VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A PROJECT REPORT

on

"SMART HEALTH ADVISOR: RISK ASSESSMENT AND LIFESTYLE GUIDANCE FOR CHRONIC DISEASE PREVENTION"

submitted by

Sanath	4SF22IS087
Kumar	4SF22IS041
L Ajithesh	4SF22IS042
Pranav M Naik	4SF22IS070

In partial fulfillment of the requirements for the award of

of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

under the guidance of

Mrs. Madhu R

Assistant Professor, Department of ISE

at

SAHYADRI

College of Engineering & Management

An Autonomous Institution

Adyar, Mangaluru - 575 007

AY: 2025 - 26

SAHYADRI

College of Engineering & Management

An Autonomous Institution

Adyar, Mangaluru - 575 007

Department of Information Science & Engineering

CERTIFICATE

This is to certify that the project entitled **"Smart Health Advisor: Risk Assessment and Lifestyle Guidance for Chronic Disease Prevention"** has been carried out by **Sanath (4SF22IS087), Kumar(4SF22IS041), L Ajithesh(4SF22IS042) and Pranav M Naik (4SF22IS070)**, the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment of the requirements for the award of Bachelor of Engineering in Information Science & Engineering (ISE) of Visvesvaraya Technological University, Belagavi during the Academic Year 2025 - 26. It is certified that all the corrections/suggestions indicated for the Continuous Internal Assessment have been incorporated in the report deposited in the library of the ISE department. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Project Guide	HoD	Principal
Mrs. Madhu R	**Dr. Rithesh Pakkala P.**	**Dr. S S Injaganeri**
Assistant Professor	Associate Professor	Professor
Dept. of ISE, SCEM	Dept.of ISE & CSE(DS), SCEM	SCEM, Mangaluru

External Viva-Voce

Examiner's Name	**Signature with Date**
1.	....
2.	....

SAHYADRI

College of Engineering & Management

An Autonomous Institution

Adyar, Mangaluru - 575 007

Department of Information Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this Project Report titled **"Smart Health Advisor: Risk Assessment and Lifestyle Guidance for Chronic Disease Prevention"** has been carried out by us at Sahyadri College of Engineering & Management, Mangaluru, under the supervision of **Mrs. Madhu R,** for the award of **Bachelor of Engineering** in **Information Science & Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Sanath (4SF22IS087)

Kumar (4SF22IS041)

L Aijthesh (4SF22IS042)

Pranav M Naik (4SF22IS070)

Dept. of ISE, SCEM, Mangaluru

Abstract

The growing incidence of chronic illnesses such as diabetes, hypertension, and obesity has increased the demand for tools that support prevention and personalized health management. The Smart Health Advisor system aims to meet this need by offering AI-based risk assessment and lifestyle guidance for both healthy users and individuals already living with these conditions. The platform evaluates demographic information along with lifestyle, nutritional, and behavioral inputs to estimate a user's likelihood of developing obesity, hypertension, or diabetes and provides appropriate preventive suggestions. For users who have already been diagnosed, the system generates individualized dietary and habit adjustment recommendations that align with their health requirements. Key indicators including BMI, activity levels, sleep patterns, and dietary habits—are processed through trained machine learning models to produce reliable risk predictions. Alongside prediction, the platform incorporates a rule-driven chatbot that offers simple and relevant advice based on the user's predicted condition. The chatbot relies on predefined rules and keyword matching to deliver guidance related to diet, exercise, and preventive practices, ensuring that users receive consistent and actionable information. All health data is securely stored in an AES-256 encrypted database, strengthening user privacy while supporting safe access to past records. By combining predictive analytics, personalized recommendations, and strong security features, the Smart Health Advisor serves as a comprehensive digital health tool that helps users manage their well-being and reduce long-term risk of chronic illnesses.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the project report on **"Smart Health Advisor: Risk Assessment and Lifestyle Guidance for Chronic Disease Prevention"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Information Science & Engineering.

We are profoundly indebted to our guide, **Mrs. Madhu R**, Assistant Professor, Department of Information Science & Engineering, for innumerable acts of timely advice and encouragement, and we sincerely express our gratitude.

We also thank **Dr. Vasudeva Rao P V**, Assistant Professor & Project Coordinator, Department of Information Science & Engineering, for the constant encouragement and support extended throughout.

We express our sincere gratitude to **Dr. Rithesh Pakkala P.**, Associate Professor & Head, Department of Information Science & Engineering and Computer Science & Engineering (Data Science) for his invaluable support and guidance.

We sincerely thank **Dr. Sudheer Shetty**, Professor, ISE & Vice Principal, and **Dr. Shamantha Rai B**, Professor, ISE & Dean Academics, Sahyadri College of Engineering & Management, who has constantly motivated us throughout the project work.

We sincerely thank **Dr. S S Injaganeri**, Principal, Sahyadri College of Engineering & Management, who have always been a great source of inspiration.

Finally, yet importantly, We express our heartfelt thanks to the nonteaching staff of the ISE department and our family & friends for their wishes and encouragement throughout the work.

Sanath (4SF22IS087)
Kumar (4SF22IS041)
L Aijthesh (4SF22IS042)
Pranav M Naik (4SF22IS070)

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Lifestyle related conditions such as diabetes, hypertension, and obesity are becoming increasingly common among young and middle-aged adults, largely due to long sitting hours, poor eating habits, high stress levels, irregular sleep, and insufficient physical activity. These factors have accelerated the onset of chronic diseases in age groups that were traditionally considered low-risk. As healthcare systems typically emphasize treatment after symptoms appear, there is a clear need for digital solutions that help individuals monitor their health, identify risks early, and adopt preventive lifestyle adjustments.

The Smart Health Advisor project addresses this requirement by integrating machine learning, rule-based decision logic, and secure data management into a single interactive platform. Users provide essential health and lifestyle information, which is processed through a trained machine learning pipeline to estimate the likelihood of developing diabetes, hypertension, or obesity. Preprocessing methods such as feature scaling and data normalization, combined with a Random Forest classifier, enhance the system's predictive accuracy and reliability.

To support behavioral change, the platform includes a rule-based chatbot that delivers personalized recommendations aligned with the predicted condition. These responses offer practical guidance on nutrition, exercise routines, sleep improvement, stress reduction, and overall wellness. By presenting information in a simple and actionable format, the chatbot encourages users to gradually adopt healthier habits.

Protecting user data is a central priority of the system. All submitted inputs and prediction outcomes are encrypted using AES-256-GCM before storage, ensuring confidentiality and safeguarding against unauthorized access. Secure authentication mechanisms further support the privacy and integrity of user records.

Developed as a Flask web application, the Smart Health Advisor provides an accessible and user-friendly interface that combines predictive analytics, personalized feedback, and strong data security. The platform aims to empower individuals by offering early health insights, promoting awareness of lifestyle choices, and enabling a proactive approach to long-term wellness.

1.1 Purpose

The Smart Health Advisor project is designed to tackle the growing incidence of lifestyle-related conditions—such as diabetes, hypertension, and obesity—among individuals aged 17 to 45. Although this age range is typically associated with peak productivity, many people are now experiencing chronic health issues due to factors like unhealthy eating patterns, limited physical activity, elevated stress, irregular sleep cycles, substance use, and delays in seeking medical care.

Traditional healthcare systems often emphasize treatment after a disease has already developed, which can lead to higher medical costs, later intervention, and declining overall well-being. To counter this, the Smart Health Advisor promotes a more proactive model of care by using machine learning and structured lifestyle data to predict the likelihood of key chronic diseases and provide early, meaningful insights into a user's health.

The project's purpose is to help individuals manage their health more effectively by offering early risk identification, ongoing monitoring, and personalized recommendations. Acting as a digital health assistant, the platform encourages healthier habits over time. Its rule-based chatbot delivers clear, practical guidance on nutrition, exercise, sleep, and general wellness that aligns with the user's predicted risk.

Data protection is also central to the project's mission. All user information—including inputs and prediction results is secured with AES-256 encryption to ensure privacy and maintain trust. By combining machine-learning analysis, personalized lifestyle support, and strong data-security practices, the Smart Health Advisor aims to foster long-term health awareness and help reduce the future impact of chronic diseases through an intelligent, user-friendly, and security-focused digital solution.

1.2 Scope

The scope of the Smart Health Advisor project focuses on creating a digital health platform that uses machine-learning techniques to help identify the risk of diabetes, hypertension, and obesity at an early stage. The system is intended for individuals between 17 and 45 years of age and encourages them to better understand their personal health by examining their lifestyle choices, dietary habits, and physical activity patterns. By generating risk predictions and offering tailored guidance, the platform promotes preventive awareness and informed decision-making.

To support this goal, the system analyzes important indicators such as BMI, sleep quality and duration, exercise levels, eating behavior, and basic medical measurements. These inputs help users see how their daily habits may influence their likelihood of developing chronic conditions. A key feature of the platform is its rule-based chatbot, which interacts with users, responds to common health questions, and provides straightforward recommendations aligned with the predicted risk category. Instead of relying on adaptive or learning-based AI models, the chatbot operates using predefined rules and keyword recognition to deliver clear suggestions ranging from diet improvements to physical activity tips and general wellness practices.

Because the platform manages sensitive personal information, data protection is a critical part of the project scope. All user inputs and prediction outcomes are secured using AES-256-GCM encryption prior to storage, ensuring confidential handling of health records and maintaining user trust.

Beyond supporting individual users, the system also contributes to encouraging long-term preventive habits and spreading awareness about lifestyle-related health risks. By integrating prediction models, rule-based guidance, and encrypted data management into a single interface, the Smart Health Advisor aims to provide a secure, accessible, and proactive solution for everyday health management.

1.3 Overview

The Smart Health Advisor is a digital wellness platform built around machine-learning techniques to help users better understand their risk of developing diabetes, hypertension, and obesity. The system evaluates user-provided information—such as BMI, eating habits, activity patterns, sleep duration, and basic medical details—to generate early insights into potential health concerns and support preventive decision-making.

A key feature of the platform is its rule-driven chatbot, which makes the system more interactive and user-centric. Instead of learning dynamically, the chatbot functions through predefined rules and keyword recognition to provide tailored health suggestions.

Its responses cover areas such as nutrition, exercise planning, stress reduction, hydration, and daily wellness practices, offering practical steps that users can incorporate into their routines.

