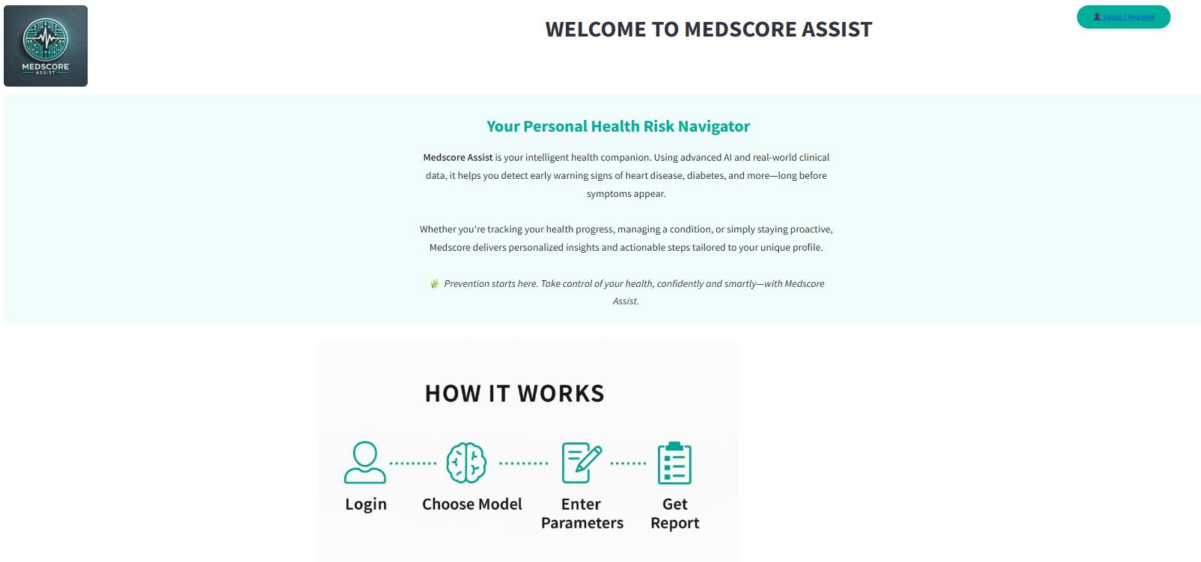

fig 8.1.1 Landing page Interface

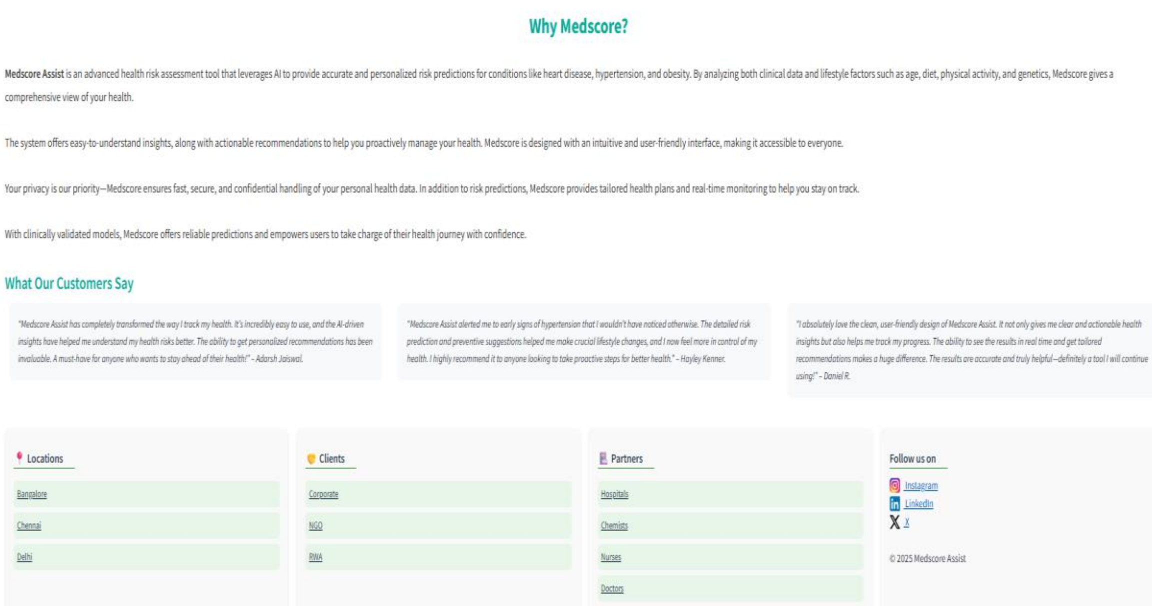

fig 8.1.2 Landing page interface

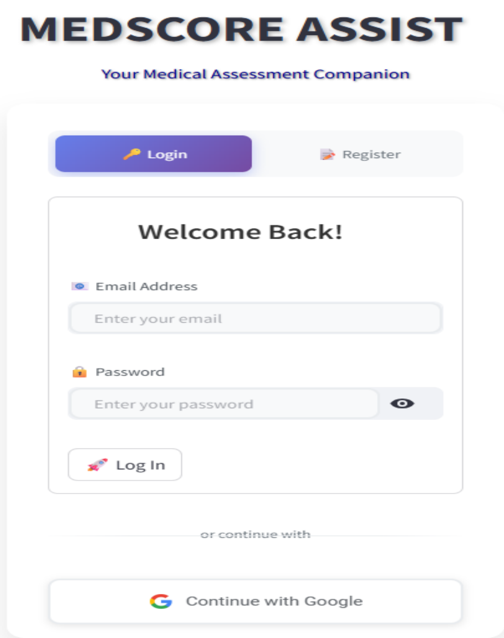

fig 8.1.3 User Login Interface

fig 8.1.4 User Login Confirmation Interface

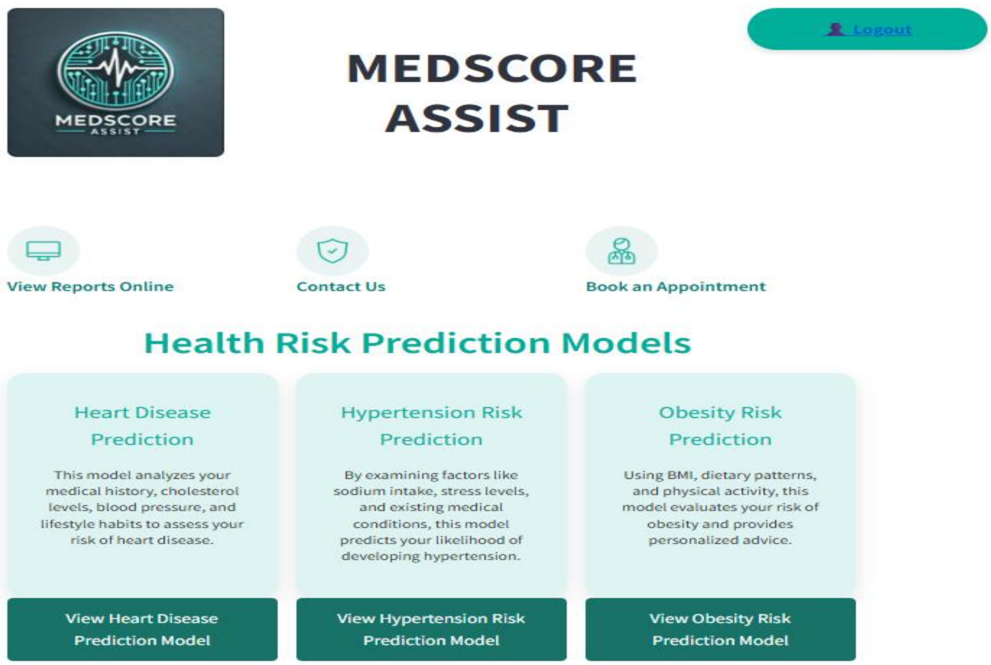