Data protection is an essential part of the system's design. Because the platform manages sensitive health information, it uses AES-256-GCM encryption to secure all stored inputs and prediction results. This approach safeguards the confidentiality of user data, prevents unauthorized access, and reinforces trust in the platform as a safe environment for health tracking.

Beyond supporting individual users, the system contributes to larger preventive-health goals by raising awareness of lifestyle-related risks and encouraging consistent self-care habits. Through its combination of predictive modeling, rule-based guidance, and encrypted data storage, the Smart Health Advisor provides a dependable and accessible digital tool for long-term wellness management—promoting a proactive approach rather than relying on traditional, treatment-focused healthcare models.

Chapter 2

Literature Survey

The prediction model is evaluated using standard performance metrics like accuracy, sensitivity, specificity, and precision. Among the classifiers, Random Forest achieves slightly higher accuracy, making it a preferable model in clinical prediction environments. The study also includes a basic diet recommendation system that suggests predefined nutritional plans based on the prediction results. However, the recommendation system is rule-based and not dynamic, which limits its effectiveness in personalized healthcare scenarios.

A. S. Maria et al. [1] conduct a comparative study of several machine-learning techniques aimed at predicting obesity risk using a mix of demographic, behavioral, and lifestyle indicators. Their analysis covers a range of supervised models, including SVM, KNN, Decision Trees, Random Forest, and Gradient Boosting. The dataset used in the study contains variables such as age, gender, eating habits, physical activity levels, sleep duration, and screen-time patterns—factors that play a substantial role in influencing obesity outcomes. Prior to model training, the authors apply a comprehensive preprocessing workflow involving treatment of missing data, normalization of numerical features, encoding of categorical attributes, and selection of relevant variables.

Model performance is measured with standard metrics—accuracy, precision, recall, and F1-score. According to the findings, ensemble approaches, particularly Gradient Boosting and Random Forest, deliver stronger results compared to standalone classifiers. Their advantage stems from the ability to model complex nonlinear patterns, reduce prediction variability, and adapt effectively to heterogeneous population groups. The authors conclude that with well-curated input features, especially those capturing behavioral tendencies, ensemble-based models can provide reliable and insightful predictions in obesity-risk assessment tasks.

K. Kahalkar and U. Vyas [2] This paper presents a vision for integrating artificial intelligence (AI) into the domains of personalized nutrition and healthcare, leveraging genomic data, microbiome analysis, and bioinformatics. The authors emphasize the use of precision medicine and AI-driven analytics to develop dietary and healthcare plans tailored to each individual's genetic and biological profile. Ethical concerns and data privacy, especially in handling sensitive genomic and health information, are also discussed as vital components of system design.

The system framework outlined involves capturing patient data both clinical and lifestyle-related—and processing it through AI models to generate recommendations. The study also highlights the potential of integrating decision medicine, where AI can aid physicians in selecting suitable treatments or nutritional interventions based on biomarkers and genetic predispositions. Keywords such as precision medicine, bioinformatics, and AI-driven healthcare dominate the discussion, pointing to a technology-centric approach.

A. Kumar et al. [3] This research explores obesity prediction using a range of supervised machine learning algorithms, including Logistic Regression, Random Forest, and Decision Trees. The dataset utilized in this study comprises demographic, behavioral, and lifestyle attributes such as age, gender, smoking habits, physical activity, alcohol consumption, and dietary patterns. The primary objective is to assess the contribution of these features to obesity prediction and compare the performance of different models. Feature selection methods are employed to isolate variables with the highest correlation to obesity outcomes. Logistic Regression, while less complex, offers transparency in prediction, allowing for interpretable outputs. Random Forest and Decision Trees, on the other hand, deliver higher prediction accuracy, particularly when capturing non-linear relationships between lifestyle behaviors and body weight categories. The study includes a thorough performance analysis using metrics such as accuracy, recall, precision, and confusion matrix evaluation. The results indicate that while all models perform reasonably well, ensemble methods yield better results for practical deployment. However, the system proposed is designed solely for prediction, lacking any mechanism to provide feedback or recommendations post-diagnosis.

K. S. Kulkarni et al. [4] This study introduces a recommender system for personalized diet and exercise plans using similarity modeling techniques such as cosine similarity and Pearson correlation. The goal is to suggest lifestyle improvements based on users with similar health profiles and behavior patterns. The system matches a user's health data—like BMI, age, activity level, and dietary preferences—with others in the dataset and generates personalized wellness plans based on correlations observed. The SmartHealth model focuses on user clustering and collaborative filtering, drawing on shared attributes and historical success rates to make recommendations. This approach allows users to receive practical, real-world suggestions backed by data-driven logic. It emphasizes health informatics and wellness optimization over clinical diagnosis or treatment. Although the recommendation engine performs well in providing tailored guidance, it does not include a disease prediction module. The system is primarily focused on fitness and wellness and does not offer medical insights or risk scores for chronic conditions. Moreover, there is no chatbot interface for real-time interaction, nor does the system provide a feedback loop for evaluating user adherence and adjusting recommendations. Importantly, there is also no mention of secure data handling practices—something critical for platforms processing sensitive health data.

N. Tripathy et al. [5] This paper provides a comparative analysis of traditional machine learning and deep learning models for the prediction of diabetes using real-world datasets. The authors explore a wide array of classifiers, including Support Vector Machines (SVM), Logistic Regression, Random Forest, Artificial Neural Networks (ANN), and Deep Neural Networks (DNN). The dataset used includes demographic and medical variables such as blood pressure, glucose levels, age, insulin, and skin thickness. The models are rigorously evaluated using key performance indicators like accuracy, precision, recall, and F1-score. The results demonstrate that deep learning models, especially DNNs, tend to outperform classical ML algorithms in terms of accuracy, although they require more training time and computational resources. This makes DL models suitable for larger-scale deployments where prediction precision is critical.

S. S. Bhat et al. [6] This study introduces an approach that uses machine-learning methods to estimate an individual's risk of developing diabetes and to offer tailored dietary guidance for those identified as at risk. The model incorporates a broad range of variables, including demographic information, clinical indicators, lifestyle behaviors, and nutrition patterns. Before training, the dataset is refined through normalization, handling of missing entries, and selection of informative features to improve the reliability of the models.

Several classification techniques—such as logistic models, support-vector methods, ensemble-based learners, and probabilistic classifiers—were assessed to determine which provided the strongest predictive performance. The models were compared using common evaluation measures, including precision, recall, accuracy, and the F1-score. In these experiments, ensemble approaches and SVM-based models generally achieved superior results, particularly when dealing with complex, non-linear relationships in the clinical data.

Beyond prediction, the framework includes a module that recommends dietary patterns aligned with an individual's assessed risk level. The recommendations emphasize appropriate nutrient distribution, moderation of sugar intake, caloric balance, and adherence to dietary restrictions. The findings suggest that combining risk assessment with personalized nutritional advice can assist in supporting early intervention, encouraging healthier habits, and promoting preventive care for individuals concerned about diabetes.

K. Jain and S. Gupta [7] This study introduces a text-based AI chatbot—Health-Bot—designed to simulate medical consultation by answering health-related queries using natural language processing (NLP), knowledge graphs, and decision trees. The system aims to support users with information about symptoms, treatments, and general healthcare advice in a conversational format. It leverages probability analysis to infer appropriate responses based on user input and context derived from preloaded medical data. The chatbot is structured around a decision-tree model augmented with a knowledge graph, which helps in mapping medical terminology and relationships between symptoms, diseases, and treatments. The system offers fast responses and supports frequently asked medical queries, which could reduce the burden on healthcare professionals and improve health literacy among users. Despite its usefulness as an interactive tool, the paper lacks integration with predictive analytics. It does not support disease risk prediction or personalized health tracking based on user input. The model is limited to QA functionality and lacks dynamic adaptation or follow-up coaching features. Moreover, the paper does not mention any data encryption or access control mechanisms to protect sensitive user data, which is a serious concern in healthcare deployments.

H. S. Dharmarajan et al. [8] This paper explores the use of artificial intelligence for dynamic nutritional planning and diet management. The authors argue that traditional diet plans are static and often fail to address individual needs or adapt to real-time lifestyle changes. To address this, the system proposed in the study uses machine learning to analyze user data and adjust dietary recommendations based on ongoing behavior, nutritional deficiencies, and health goals.

The system integrates user-specific data including daily activity, food intake, health goals, and demographic information. ML algorithms are used to predict dietary needs and deficiencies and to recommend food items that align with users' health objectives. Unlike static diet charts, this AI-based system continuously learns and adapts as new user data is collected, creating a personalized and evolving nutrition plan. The model demonstrates strong applicability for personalized health apps and wellness platforms. However, it lacks interaction capability—there is no mention of a chatbot or conversational interface to guide the user through their health journey. Moreover, while the paper discusses the technical aspects of prediction and personalization, it does not address data security or privacy. This is a major shortfall, as personal dietary and health data are highly sensitive and subject to data protection laws.

A. A. Jikar and S. Mendhe [9] This paper presents a health advisory chatbot that integrates machine learning algorithms with a conversational interface to assist users in receiving health-related advice. The chatbot is designed to interact with patients, identify symptoms, and provide preliminary health guidance based on the underlying classification models. The algorithms implemented include Support Vector Machine (SVM), Decision Tree, and Random Forest classifiers, which are used to analyze user input and suggest potential health concerns or lifestyle adjustments.
The system is primarily rule-driven, with machine learning employed for behind-the-scenes classification based on structured symptom and health data. It performs well in matching user inputs to probable health issues and is capable of providing instant responses through its text-based interface. This model is aimed at providing immediate informational support to users and reducing dependency on initial clinical consultations for minor or repeat concerns.

A. P. P. et al. [10] This study introduces a supervised machine learning–based approach for building a personalized diet recommendation system. The authors employ algorithms such as K-Nearest Neighbors (KNN), Random Forest, and Naïve Bayes to predict suitable dietary plans based on a variety of user inputs including age, BMI, daily activity level, and nutritional needs. The system is trained on structured health and food intake datasets and aims to assist users in making healthier dietary decisions. The core contribution of the paper lies in its comparative analysis of classifiers to identify the best-performing model in predicting diet recommendations. Random Forest yielded the most accurate results due to its ability to handle non-linear relationships and avoid overfitting. KNN provided acceptable results, especially in classifying food preferences and nutritional categories. The system is intended to be used via a mobile interface to promote easy access and practical use among health-conscious users.