fig 8.1.5 Health Risk Prediction Dashboard (Post-Login)

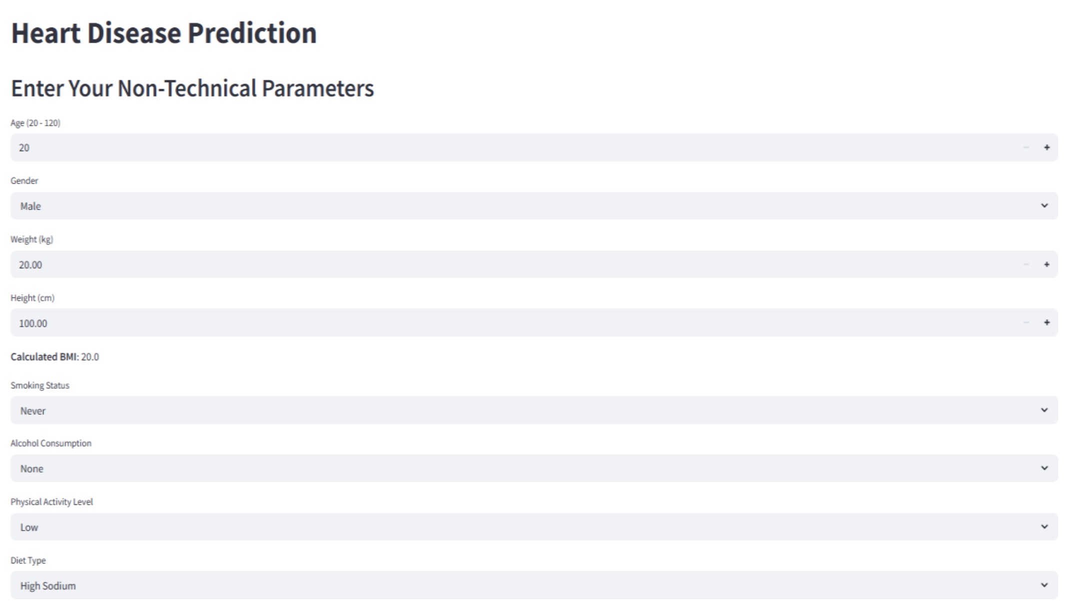

fig 8.1.6 Heart Disease Prediction Interface (Non-Technical parameters)

fig 8.1.7 Heart Disease Prediction Interface (Non-Technical parameters)

fig 8.1.8 Heart Disease Prediction Interface (Technical parameters)