F. Buluş et al. [11] This paper evaluates multiple machine learning algorithms for predicting chronic diseases using healthcare datasets. The study focuses on the early identification of conditions such as diabetes, heart disease, and hypertension. The algorithms employed include Decision Trees, Naïve Bayes, and other supervised learning models, which are applied to datasets with health attributes like age, BMI, glucose level, cholesterol, and exercise patterns.

The study compares models based on their precision, recall, accuracy, and robustness in predicting disease onset. Decision Trees are highlighted for their interpretability and ability to work well with both numerical and categorical data. Meanwhile, Bayesian models are praised for their speed and suitability for probabilistic health modeling. The paper successfully demonstrates the feasibility of ML in chronic disease prediction and underlines the importance of early intervention and proactive healthcare management.

H. L. C. Liyanarachchi et al. [12]This paper proposes a secure, cloud-based chronic disease prediction system using machine learning algorithms and encrypted data handling. The authors focus on predicting heart disease, diabetes, and other chronic conditions while addressing the privacy challenges associated with storing and processing personal health data. Logistic Regression is the primary algorithm used, selected for its simplicity, transparency, and consistent performance in clinical datasets. A distinguishing feature of this work is the incorporation of cloud security measures. The paper details user access control, data anonymization, and multi-layer authentication mechanisms as part of its health management system. This sets it apart from other works that typically focus solely on prediction. The system is designed to operate within a secure environment, ensuring that only authorized users can view or interact with sensitive medical information.

S. D. Gharge et al. [13] This paper presents the development of a healthcare chatbot application that uses artificial intelligence and natural language processing (NLP) to provide basic medical assistance and telemedicine support. The chatbot aims to reduce the workload on healthcare professionals by addressing routine queries, scheduling appointments, and offering general health advice. Built using rule-based NLP and basic classification models, the chatbot system is capable of engaging in question-answer-style interaction with patients, especially those in remote or underserved areas.

The authors highlight the importance of improving patient engagement through accessible, 24/7 AI-driven communication. The system is designed to be platform-independent, offering deployment on web and mobile interfaces. It is especially useful in environments where healthcare resources are limited or users require immediate attention but cannot access doctors in real time. The chatbot can interpret common health-related sentences and redirect users to emergency services if necessary.

S. Singh et al. [14] This research investigates the application of machine learning techniques for predicting various chronic diseases, including diabetes, heart disease, and dementia. The authors aim to provide a comprehensive comparative analysis of widely-used ML algorithms such as Decision Tree, Random Forest, Gradient Boosting, and XGBoost. The dataset incorporates clinical and lifestyle-based attributes like age, blood pressure, glucose level, BMI, and family history to determine the likelihood of developing a chronic illness.

Each model is evaluated using standard metrics such as accuracy, precision, recall, and execution time. Gradient Boosting and XGBoost are identified as the most effective models due to their superior performance in both predictive accuracy and computational efficiency. The paper's major contribution lies in presenting a multi-disease prediction framework and highlighting how ensemble models outperform simple classifiers in complex health data environments.

A. P. P. et al. [15] This paper presents an AI-powered system designed to deliver personalized dietary recommendations using machine learning, collaborative filtering, and NLP. The system profiles users based on preferences, allergies, activity levels, and dietary goals, and recommends food items that align with their health targets. The approach combines user input with AI models to adapt recommendations in real time, creating a dynamic and responsive nutrition advisor. The architecture includes supervised ML models to categorize food items, collaborative filtering to suggest meals based on user similarity, and NLP to interpret natural language inputs such as meal descriptions or user feedback. The system is designed for mobile deployment and focuses on maintaining a user-friendly interface. It supports goal tracking and provides meal planning based on caloric and macronutrient requirements.

Chapter 3

Problem Statement

3.1 Problem Statement

Chronic illnesses associated with lifestyle are becoming very common among people between age 17 and 45. Regardless of technological innovations in health care, a number of people are oblivious to the health risks of the condition until it is in its advanced stages. The solutions that are available are mainly based on post diagnosis treatment and do not provide much support on lifestyle correction or early diagnosis. They are usually deprived of personalized information, data protection, or real-time communication with the user.

Also there is fragmentation of the digital health ecosystem. There are those that offer simple health monitoring and those that might offer inert dietary recommendations. Nevertheless, not many systems combine the predictive analytics, secure health data processing, and user interaction instruments into one unified platform. This causes a low level of user adherence, non-actionable advice, and a lack of data security.

The other issue of concern that is missing is the lack of a system that will address preventive as well as therapeutic healthcare demands. Normal people with no particular health issues do not attend to prevention care, which is why they receive less special attention, and patients with long-term health problems tend to have problems with general and non-specific care prescriptions. In addition, there is also no focus on the utilization of encryption solutions, such as AES-256, to protect the medical information stored and shared in such systems.

The project will mitigate these concerns by introducing a smart assistant that will provide proactive risk evaluation to healthy users and customized health plan to patients coupled with secure storage and interactive chatbot to ensure a smooth interaction.

3.2 Objectives

- To develop a machine learning-based model to assess the risk of chronic diseases among users aged 17–45 based on lifestyle, dietary, and substance use data.
- To implement an AI recommendation engine that suggests corrective lifestyle and dietary measures for both at-risk individuals and chronic disease patients.
- To build an AES-256 encrypted storage system for sensitive health records and integrate an AI chatbot assistant for user interaction, ensuring data privacy and accessibility.

Chapter 4

Software Requirements Specification

4.1 Introduction

The Smart Health Advisor is an AI-based multi-level web application aimed at the promotion of preventive care and customized wellness management. It incorporates machine learning (ML), data analytics, and conversation-like chatbot interface to deliver the user real-time health insights, lifestyle recommendations, and behavioral support. The system allows users to record and track their health-related parameters including diet, sleep, physical activity, and stress levels and obtain adaptive advice depending on their habits and progress.

The system has a software architecture that is modular, and scalable with a few major components operating in harmony with each other. The backend is coded in Python and Flask web framework that handles the application logic and communications between the frontend and the database. The backend incorporates a machine learning engine that processes user data and makes predictions about possible health risks as well as provides personalized recommendations. The frontend is programmed in HTML, CSS, JavaScript, and Bootstrap that provides a responsive and user-friendly interface that can be opened using web browsers on several devices. The system is safe with the data stored in the SQLite database that contains user profiles, the history of activities, and the history of interaction with the chatbot.

The system works on a specific hardware and software environment to comply with seamless performance, information privacy, and reliability. This section defines these components and their interactions, and this is used to guide the system setup, development, testing, and deployment.

4.2 Purpose

This section aims at defining the software and hardware environment that would be used to develop, test and implement the Smart Health Advisor system. It provides the technical background to the realization of the system goals of customized health tracking, preventive therapeutic care, and adaptive user interface.

Particularly, this section determines the programming languages, frameworks, databases, libraries, and tools that are necessary in the operation of the system. It also outlines the Hardware requirements and platform requirements that may be required to provide an optimal performance and compatibility across various devices. The described environment helps developers to develop and maintain the system effectively, provides testers with the opportunity to check the system performance and reliability, and gives deployment recommendations to be used on the stable functioning during the work in a real-life environment.

These parameters will be defined in this section to make sure that all the steps of the project, such as development and interaction with the users, will be backed by a stable and steady technical infrastructure. This eventually ensures that the Smart Health Advisor operates as it should: safely, efficiently and effectively helping users to take care of their health using smart information-driven methods.

4.3 User Characteristics

The Smart Health Advisor system is designed primarily for end users, who are individuals seeking to monitor and improve their overall health and well-being. These users are typically adults aged between 18 and 40 years, a group increasingly affected by lifestyle-related health issues such as stress, obesity, hypertension, and poor sleep habits. They are generally comfortable using technology and prefer digital tools that are simple, engaging, and easy to navigate.

The users are expected to have basic computer or smartphone literacy, enabling them to interact with the platform through its web-based or mobile interface. Their main activities on the system will include logging daily health information, tracking habits such as diet, exercise, and sleep, and receiving personalized health insights and recommendations generated by the AI.

Most users will be health-conscious individuals looking for preventive care solutions, though some may be new to regular health tracking and require an intuitive design and clear guidance to stay motivated. The system is therefore built to support varying levels of technical and health knowledge, providing an accessible and user-friendly experience for all.

Since this project focuses solely on user interaction, there are no distinct developer or system administrator roles involved in regular system use. After deployment, the platform will be fully operated by end users, with all backend maintenance managed internally by the project team. The ultimate goal is to empower users to take control of their health through continuous engagement, personalized support, and simple, technology-driven preventive care.

System Administrators: Who will be responsible for deploying and maintaining the server hardware, database, and backend services.

4.4 Interfaces

4.4.1 Hardware Interfaces

Table 4.1: Hardware Interface Requirements

Component	Specification	Purpose
Processor (CPU)	Intel i5 9th Generation or equivalent AMD processor, 2.5 GHz+	Handles computational load of Flask web server, ML model inference, and simultaneous database operations.
Memory (RAM)	8 GB DDR4 (16 GB or higher preferred)	Supports concurrent execution of IDE, browser tabs, local server, and in-memory data processing during model prediction.
Storage	100 GB free space on SSD	Provides high-speed storage for OS, Anaconda, project files, datasets, SQLite database, and logs.
Operating System	Windows 10 or higher	Ensures compatibility with Anaconda distribution and Python development environment.
Internet Connectivity	10 Mbps or higher	Required for package installations, version control operations, and potential cloud deployment.

The Smart Health Advisor system depends on suitable hardware resources to ensure efficient backend operations, smooth machine-learning inference, and responsive interaction through the web interface. To support these tasks, the system recommends a processor in the range of an Intel Core i5 (9th generation) or an equivalent AMD CPU running at 2.5 GHz or higher. This level of processing power allows the machine to manage simultaneous tasks such as operating the Flask web server, executing prediction routines, and performing database operations without performance bottlenecks.

For memory, the system requires a minimum of 8 GB DDR4 RAM, which is adequate for basic development and runtime execution. However, 16 GB RAM is recommended to ensure smoother multitasking—particularly when working with Jupyter notebooks, running preprocessing scripts, hosting local servers, or handling in-memory datasets during model testing and prediction.