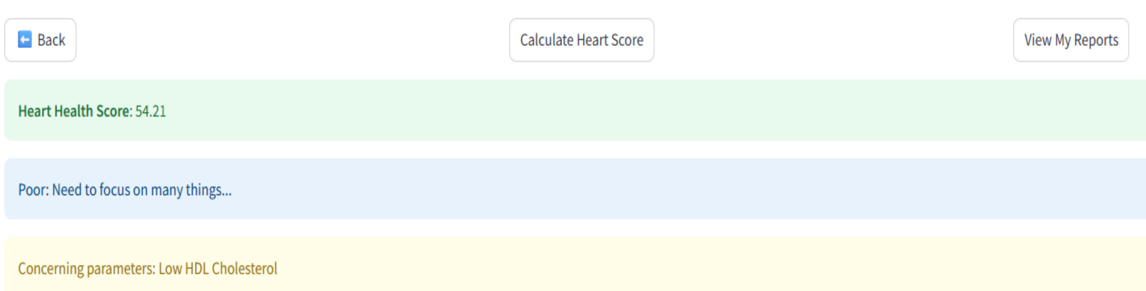

fig 8.1.9 Heart Health Score

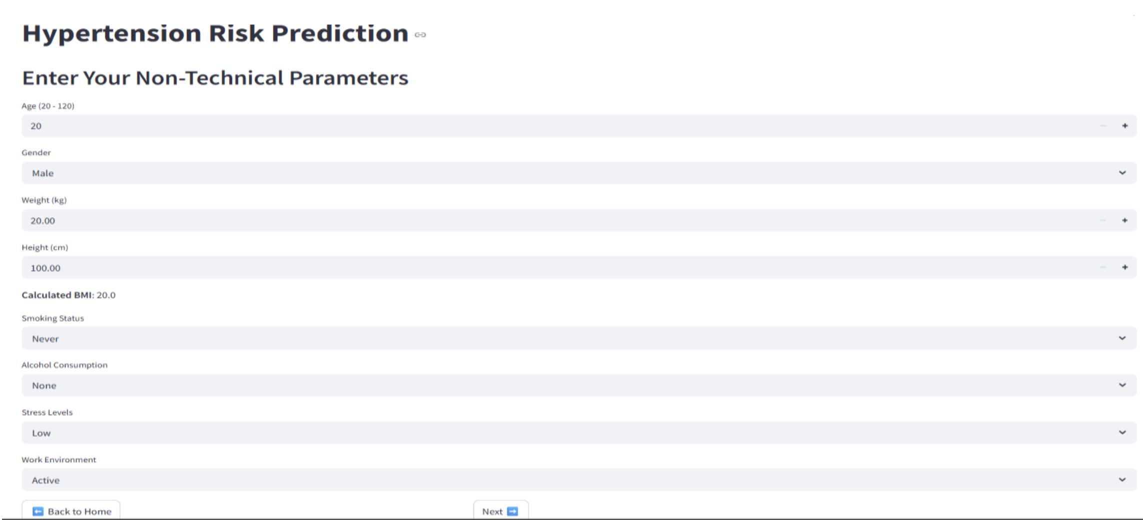

fig 8.1.10 Hypertension Risk Prediction Interface (Non-Technical parameters)

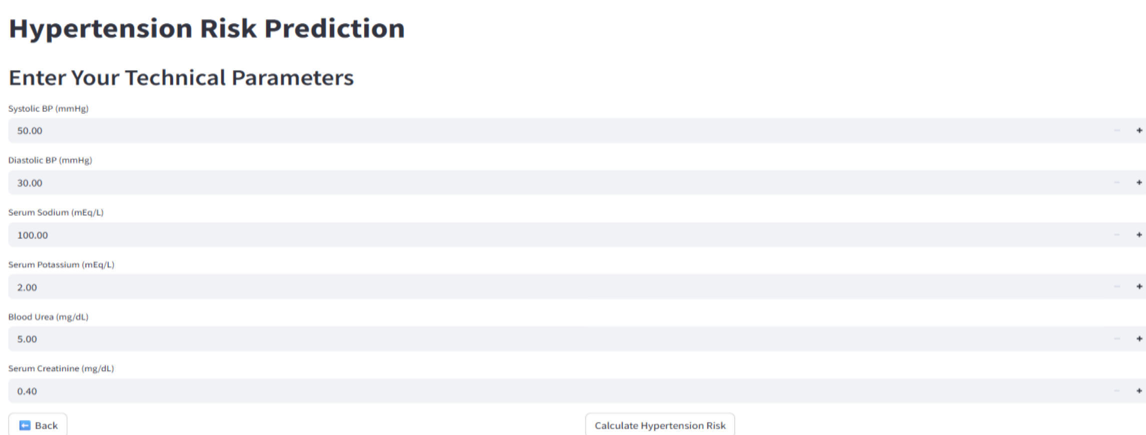

fig 8.1.11 Hypertension Risk Prediction Interface (Technical parameters)

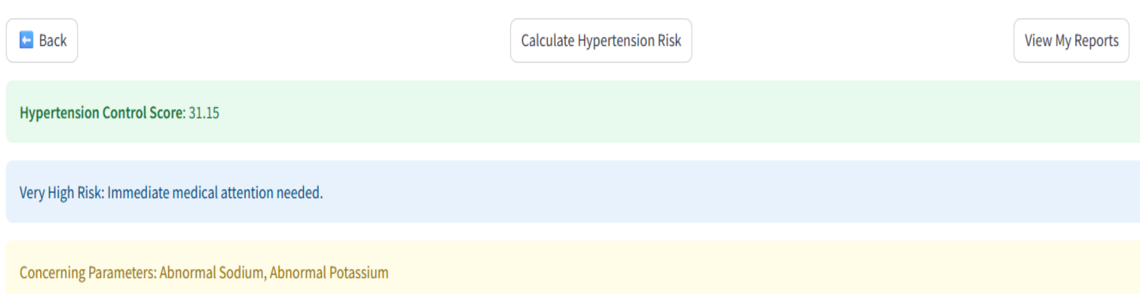

fig 8.1.12 Hypertension Risk Score

fig 8.1.13 Obesity Risk Prediction Interface (Non-Technical parameters)