Storage capacity is also an important consideration. The system suggests having at least 100 GB of available SSD space, which provides fast access times for reading and writing datasets, maintaining logs, and running dependencies. An SSD also helps ensure smooth performance when working with Python environments, the Anaconda distribution, and SQLite database files, as well as storing all assets and libraries required for the Smart Health Advisor.

The system must operate on Windows 10 or later, as this environment supports the necessary Python ecosystem, packages, drivers, and development tools needed for model integration and backend deployment.

Stable internet connectivity serves as another essential hardware interface. A minimum of 10 Mbps bandwidth is recommended to facilitate downloading Python packages, performing updates, synchronizing code with Git repositories, and supporting any optional cloud-based hosting or deployment processes.

Collectively, these hardware resources ensure that the Smart Health Advisor can be developed and executed efficiently, providing sufficient computing power for machine-learning tasks, secure data handling, and responsive interaction through the web application.

4.4.2 Software Interfaces

The Smart Health Advisor system is built on a collection of software interfaces that work together to enable machine learning functionality, secure information processing, and smooth user interaction. The application is deployed on Windows 10 or later, providing

Table 4.2: Software Interface Specifications

Component	Specification	Purpose / Interaction
Operating System	Windows 10 or higher	Foundational platform compatible with all software components.
Programming Language	Python 3.10.9 (via Anaconda)	Core language for server-side logic, ML model training, and data preprocessing.
Web Framework	Flask	Lightweight WSGI web framework for backend server and HTTP request handling.
Development Environment	Jupyter Notebook & Anaconda (2023.03-1)	For exploratory data analysis, model prototyping, and Python environment management.
Database	SQLite3	Lightweight, file-based DBMS for storing user credentials and session data.
Frontend Technologies	HTML5, CSS3	Creates responsive UI; communicates with Flask backend via form submissions.
Version Control	Git	Source code management, version tracking, and team collaboration.

a stable platform that supports all required libraries, development utilities, and runtime components.

The backend is implemented in Python 3.10.9, which forms the core programming environment for executing preprocessing routines, running the machine-learning model, performing AES-256 encryption, and handling server-side logic. Python's rich ecosystem of data-analysis and ML libraries makes it well suited for developing a predictive health-advisory system.

To manage HTTP requests and coordinate interaction between the frontend and backend, the system uses Flask, a lightweight WSGI framework. Flask provides flexible routing, form-handling capabilities, and a straightforward interface for connecting the user interface with the ML inference engine. During the initial development stages, Jupyter Notebook and Anaconda (2023.03-1) were used for tasks such as exploratory data analysis, visualization of dataset patterns, preprocessing experimentation, and iterative refinement of the prediction model.

Data storage is handled by SQLite3, a small and portable relational database engine that stores encrypted user details, past prediction results, and other system records. Its simplicity and file-based architecture make it highly suitable for local development and secure storage of structured data.

The frontend interface is created using HTML5 and CSS3, enabling clean layouts, responsive design, and an intuitive user experience. These technologies support the creation of interfaces for login, registration, health-data submission, prediction viewing, and chatbot communication.

For code management, collaboration, and version tracking, Git is used as the version-control system. It supports organized development workflows, maintains a history of changes, and ensures consistency across updates.

Together, these software components form the technological foundation of the Smart Health Advisor, enabling reliable ML execution, secure data handling, responsive web interaction, and a smooth overall user experience.

4.4.3 Key Python Libraries

The Smart Health Advisor system is supported by a diverse set of Python libraries that collectively enable its machine-learning workflow, data-processing operations, visualization features, and web-application integration. For predictive modeling, the system leverages scikit-learn, XGBoost, and TensorFlow/Keras, each contributing different strengths to the development cycle. Scikit-learn is used to build models such as Random Forest and Logistic Regression, offering a reliable suite of classical ML tools. XGBoost provides an optimized gradient-boosting framework that excels in handling complex tabular data, while TensorFlow/Keras facilitates experimentation with deep-learning architectures during the research and evaluation phases.

Data handling and transformation rely heavily on pandas, which simplifies tasks such as data cleaning, filtering, feature selection, and restructuring. Complementing pandas, NumPy supplies fast numerical operations and efficient multi-dimensional array management, which are essential for scaling and normalizing the health-related parameters used in prediction.

To evaluate model behavior and discover underlying patterns in the dataset, the system incorporates visualization libraries including matplotlib, seaborn, and plotly. These tools make it possible to generate static plots, statistical charts, and interactive dashboards that support performance analysis, metric comparisons, and exploratory visual inspection.

For the web-application layer, Flask acts as the micro-framework responsible for rout-

Table 4.3: Key Python Libraries and Their Functions

Library Category	**Library Name**	**Purpose / Functionality**
Machine Learning	scikit-learn	Building, training, and evaluating traditional ML models (Random Forest, SVM, Logistic Regression).
	XGBoost	Gradient boosting framework for high-performance ML model training and prediction.
	TensorFlow/Keras	Deep learning framework for neural network implementation.
Data Handling	pandas	Data manipulation, analysis, and preprocessing of structured datasets.
	NumPy	Numerical computations and array operations for mathematical transformations.
Visualization	matplotlib	Creating static, interactive, and animated visualizations for data analysis.
	seaborn	Statistical data visualization based on matplotlib with high-level interface.
	plotly	Interactive graphing library for web-based visualizations and dashboards.
Web & Serialization	Flask	Micro web framework for backend development and API endpoints.
	Werkzeug	WSGI utility library for Flask, handling web request/response cycles.
	Jinja2	Templating engine for generating dynamic HTML content in web applications.

ing user requests, processing form inputs, and connecting the frontend interface with the machine-learning backend. Flask works in conjunction with Werkzeug—which handles request and response management—and Jinja2, the templating engine used to render dynamic HTML pages. Together, these libraries ensure that prediction outputs, chatbot responses, and encrypted historical records can be seamlessly integrated into the user interface.

Overall, this combination of Python libraries forms a robust and cohesive development ecosystem, enabling the Smart Health Advisor to deliver accurate predictions, efficient data processing, meaningful visual insights, and a responsive, secure web-based experience.

Chapter 5

System Design

5.1 Architecture Design

The architecture diagram for Smart Health Advisor provides insight into the movement of user information within the system and how each component helps in generating secure predictions and personalized lifestyle support. The process flow begins with the User Input stage, where individuals provide information about health and routine data, such as age, BMI, weight, frequency of activities, sleep behaviors, diet preferences, and more. These inputs act as the basis for all further analyses conducted by the platform.

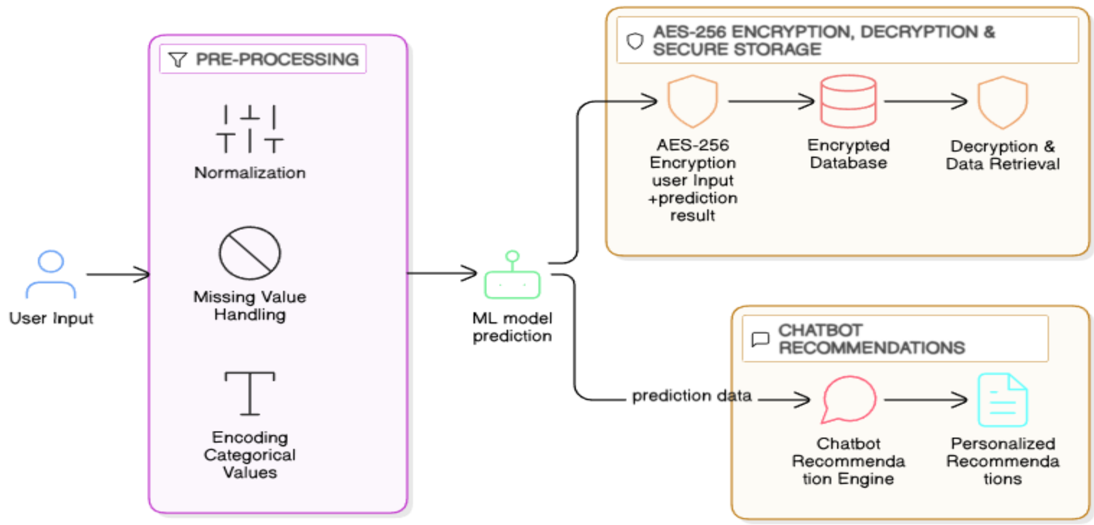


Figure 5.1: Architecture Diagram

The information is then channeled to the Preprocessing Module, which prepares the data for machine-learning evaluation. In this step, it replaces missing entries, scales numerical attributes, and converts categorical variables into machine-readable forms. The

Preprocessing Module consolidates input into a standard state such that noiseless, consistent data reaches the learning model, improving predictability and reliability.

Further, the processed data goes to the Machine Learning Model, in which a Random Forest classifier evaluates the input data and identifies the user's risk concerning obesity, hypertension, or diabetes. The model uses patterns learnt during training for producing a prediction that will drive further steps in the system.

Once a prediction is generated, the workflow divides into two functional paths: the first leads to the AES-256-GCM Encryption and Secure Storage Unit, where both the user's original inputs and the generated prediction are encrypted and saved in the database. This approach ensures that health information remains confidential and can be safely retrieved later. If the user decides to review their history, the system decrypts the stored records and will present the original details in a protected and controlled manner.

The second path sends the prediction to the Chatbot Recommendation Engine. This module uses rule-driven instead of adaptive AI to generate recommendations focused on a detected risk category. To illustrate, users identified as obese would get recommendations relating to calorie intake and exercise, while those at risk for hypertension are advised to manage their stress, limit the use of sodium, and practice habits that are heart-friendly. Individuals with a high risk for diabetes might be advised about controlling blood sugar levels, nourishing their bodies properly with balanced nutrition, and participating in regular exercise. These recommendations are provided in a user-friendly, readable format to support the adoption of practical changes into one's daily routine.

Put together, these modules provide an architecture that progresses user information from its raw input, through preprocessing and prediction, to secure storage and personalized assistance. The diagram shows a system not only for the precise evaluation of health risks but also for a strong data protection and clear, helpful guidance for the users—and all that makes Smart Health Advisor a reliable tool for proactive health management.