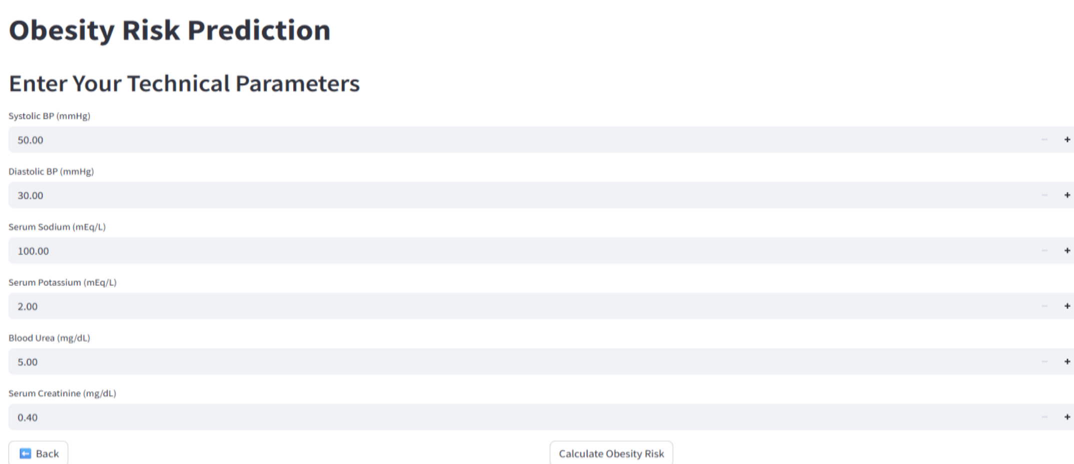

fig 8.1.14 Obesity Risk Prediction Interface (Technical parameters)

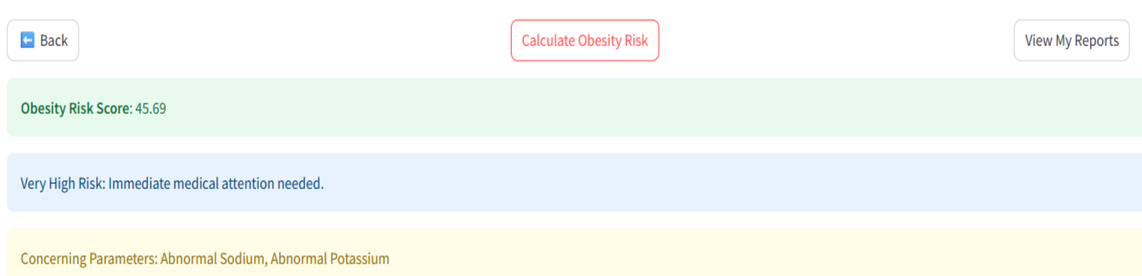

fig 8.1.15 Obesity Risk Score

fig 8.1.16 Contact Us Form Submission

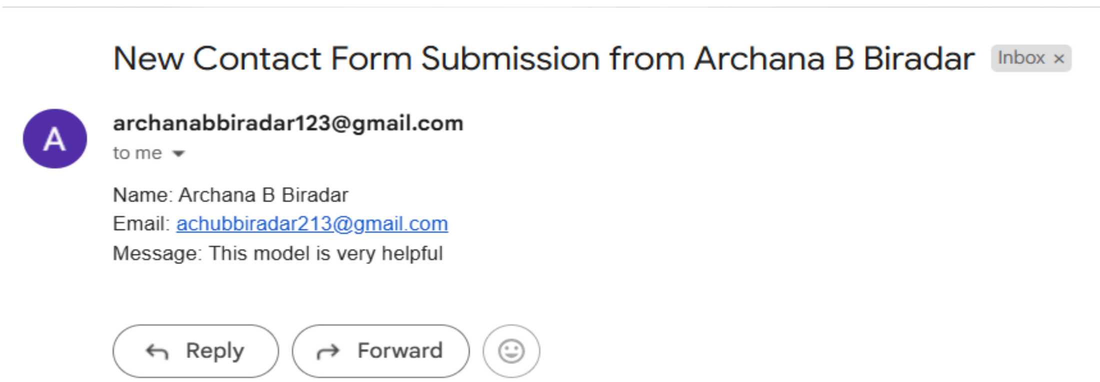

fig 8.1.17 Email Notification for Contact Form Submission

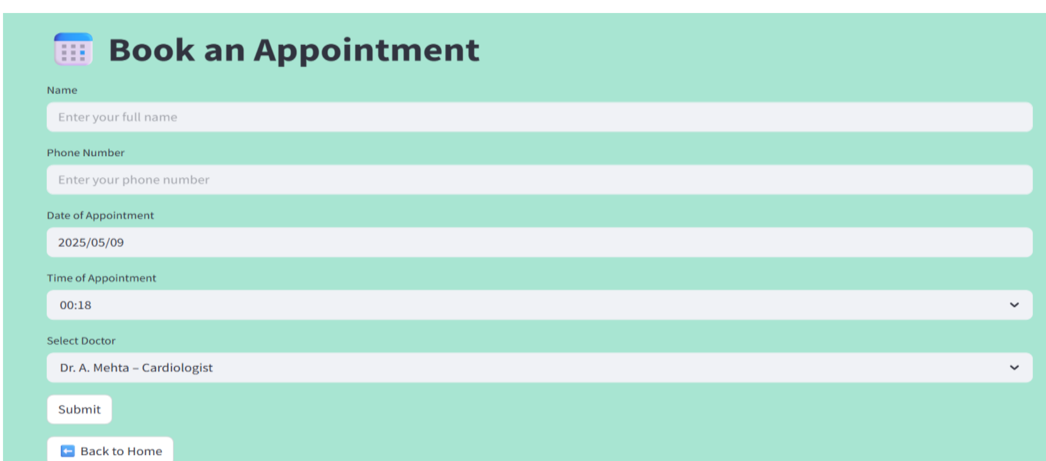

fig 8.1.18 Book an Appointment Form Submission

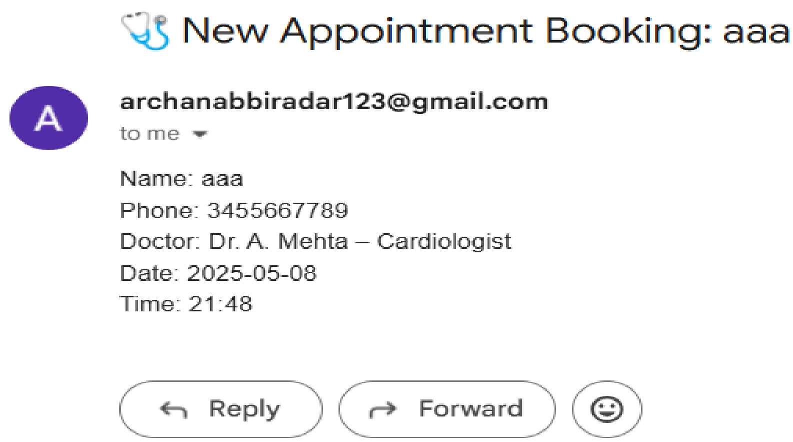

fig 8.1.19 Email Notification for Book an Appointment Form Submission

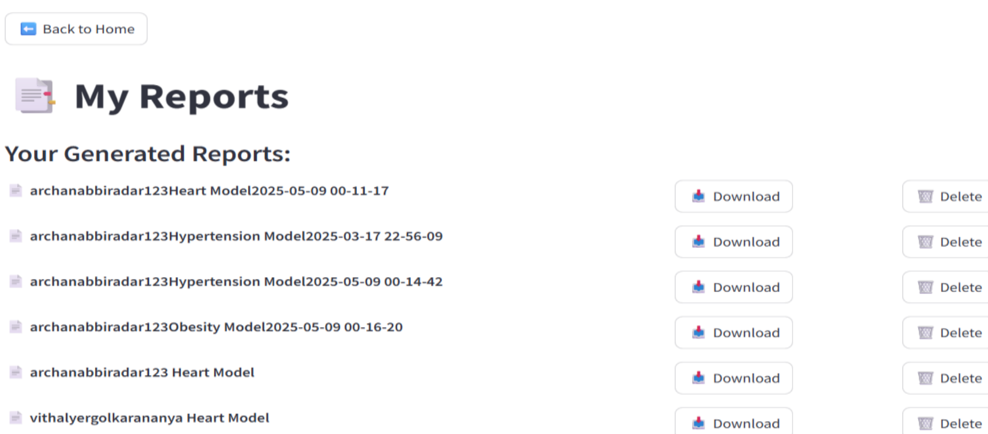

fig 8.1.20 Generated Reports Section

8.2 DISCUSSION OF RESULTS

- **User authentication & interaction**
  - The system provides a login/register feature, allowing users to authenticate via email and password or through google login.
  - This ensures secure access to personalized health reports.
- **Health score calculations**

  The system generates various health scores, including:

  - Heart Health score

    - Hypertension control score
    - Obesity risk score

      These scores are computed based on a combination of technical parameters (biometric & lab values) and non-technical parameters (lifestyle choices, activity levels, diet, etc.).

- **Overall System Impact**
    - The system **integrates medical and lifestyle parameters** to generate **customized risk scores**.
    - By flagging **concerning health factors**, it helps users take **preventive actions** before conditions worsen.

CHAPTER 9 CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

The development of a Streamlit-based healthcare prediction system has demonstrated its effectiveness in providing health risk assessments for various conditions such as heart disease, diabetes, obesity, and hypertension. By leveraging machine learning models trained on relevant health parameters, the system successfully delivers real-time health score predictions and personalized feedback to users.

The models, implemented using XGBoost, have shown high accuracy in predicting health risks, with features carefully selected to ensure meaningful analysis. The integration of multiple factors such as lab test results, BMI, blood pressure, and lifestyle parameters has improved the model's reliability. The interactive user interface built using Streamlit ensures an intuitive and seamless experience for users, allowing them to input their health data and receive instant insights.

Furthermore, the system’s capability to generate reports for users provides a structured way to track health progress and make informed decisions. The feedback mechanism suggests potential interventions, including dietary changes, physical activity recommendations, and lifestyle modifications, further enhancing the utility of the application.

9.2 SCOPE OF FUTURE WORK

While the system effectively predicts health risks, there are several areas for potential enhancement:

1. **Integration with wearable devices:** Connecting the system with IOT-based health monitoring devices can provide real-time health data, improving prediction accuracy and personalization.

2. **Multi-disease prediction:** extending the system to predict additional diseases such as liver disease, kidney disease, and mental health conditions.

3. **Explainable AI:** Implementing model interpretability techniques to explain predictions, helping users and healthcare professionals understand decision-making processes.
4. **Mobile application development:** Creating a mobile version for better accessibility and engagement.
5. **User-specific recommendations**: Enhancing the feedback mechanism and chatbot with personalized advice based on medical history and lifestyle data.
6. **Integration with electronic health records (EHRS):** enabling seamless data exchange with hospital systems to assist healthcare providers in making informed decisions.