5.2 Class Diagram

The **Class Diagram** of the *Smart Health Advisor* presents the object-oriented layout of the system by outlining its main classes, their attributes, operations, and how they relate to one another. It offers a static snapshot of the system, highlighting how data and functionality are organized within each module.

This diagram provides developers with a clear understanding of the system's internal structure, showing how responsibilities such as data preparation, model training, user management, prediction, encryption, and chatbot interaction are separated across different classes. By visualizing these relationships, the Class Diagram supports efficient development, modular design, and easier future scalability.

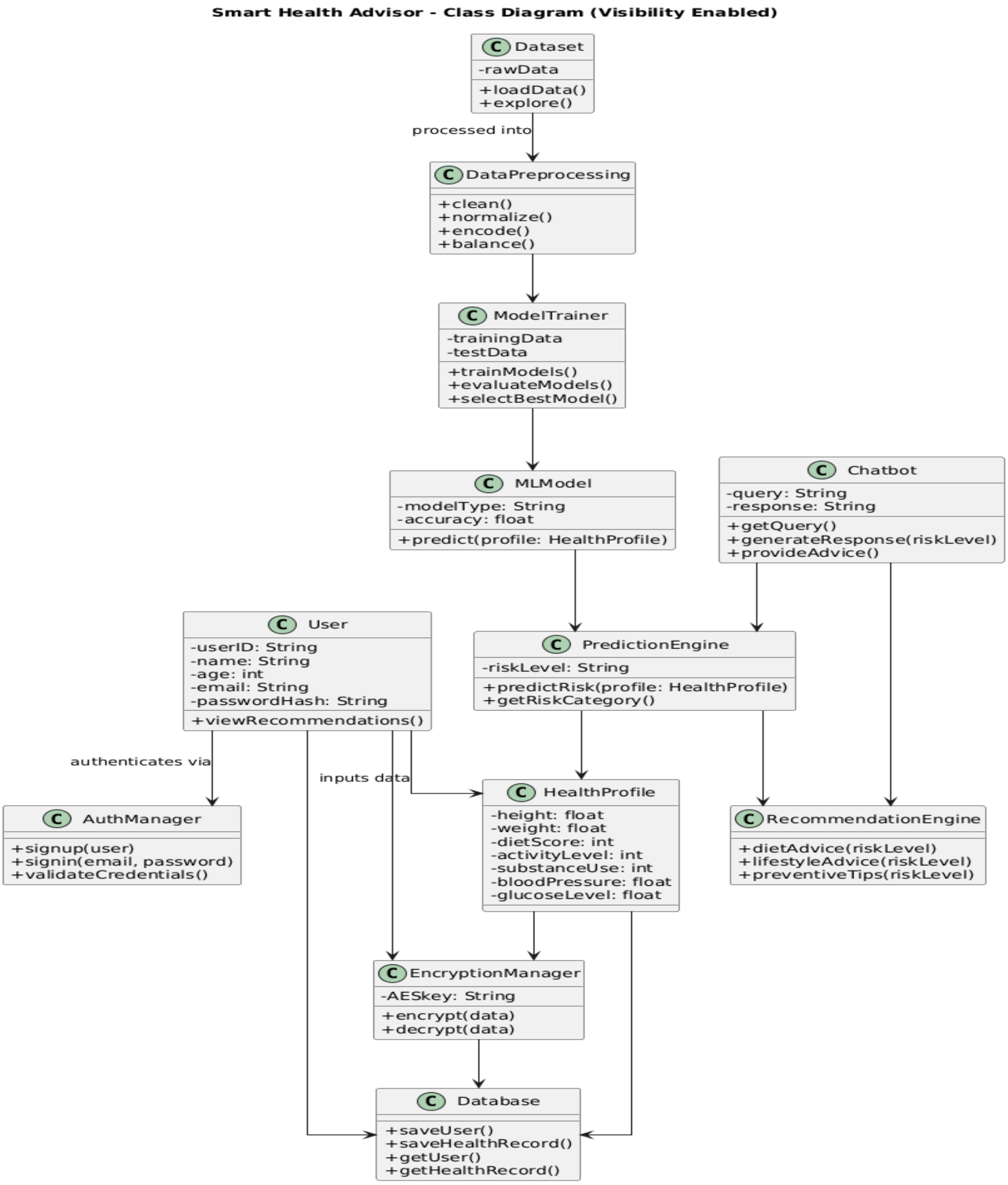


Figure 5.2: Class Diagram

In this regard, the class diagram of the Smart Health Advisor system describes how the major components are structured and interrelated to support activities in the application, such as prediction, security, and personalized guidance. First of all is the Dataset class, which is responsible for loading and inspecting the raw health dataset used for building the prediction model. The raw dataset is passed on to the DataPreprocessing class, where cleaning of incomplete entries, scaling of numeric values, encoding of categorical attributes, and balancing of class distributions among other activities are carried out to prepare the data for training.

After preprocessing is complete, the resultant dataset is passed to the ModelTrainer class, which executes a set of machine-learning algorithms, compares performance, and trains the best model. The best model selected is then saved in the instance of MLModel class, comprising the algorithm used, its evaluation metrics, and its prediction method for future user inputs.

The User class is the very beginning of the user side of this system. It stores information like the unique ID of the user, personal info, and the hash of the password. Authentication-related tasks are handled by the AuthManager class, which creates and manages accounts, handles logins, and checks credentials. After authentication, users input their health profile information, which is stored by an instance of the HealthProfile class. This class aggregates vital metrics such as height, weight, diet scores, activity level, substance use, blood pressure, and glucose measurements.

The PredictionEngine class links the user data with the trained model. It receives the HealthProfile instance, pre-processes attributes if needed, and then uses the MLModel to predict the user's risk classification for obesity, hypertension, or diabetes. Finally, before any prediction result is persisted, the EncryptionManager applies an AES-256-GCM encryption on both the submitted health data and the model's output. Secure storage and retrieval of these encrypted records are handled by the Database class, which maintains user accounts, encrypted health profiles, and historical prediction entries.

The RecommendationEngine and Chatbot classes within the system support personalized health improvement. The RecommendationEngine generates recommendations, nutritional advice, planning of activities, and changes in behavior according to the predicted risk category. A Chatbot with rule-based logic and keyword recognition can be used to interact with users, answer health-related questions, and give targeted suggestions related to the predictions derived from the system.

Collectively, these classes form a cohesive design to be able to smoothly move information from data ingestion and preprocessing through prediction, encryption, storage, and personalized health recommendations. The class diagram shows how the pieces work together to create a secure intelligent digital health advisor.

5.3 State Diagram

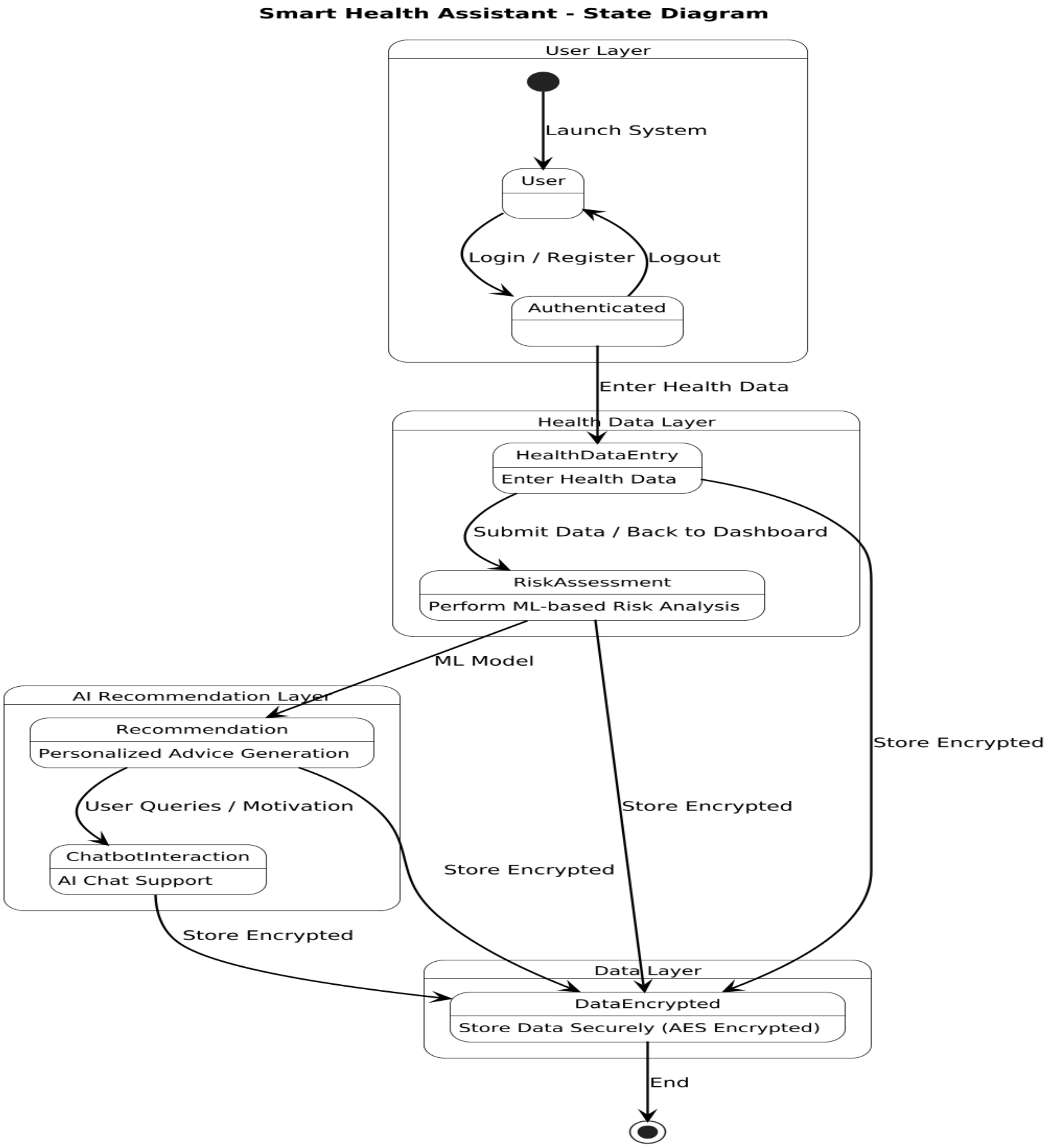


Figure 5.3: State Diagram

The Smart Health Advisor State Diagram describes the transition of the system in moving through different phases of operations during one full user session. It explains the sequence of internal steps the platform performs based on user actions, prediction workflows, and requirements for secure data handling. This representation helps explain how diverse components, such as authentication, preprocessing, prediction, the chatbot, and encryption, interact right from the moment a user logs into the system to the end.