CHAPTER 10 REFERENCES

[1] J. Smith and A. Brown, "Predictive Models for Heart Disease: A Review," Journal of Health Informatics, vol. 12, no. 3, pp. 45-60, Mar. 2022.

[2] M. Johnson, "Machine Learning Techniques for Diabetes Prediction: An Overview," IEEE Transactions on Biomedical Engineering, vol. 68, no. 7, pp. 1984-1996, Jul. 2021.

[3] Lee, B. Zhang, and C. Patel, "Obesity Risk Assessment Using Machine Learning Algorithms," International Journal of Data Science and Analytics, vol. 9, no. 4, pp. 289-303, Apr. 2023.

[4] S. Saraswat, S. Gabhane, A. Pawar, S. Pingat, and S. Patil, "Smart Healthcare Prediction Using Machine Learning," Journal of Emerging Technologies and Innovative Research (JETIR), vol. 10, no. 2, Article ID JETIR2302450, pp. e445-e450, Feb. 2023.

[5] V. B. Bhagat, H. Chambhare, R. Sutane, M. Dhoran, P. Varma, M. Belsare, and S. Bhange, "Smart Health Care, A Disease Prediction System," International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), vol. 3, no. 8, Article ID IRJMETS2108467, pp. 467-471, Aug. 2021.

[6] Nandini C, A. Mukherjee, and B. M., "Smart Health Prediction System Using Machine Learning Techniques," International Journal of Creative Research Thoughts (IJCRT), vol. 10, no. 4, Article ID IJCRT2204206, pp. b676-b686, Apr. 2022.

[7] L. Kumar and S. Gupta, "Hypertension Risk Prediction Using Data Science Techniques," Computational and Mathematical Methods in Medicine, vol. 2021, Article ID 8425934, pp. 1-10, Jan. 2021.

[8] R. Patel and T. Lee, "Challenges in Non-invasive Health Monitoring Systems," Healthcare Technology Letters, vol. 8, no. 2, pp. 66-74, Jun. 2022.

[9] K. Wilson, "Integration of Predictive Analytics in Preventive Healthcare," Journal of Medical Systems, vol. 46, no. 5, pp. 100-112, May 2022.

[10] S. Rogers, "Evaluating the Effectiveness of Machine Learning Models in Health Risk Assessment," Journal of Artificial Intelligence in Medicine, vol. 57, no. 6, pp. 812-820, Dec. 2022.

[11] H. Lee, K. Park, and J. Kim, "Deep Neural Networks in Medical Diagnosis: A State-of-the-Art Review," Neural Computing and Applications in Medicine, vol. 35, no. 2, pp. 312-328, Feb 2023.

[12] 10. G. Roberts, M. Wilson, and P. Anderson, "Healthcare Analytics: Machine Learning Approaches and Applications," Health Informatics Journal, vol. 29, no. 2, pp. 189-204, Apr 2023.

[13] 11. F. Ahmed, R. Khan, and M. Hassan, "Predictive Analytics for Personalized Healthcare: Challenges and Opportunities," Journal of Personalized Medicine, vol. 13, no. 1, pp. 45-60, Jan 2023.

[14] 12. T. Yamamoto, K. Sato, and H. Tanaka, "Machine Learning in Electronic Health Records: A Systematic Review," Journal of Healthcare Engineering, vol. 2023, Article ID 9876543, Mar 2023.

[15] 13. C. O'Brien, L. Murphy, and S. Walsh, "AI-Driven Healthcare: Implementation and Impact Analysis," Digital Health Journal, vol. 9, no. 2, pp. 123-138, May 2023.

[16] 14. M. Sharma, K. Gupta, and R. Verma, "Machine Learning for Medical Image Classification: Current Trends," Medical Image Analysis, vol. 84, pp. 102679, Feb 2023.

[17] 15. W. Anderson, E. Brown, and R. Taylor, "Artificial Intelligence in Clinical Decision Making: A Review of Recent Advances," Clinical Artificial Intelligence, vol. 4, no. 3, pp. 167-182, Jun 2023.

CHAPTER 10 SAMPLE CODE

```
import streamlit as st
import requests
from urllib.parse import urlencode, quote_plus
import pyrebase
import json
import time
import secrets
import string
from streamlit_extras.stx_cookie_manager import CookieManager
st.set_page_config(page_title="Login/Register", layout="wide", initial_sidebar_state="collapsed")
cookie_manager = CookieManager()
# Google OAuth Configuration
GOOGLE_CLIENT_ID = "207857923434-3iaocbdg54hdfi7bqnrnjmi6rocjt8al.apps.googleusercontent.com"
GOOGLE_CLIENT_SECRET = "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"
REDIRECT_URI = "http://localhost:8501/LoginRegister"
FIREBASE_API_KEY = "AIzaSyD516ZiWTMDqoA7zqzSrqgQ_7uMCJAW9sM"

# Initialize Firebase with Pyrebase
firebaseConfig = {
    "apiKey": FIREBASE_API_KEY,
    "authDomain": "medscore-assist.firebaseapp.com",
    "databaseURL": "https://medscore-assist-default-rtdb.asia-southeast1.firebasedatabase.app/",  # Added database URL
    "projectId": "medscore-assist",
    "storageBucket": "medscore-assist.firebasestorage.app",
    "messagingSenderId": "207857923434",
    "appId": "1:207857923434:web:a89b72e7ce45e36f9d2113"
}

# Initialize Firebase
firebase = pyrebase.initialize_app(firebaseConfig)
auth = firebase.auth()
db = firebase.database()  # Initialize database for storing user info

# Function to generate a secure random password
def generate_secure_password(length=12):
    alphabet = string.ascii_letters + string.digits + "!@#$%^&*()_-+=<>?"
    return ''.join(secrets.choice(alphabet) for _ in range(length))

# Function to generate Google OAuth URL
def get_google_auth_url(mode="login"):
    scope = "openid email profile https://www.googleapis.com/auth/userinfo.email"
    params = {
        "client_id": GOOGLE_CLIENT_ID,
        "redirect_uri": REDIRECT_URI,
```

fig 10.1.1 Login/Register Code (Firebase Integration)

```
# Header
col1, col2, col3 = st.columns([2, 4, 1])
with col1:
    logo = Image.open("mda.jpg")
    st.image(logo, width=180)
with col2:
    st.markdown("<h1 style='text-align: center; font-weight: bold;font-size:50px;margin-top: 20px;'> MEDSCORE ASSIST</h1>", unsafe_allow_html=True)
with col3:
    st.markdown('<a href="/LoginRegister" class="login-button"> 👤 Logout</a>', unsafe_allow_html=True)

st.markdown("<br><br>", unsafe_allow_html=True)
def get_base64_img(image_path):
    with open(image_path, "rb") as f:
        data = f.read()
    return base64.b64encode(data).decode()

img_base64 = get_base64_img("desktop.png")
safe_img_base64 = get_base64_img("safe.png")
doc_img_base64 = get_base64_img("doc.png")
# Icons
col1, col2, col3 = st.columns(3)

with col1:
    st.markdown(
        f"""
        <a href="/MyReports" target="_self">
            <img src="data:image/png;base64,{img_base64}" width="60">
        </a>
        <p style='text-align:justify; color:#007c74; font-weight:600;'>View Reports Online</p>
        """,
        unsafe_allow_html=True
    )

with col2:
    st.markdown(
        f"""
        <a href="/ContactUs" target="_self">
            <img src="data:image/png;base64,{safe_img_base64}" width="60">
        </a>
        <p style='text-align:justify; color:#007c74; font-weight:600;'>Contact Us</p>
        """,
        unsafe_allow_html=True
    )

with col3:
```

fig 10.1.2 Landing Page Code

```python
# pages/03HeartModel.py
import streamlit as st
import numpy as np
from utils.predictions import load_model, predict_score
from utils.preprocessing import load_scaler_heart
from pages.generate_report import generate_report
#from streamlit_extras.stx_cookie_manager.stx_cookie_manager import CookieManager


# Ensure user is logged in
if 'user' not in st.session_state:
    st.warning("Please log in to view this page.")
    st.stop()
user_email = st.session_state["user"]["email"]

user_email = st.session_state["user"]["email"]
def run_heart_disease_prediction():
    if 'user' not in st.session_state:
        st.warning("Please log in to view this page.")
        st.stop()
# Setup session state for navigation
if 'heart_form_page' not in st.session_state:
    st.session_state.heart_form_page = "non_tech"

st.title("Heart Disease Prediction")

if st.session_state.heart_form_page == "non_tech":
    st.header("Enter Your Non-Technical Parameters")

    age = st.number_input("Age (20 - 120)", min_value=20, max_value=120, key="age")
    gender_options = {0: "Male", 1: "Female"}
    gender = st.selectbox("Gender", options=list(gender_options.keys()), format_func=lambda x: gender_options[x], key="gender")

    weight = st.number_input("Weight (kg)", min_value=20.0, max_value=300.0, key="weight")
    height = st.number_input("Height (cm)", min_value=100.0, max_value=250.0, key="height")
    bmi = round(weight / ((height / 100) ** 2), 1)
    st.session_state.bmi = bmi
    st.write(f"**Calculated BMI**: {bmi}")

    smoking_options = {0: "Never", 1: "Occasional", 2: "Regular", 3: "Heavy", 4: "Chain-Smoker"}
    alcohol_options = {0: "None", 1: "Social", 2: "Moderate", 3: "Frequent", 4: "Alcoholic"}
    physical_activity_options = {1: "Low", 2: "Medium", 3: "High"}
    diet_options = {0: "High Sodium", 1: "Low Potassium", 2: "Balanced", 3: "Other"}
```

fig 10.1.3 Heart Model Code

```python
# pages/04HypertensionModel.py
import streamlit as st
import numpy as np
from utils.predictions import load_model, predict_score
from utils.preprocessing import load_scaler_hypertension
from pages.generate_report import generate_report

# Ensure user is logged in
if 'user' not in st.session_state:
    st.warning("Please log in to view this page.")
    st.stop()

user_email = st.session_state["user"]["email"]

def run_hypertension_risk_prediction():
    st.write("Running the Hypertension Risk Prediction model...")
    # Your prediction model code goes here
    st.write("Prediction: Hypertension Risk Level")
# Setup session state for navigation
if 'hypertension_form_page' not in st.session_state:
    st.session_state.hypertension_form_page = "non_tech"

st.title("Hypertension Risk Prediction")

if st.session_state.hypertension_form_page == "non_tech":
    st.header("Enter Your Non-Technical Parameters")

    age = st.number_input("Age (20 - 120)", min_value=20, max_value=120, key="age")
    gender_options = {0: "Male", 1: "Female"}
    gender = st.selectbox("Gender", options=list(gender_options.keys()), format_func=lambda x: gender_options[x], key="gender")

    weight = st.number_input("Weight (kg)", min_value=20.0, max_value=300.0, key="weight")
    height = st.number_input("Height (cm)", min_value=100.0, max_value=250.0, key="height")
    bmi = round(weight / ((height / 100) ** 2), 1)
    st.session_state.bmi = bmi
    st.write(f"**Calculated BMI**: {bmi}")

    smoking_options = {0: "Never", 1: "Occasional", 2: "Regular", 3: "Heavy", 4: "Chain-Smoker"}
    alcohol_options = {0: "None", 1: "Social", 2: "Moderate", 3: "Frequent", 4: "Alcoholic"}
    stress_options = {0: "Low", 1: "Medium", 2: "High", 3: "Very High"}
    work_env_options = {0: "Active", 1: "Sedentary"}

    st.selectbox("Smoking Status", list(smoking_options.keys()), format_func=lambda x: smoking_options[x], key="smoking")
```