The interaction starts with the Unauthenticated State, at which point a user has to either create an account or log in using existing credentials. In cases of failed login attempts, the system remains in this state to protect it from unauthorized access. When the user inputs the correct credentials, the system moves to the Authenticated State, where a user can assess the health-prediction capabilities.

Following authentication, the system transitions to the HealthDataEntry State, where users can input data like height, weight, BMI, sleep hours, activity level, blood pressure readings, glucose values, and other lifestyle attributes. After the necessary information is furnished, the system processes the input and transitions into the RiskAssessment State. During this state, the platform cleans, normalizes, and encodes the input data before passing the final input vector to the trained Random Forest classifier; then it makes a prediction of the user's risk level related to obesity, hypertension, or diabetes.

Once a prediction is made, the system triggers two states to run in parallel. The first one is the Recommendation State, wherein the output is fed to the rule-based chatbot and recommendation engine; the chatbot, using predefined logic with keyword matching, generates specific suggestions regarding diet, exercise, sleep, and lifestyle management for the predicted condition.

The second parallel state, DataEncrypted State, is responsible for ensuring the security of user data. In this state, input data and the corresponding prediction are encrypted using AES-256-GCM and stored in the database. If a user checks his history later on, the system revisits this state, decrypting and returning the stored records safely.

In case users want to keep on asking health-related questions, the system moves into the ChatbotInteraction State, where the rule-based chatbot can give more advice without starting another prediction cycle. This interaction will proceed until such time as the user decides to end the conversation.

The session concludes when the system transitions into the Logout State, which revokes access rights and returns the platform to the Unauthenticated State, ready for the next login attempt.

Overall, the State Diagram represents the complete operational logic of the Smart Health Advisor: from authentication and data entry to prediction and recommendation, encryption, and session termination. Such a structured approach provides clarity regarding system behavior, assists in debugging and maintenance, and ensures functional requirements can be consistently validated during development.

5.4 Use Case Diagram

The **Use Case Diagram** illustrates how users interact with the Smart Health Advisor system and outlines the primary functions the platform provides. It presents the key activities a user can perform—such as creating an account, logging in, entering personal health information, requesting risk assessments, chatting with the chatbot, and ensuring their data is stored securely.

By visualizing these interactions, the diagram clarifies the system's functional requirements and helps define the scope of services offered to the user.

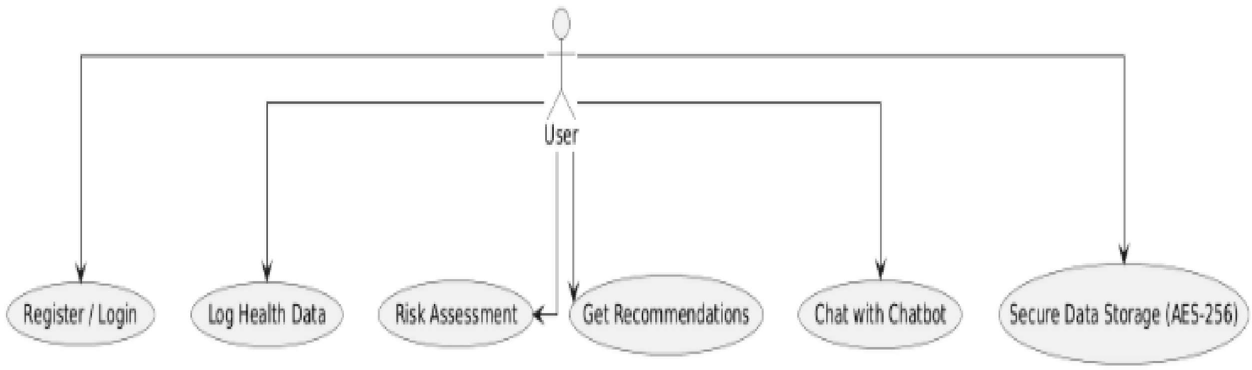


Figure 5.4: Use Case Diagram

The Use Case Diagram for the Smart Health Advisor system depicts the interaction of the primary user with the main functions provided by the platform. The User is the central actor and the instigator of all activities involving the system. The use case interaction starts with the Register/Login, enabling the creation of new users or authentication of returning users. This ensures that only verified persons can access the health-analysis features provided by the system.

The user then logs in and moves to the Log Health Data use case. The user fills in the necessary lifestyle and physiological information, such as BMI, dietary behavior, exercise, sleep hours, blood pressure, and blood glucose level, required as an input to feed into the system's analysis.

Having provided this data, the user can activate the use case called Risk Assessment. This action will trigger the machine-learning model of the system to analyze the data provided and issue a risk estimate for diabetes, hypertension, or obesity. The result of such an analysis immediately interfaces with the Get Recommendations use case, wherein the platform will provide actionable suggestions that might include nutritional advice, an exercise routine, better habits for daily living, or even preventive activities based on the level of risk resulting from the analysis.

Another use case also included in the Use Case Diagram is the Chat with Chatbot feature, which will engage the user in an interaction with a rule-based chatbot for prompt, formatted answers to health-related questions. It allows the user to receive immediate advice or clarification without the need to spend time studying complicated menus or understand technical and medical details.

The other most vital part represented in the diagram is the Secure Data Storage use case. Since the system deals with sensitive health information, the data submitted and the prediction results will be encrypted with AES-256-GCM before being stored into the database. This guarantees privacy, prevents access from unauthorized actors, and enables users to fetch their historical records safely whenever needed.

The boundary of the system in this diagram separates the user-initiated actions from the internal processes of authentication, data preprocessing, model evaluation, encryption, and recommendation generation. The associations between the user and each of the use cases reflect a very straightforward workflow-from logging in and entering data to obtaining predictions, obtaining lifestyle guidance, and securely storing health information. Overall, the Use Case Diagram expresses the main capabilities of the Smart Health Advisor system, further elucidating how a user interacts with its features. Basically, it is an important reference for comprehension of both functional requirements that the system meets and common user journeys across the system.

5.5 Sequence Diagram

The **Sequence Diagram** demonstrates the step-by-step flow of interactions between the main components of the Smart Health Advisor system. It outlines how the User, Frontend, Backend, Database, ML Model, and Chatbot communicate with one another to complete various operations.

By visualizing the order of messages exchanged, this diagram helps clarify how real-time processing works within the system and how data moves across different modules during user activities.

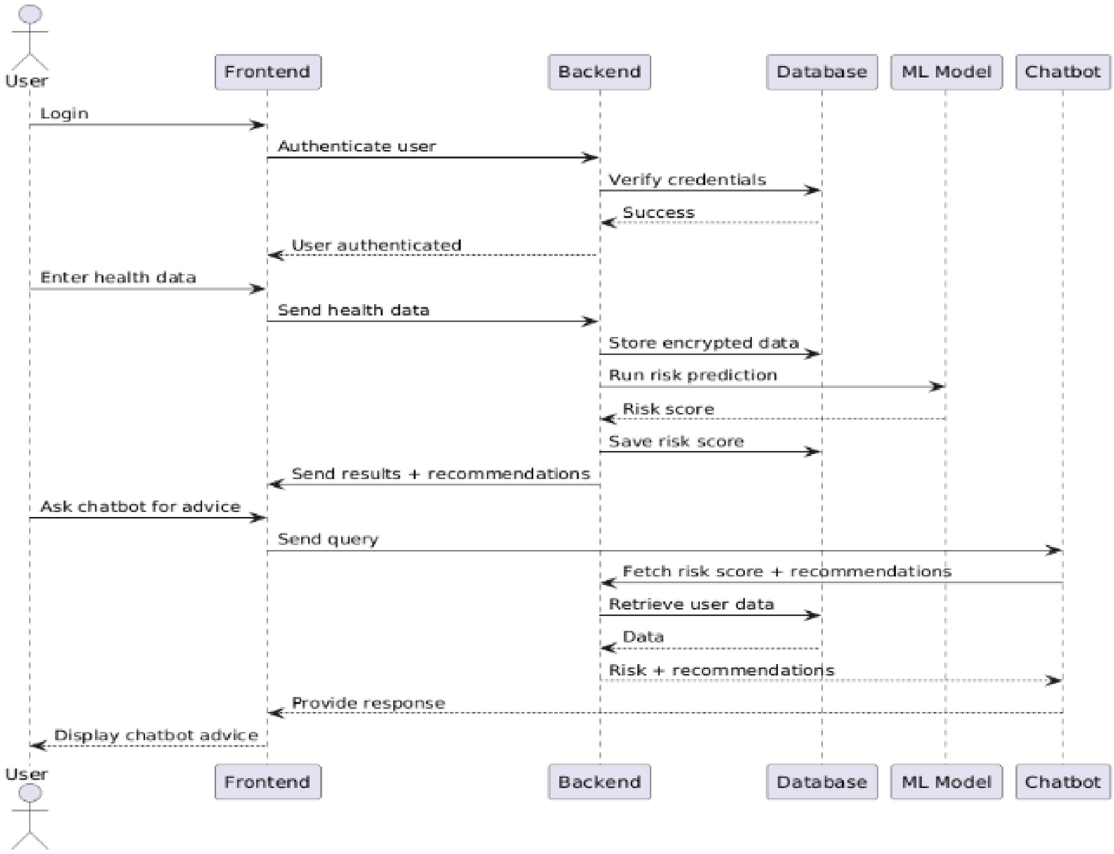


Figure 5.5: Sequence Diagram

The Smart Health Advisor system's Sequence Diagram shows the sequential exchanges that take place when the user interacts with various backend components. When a user interacts with the Frontend Interface to log in or create a new account, the process starts. The Backend verifies the user's identity by querying the database after receiving the submitted credentials from the Frontend. The backend either returns an authentication success or a prompt asking for updated login information, depending on the outcome. Following a successful login, the user provides health-related information via the frontend, which then sends it to the backend.