fig 10.1.4 Hypertension Model Code

```python
# pages/05ObesityModel.py
import streamlit as st
import numpy as np
from utils.predictions import load_model, predict_score
from utils.preprocessing import load_scaler_obesity
from pages.generate_report import generate_report

# Ensure user is logged in
if 'user' not in st.session_state:
    st.warning("Please log in to view this page.")
    st.stop()

user_email = st.session_state["user"]["email"]
def run_obesity_risk_prediction():
    st.write("Running the Obesity Risk Prediction model...")
    # Your prediction model code goes here
    st.write("Prediction: Obesity Risk Level")
# Setup session state for navigation
if 'obesity_form_page' not in st.session_state:
    st.session_state.obesity_form_page = "non_tech"

st.title("Obesity Risk Prediction")

if st.session_state.obesity_form_page == "non_tech":
    st.header("Enter Your Non-Technical Parameters")

    age = st.number_input("Age (20 - 120)", min_value=20, max_value=120, key="age")
    gender_options = {0: "Male", 1: "Female"}
    gender = st.selectbox("Gender", options=list(gender_options.keys()), format_func=lambda x: gender_options[x], key="gender")

    weight = st.number_input("Weight (kg)", min_value=20.0, max_value=300.0, key="weight")
    height = st.number_input("Height (cm)", min_value=100.0, max_value=250.0, key="height")
    bmi = round(weight / ((height / 100) ** 2), 1)
    st.session_state.bmi = bmi
    st.write(f"**Calculated BMI**: {bmi}")

    smoking_options = {0: "Never", 1: "Occasional", 2: "Regular", 3: "Heavy", 4: "Chain-Smoker"}
    alcohol_options = {0: "None", 1: "Social", 2: "Moderate", 3: "Frequent", 4: "Alcoholic"}
    stress_options = {0: "Low", 1: "Medium", 2: "High", 3: "Very High"}
    work_env_options = {0: "Active", 1: "Sedentary"}

    st.selectbox("Smoking Status", list(smoking_options.keys()), format_func=lambda x: smoking_options[x], key="smoking")
    st.selectbox("Alcohol Consumption", list(alcohol_options.keys()), format_func=lambda x: alcohol_options[x], key="alcohol")
    st.selectbox("Stress Levels", list(stress_options.keys()), format_func=lambda x: stress_options[x], key="stress_levels")
```

fig 10.1.5 Obesity Model Code

```python
import streamlit as st
import os

# Reports folder path
REPORTS_FOLDER = r"C:\Users\Hp\Desktop\MP_authentication_7thMay2\reports"
os.makedirs(REPORTS_FOLDER, exist_ok=True)  # Ensure folder exists
if 'user' not in st.session_state:
    st.warning("Please log in to view this page.")
    st.stop()
def get_report_files():
    """Retrieve all stored PDF report files."""
    return [f for f in os.listdir(REPORTS_FOLDER) if f.endswith(".pdf")]

def my_reports_page():
    """Displays stored reports and provides download options."""
    st.title("  My Reports")

    reports = get_report_files()

    if not reports:
        st.info("No reports available. Run a model to generate a report.")
        return

    st.write("### Your Generated Reports:")

    for report in reports:
        report_path = os.path.join(REPORTS_FOLDER, report)

        # Extract username and model name from the file
        report_name = report.replace("_", " ").replace(".pdf", "")

        col1, col2, col3 = st.columns([3, 1, 1])  # Adjust sizes for alignment

        with col1:
            st.write(f"  **{report_name}**")

        with col2:
            with open(report_path, "rb") as file:
                report_bytes = file.read()
                st.download_button(
                    label="  Download",
                    data=report_bytes,
                    file_name=report,
```

fig 10.1.6 Report Generation Code

```python
import streamlit as st
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from datetime import datetime

# Set light blue background color
st.markdown(
    """
    <style>
    body {
        background-color: #a8e5d2;
    }
    .stApp {
        background-color: #a8e5d2;
    }
    </style>
    """,
    unsafe_allow_html=True
)

st.title("📅 Book an Appointment")

# Input fields
name = st.text_input("Name", placeholder="Enter your full name")
phone = st.text_input("Phone Number", placeholder="Enter your phone number")
appointment_date = st.date_input("Date of Appointment")
appointment_time = st.time_input("Time of Appointment")

# Doctor selection dropdown
doctors = {
    "Dr. A. Mehta - Cardiologist",
    "Dr. B. Sharma - Endocrinologist",
    "Dr. C. Reddy - General Physician",
    "Dr. D. Iyer - Neurologist",
    "Dr. E. Singh - Orthopedic Surgeon",
    "Dr. F. Gupta - Pulmonologist",
    "Dr. G. Desai - Psychiatrist",
    "Dr. H. Khan - Dermatologist"
}
selected_doctor = st.selectbox("Select Doctor", sorted(doctors))
```

fig 10.1.7 Book Appointment Code

GitHub Repository Link: https://github.com/eng21cs0045/MedScore-Assist