After receiving the health parameters, the backend performs some initial preprocessing before sending the cleaned data to the machine learning model. After analyzing the data, the model predicts the risk of obesity, diabetes, or hypertension. After the prediction, the user input and the prediction result are encrypted using AES-256-GCM by the Encryption Module, which is contacted by the backend. To ensure safe long-term storage, the encrypted package is kept in the database. The encrypted records are retrieved by the backend, sent to the encryption module for decryption, and returned to the frontend for display when a user subsequently requests their history.

Simultaneously, the backend sends the prediction output to the Chatbot Module, which employs pre-established rules to produce customized lifestyle recommendations, such as dietary advice, physical activity recommendations, sleep enhancement techniques, and preventive advice. The user can review these responses after they are returned to the frontend. If the user keeps interacting with the chatbot, the frontend sends each new query to the backend, which then speaks with the chatbot once more to generate more responses, creating an ongoing dialogue.

When the user chooses to log out, the interaction comes to an end. The backend closes the active session and restores the application to its starting state as a result of this action.

Overall, the sequence diagram illustrates the coordinated flow of information across system components—from input collection and model prediction to secure storage and personalized guidance—highlighting how real-time communication between modules shapes the Smart Health Advisor's behavior.

5.6 Activity Diagram

The Activity Diagram for the Smart Health Advisor shows the order in which a user interacts with the platform, from logging in to finishing a session. The flow starts when the user opens the app and goes to the login screen. The process moves forward if the user gives valid credentials. If not, the user is taken through the registration process and then sent back to the login step. The user goes to the Health Data Entry activity after they have been authenticated. Here, they enter information like their BMI, dietary category, activity level, sleep hours, blood pressure, and glucose levels.

The system goes into the Preprocessing Activity after it gets the data. This is where it standardizes the input by dealing with missing values, scaling numerical fields, and encoding categorical entries. After preprocessing is done, the workflow moves on to the Risk Prediction stage. Here, the trained machine-learning model looks at the formatted input and gives a risk level for diabetes, high blood pressure, or obesity.

The workflow divides into two concurrent tasks after a prediction is produced. The first route leads to the Encryption and Secure Storage stage, where AES-256-GCM encryption is used to securely store both the prediction output and the user-provided data in the database.

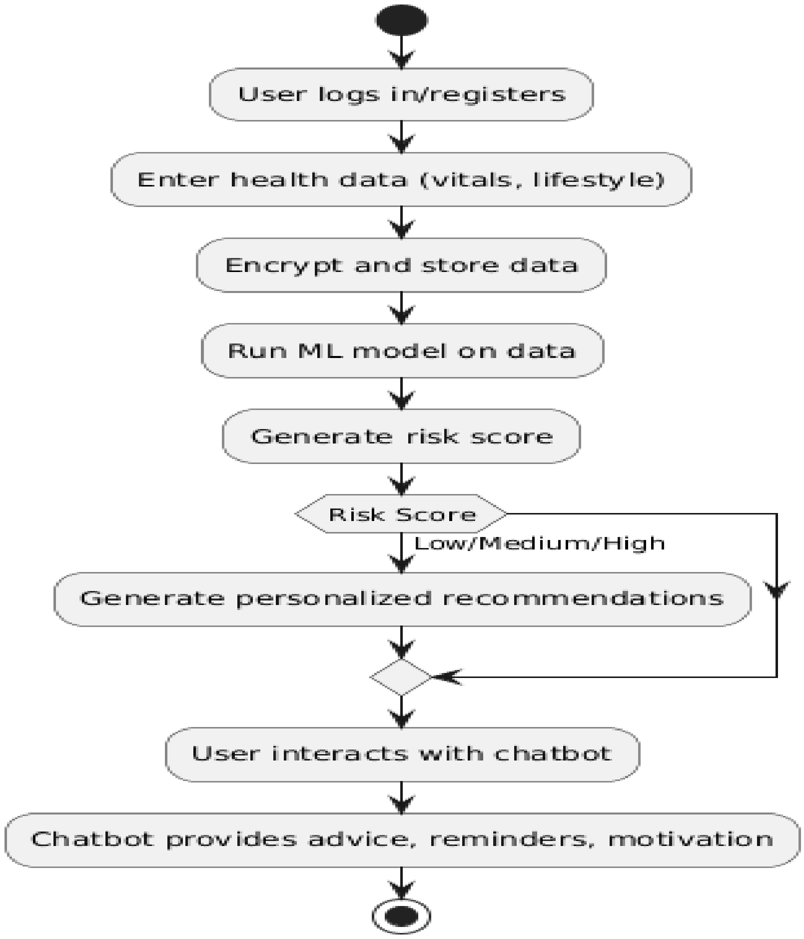


Figure 5.6: Activity Diagram

The second route leads to the Chatbot Recommendation activity, where a rule-based chatbot analyzes the prediction result and offers customized recommendations on preventive lifestyle habits, exercise, stress management, diet, and sleep patterns. The activity proceeds through additional chatbot interactions if the user requests more assistance; otherwise, the flow proceeds to the logout step.

The user's interaction with the system comes to an end when the session ends at the End Point. All things considered, the Activity Diagram offers an organized perspective of the system's operational logic, showing how various processes work both sequentially and concurrently to provide a smooth user experience. This includes everything from data submission and authentication to prediction, safe storage, and customized recommendations.

5.7 Data Flow Diagram

The **Data Flow Diagram (DFD)** illustrates how information moves through the Smart Health Advisor system. It highlights each stage of the data lifecycle—including user input, secure encryption, machine-learning analysis, and the generation of personalized feedback.

By visualizing these flows, the DFD provides a clear picture of how the system processes and manages data, making it easier to understand the logic and structure behind its operations.

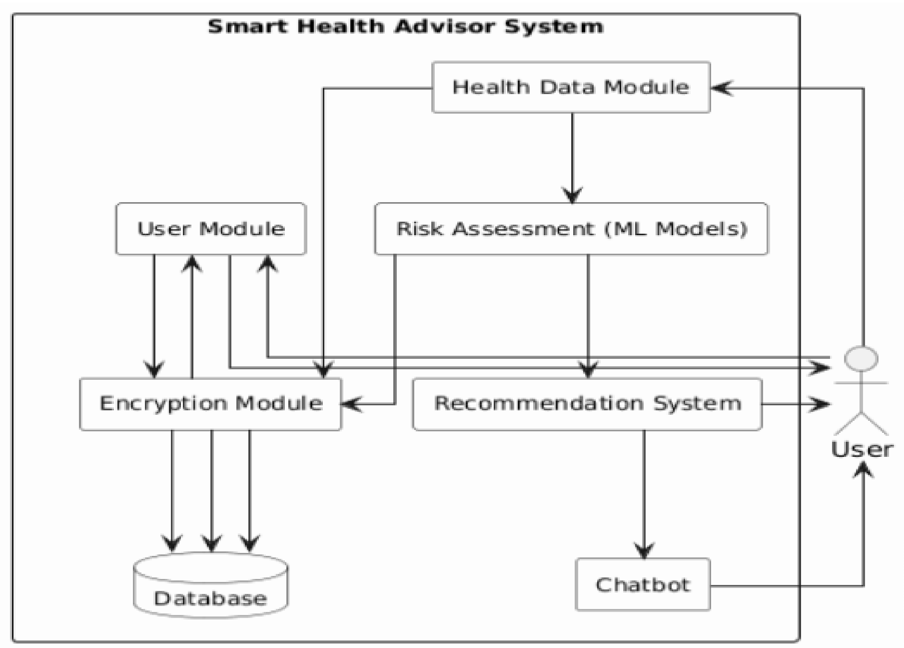


Figure 5.7: Data Flow Diagram

The Smart Health Advisor's Data Flow Diagram (DFD) shows the flow of data between users, internal data stores, and system processes. The user initiates the flow by entering vital health-related information such as age, blood pressure, glucose readings, activity level, dietary habits, BMI, and other lifestyle factors. These inputs go through the Health Data Entry process, where the platform gathers and validates the supplied values before forwarding them for additional analysis.

Preprocessing and Risk Evaluation are the next steps for the verified inputs. In this case, the system prepares the data by encoding categorical attributes, scaling numerical features, and carrying out any required cleaning. The Machine Learning Model receives the processed data and uses a trained Random Forest classifier to determine the probability of conditions like obesity, diabetes, or hypertension. For later system actions, the generated prediction is included in the outgoing data stream.

Preprocessing and Risk Evaluation are the next steps for the verified inputs. In this case, the system prepares the data by encoding categorical attributes, scaling numerical features, and carrying out any required cleaning. The Machine Learning Model receives the processed data and uses a trained Random Forest classifier to determine the probability of conditions like obesity, diabetes, or hypertension. For later system actions, the generated prediction is included in the outgoing data stream.

The Recommendation Generation process is approached by the other stream. This component determines the kind of lifestyle advice the user should receive based on the predicted disease category. The outcome is sent to the Chatbot Interaction process, which generates pertinent guidance using rule-driven logic. The chatbot may make dietary, exercise, behavioral, or general wellness recommendations based on the identified risk. The advisory cycle is then completed by sending these outputs straight back to the user.

The overall flow of user data through validation, preprocessing, prediction, secure storage, and recommendation delivery is depicted in the diagram. While upholding stringent protection of personal data, it emphasizes the organized interaction between key system components, including machine learning, encryption, database access, and chatbot-based guidance. The DFD gives a clear picture of how the Smart Health Advisor transforms unprocessed health data into secure records and relevant, tailored recommendations.

5.8 Modular Diagram

The **Modular Diagram** divides the Frontend, Backend, and Database—the three primary layers of the Smart Health Advisor—into discrete, manageable modules. Because each module is in charge of a particular aspect of the system's functionality, the platform can function effectively and continue to be simple to maintain or upgrade in the future.

The diagram illustrates how various parts cooperate while maintaining their independence by dividing the system into these modules. As new features are added, the system becomes more adaptable thanks to this structure's support for a scalable and modular architecture.

The Frontend, Backend, and Database are the three main parts of the Smart Health Advisor's Modular Diagram, each of which is responsible for a specific set of tasks. The system's architecture is made clearer by this division, which also facilitates future growth and maintenance.

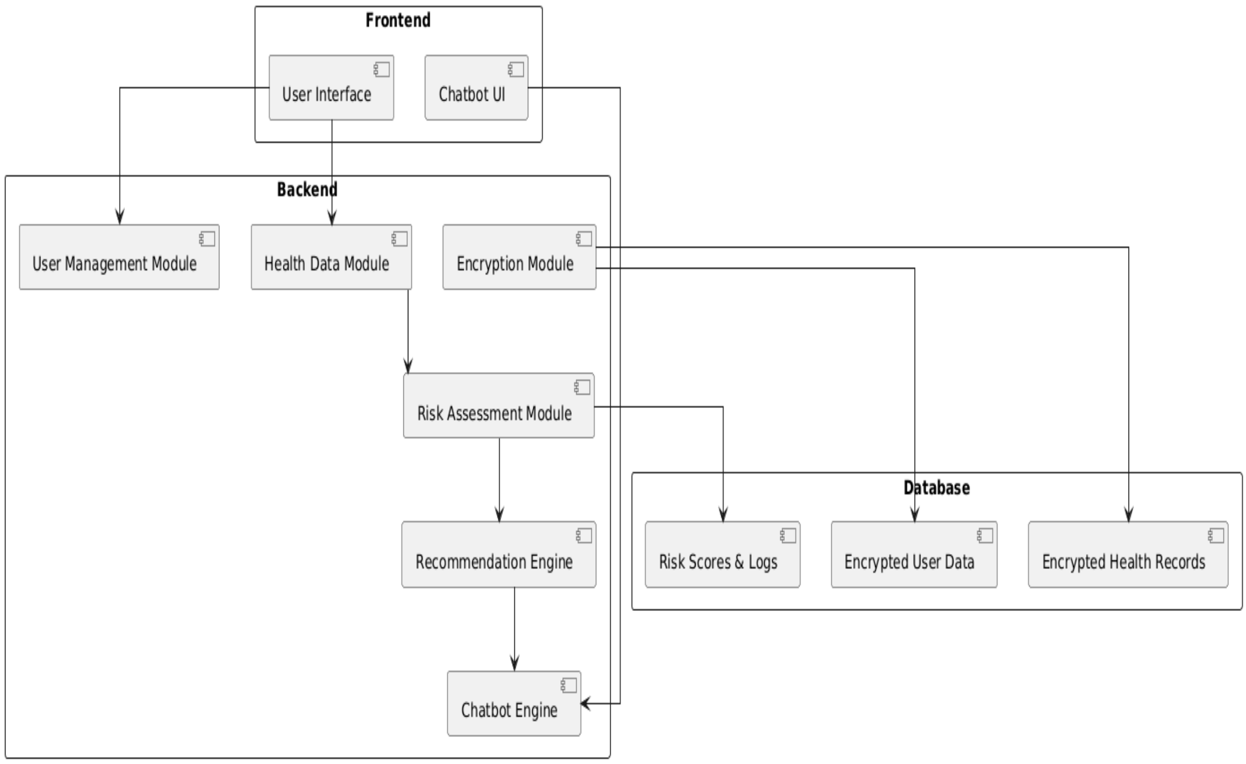


Figure 5.8: Modular Diagram

The Frontend Module, which represents everything the user directly interacts with, is at the top layer. This includes the health data entry forms, the prediction display screen, the conversational chatbot panel, and the login and registration interfaces. This module initiates all user actions, including sending questions to the chatbot, viewing results, and submitting health information. Its main objective is to serve as the gateway to the system's features while offering a responsive and easy-to-use user experience.

The Smart Health Advisor's computational and logical core is the Backend Module. It was created with Flask and handles data preprocessing, session management, authentication, and machine learning-based disease prediction. In addition to executing the rule-based logic that drives the chatbot's responses, the backend uses the Random Forest classifier to determine the user's risk of diabetes, hypertension, or obesity. Furthermore, the AES-256-GCM encryption process is integrated into the backend to guarantee that user inputs and prediction outputs are encrypted prior to storage. This module is in charge of organizing all significant activities, such as making suggestions, answering user inquiries, and enforcing data security protocols.

The Database Module serves as the system's secure storage layer. It contains prediction histories, encrypted health records, and user account information. Confidentiality and integrity are maintained even in the case of unwanted access attempts because all data stored in the database is encrypted. The database and the backend work closely together to support functions like user information updates, past prediction retrieval, and login verification. It offers a reliable and long-lasting repository for the data in the system.

The core of the system's data flow is communication between these three modules. The frontend sends user inputs to the backend for processing; the backend stores or retrieves encrypted records by consulting the database; and the processed outputs, including past results, chatbot recommendations, and predictions, are returned to the frontend for display.

When combined, these elements form a unified and well-structured architecture with distinct roles for computation, storage, and presentation. The modular diagram illustrates how the Smart Health Advisor combines interactive guidance, secure data handling, and machine learning into a single digital health platform.

Chapter 6

Implementation

This chapter outlines how the **Smart Health Advisor: Risk Assessment and Lifestyle Guidance for Chronic Disease Prevention** was implemented. It details the design and execution of each module based on the system's architectural plan. All components were developed in Python, with Flask serving as the primary web framework. These modules work together with machine learning models and a secure database to create a fully integrated, AI-driven health advisory system.

6.1 Algorithm

6.1.1 Modules and Pseudocode

The Smart Health Advisor system is implemented using a modular and structured approach to ensure scalability, clarity, and ease of maintenance. Each module performs a specific function within the overall system workflow—from data preparation to model prediction and user interaction. The following subsections provide an overview of the key modules involved in the implementation, along with pseudocode that outlines their internal operations.

1. Data Import and Exploration Module

Purpose: To load the dataset, examine its structure, and understand its statistical distribution. This step helps identify missing values, outliers, and inconsistencies that may affect model performance.

Pseudocode:

```
Start
```

```
Import pandas, numpy, matplotlib, seaborn
Load dataset into DataFrame
Display dataset info(), head(), describe()
Identify missing or null values
End
```

2. Data Preprocessing Module

Purpose: To clean and transform the dataset into a standardized format. This includes handling missing values, encoding categorical attributes, and normalizing numerical fields for consistent model training.

Pseudocode:

```
Start
For each column in dataset:
    If categorical → apply label encoding
    If numerical → apply normalization
Check for null values and replace with mean/median
Split dataset into input (X) and target (Y)
End
```

3. Data Balancing Module

Purpose: To resolve class imbalance in the dataset using Random Oversampling. This ensures equal representation of all classes and improves the model's recall and F1-score.

Pseudocode:

```
Start
Check class imbalance using value_counts()
If imbalance detected:
    Apply RandomOverSampler()
Return balanced dataset
End
```

4. Model Training Module

Purpose: To train and evaluate multiple ML models—Random Forest, Decision Tree, Logistic Regression, SVM, MLP, XGBoost, and Ensemble. The best-performing model is selected and saved for deployment.

Pseudocode:

```
Start
Split data into training (70%) and testing (30%)
Initialize models:
    RF, DT, LR, SVM, MLP, XGB, Ensemble
For each model in list:
    Train model on training data
    Predict on test data
    Calculate accuracy, precision, recall, F1
Select best performing model
Save model using joblib
End
```

5. Flask Web Interface Module

Purpose: To provide a web interface where users can register, log in, and submit health data for prediction.

Pseudocode:

```
Start Flask server
Define routes: /login, /register, /predict
On form submission:
    Validate input data
    Preprocess input data
    Pass data to trained ML model
    Return predicted risk category
End
```

6. Chatbot Integration Module

Purpose: To provide an interactive chatbot that delivers personalized guidance, dietary tips, and preventive measures.

Pseudocode:

```
Start
Initialize chatbot engine
While chat session active:
    Accept user query
    Match keywords to response rules
    Retrieve health tips/recommendations
    Display response
End
```

7. Output and Visualization Module

Purpose: To display results such as prediction accuracy, precision, recall, F1-score, and comparison charts for model performance.

Pseudocode:

```
Start
Collect model evaluation metrics
Use matplotlib/seaborn to plot comparison graphs
Display highest accuracy model
End
```

—

6.1.2 Algorithm: Random Forest for Chronic Disease Risk Assessment

1. Input: Preprocessed health dataset $D = \{x_1, x_2, ..., x_n\}$ with class labels.
2. Split D into training and testing sets.
3. For $i = 1$ to N (number of trees):
   (a) Randomly sample data with replacement (bootstrap).

   (b) Build a decision tree using a random subset of features.

4. For each new input x:

   (a) Predict class from each decision tree.

   (b) Take majority vote across all trees.

5. Output: Final class label (Risk)

Complexity: $O(n \log n)$ per tree; Ensemble improves overall accuracy and generalization.

Because of its ability to learn from extensive and intricate datasets in the health sector, Random Forest was selected to be utilized within the Smart Health Advisor system as the primary classification algorithm. Unlike single decision trees, which are taught how to classify based on one collection of data, the Random Forest method encases several decision trees (given the plethora of variables available), with each tree building its foundation on the basis of random sample variations taken from the same dataset. Using this method, Random Forest is able to identify many different patterns through the variables or attributes expressed by the user, i.e. age, Body Mass Index, glucose levels, amounts of sleep per day, eating habits, levels of physical training/exercise etc.

Once a prediction is made, each tree within the Random Forest votes as to which risk category the user fits into. The final predicted risk category is derived from the voting by all of the trees (each tree casts one vote). This voting mechanism for predicting risk categories helps offset the potential overfitting error that any single decision tree could create as well as ensuring consistent and reliable output from the Random Forest algorithm.

For healthcare applications, a substantial benefit provided by Random Forest is that it may be used with numerical or categorical data and effectively processes and analyses the data even when the data set is not evenly distributed or there exists missing records.

Within the parameters of this project, Random Forest out performed all other algorithms evaluated within this study with regard not only to the accuracy of the predictions but also with respect to the F1-scores of predictions, making it (Random Forest) the best choice for predicting chronic disease risk outcomes.

6.1.3 Algorithm:AES-256-GCM Encryption for Secure Health Data Storage

1. Input: Username u, Data D (inputs + prediction result).
2. Derive a 256-bit key:
   (a) Convert u to bytes.
   (b) Compute $key = SHA256(u)$.
3. Convert D to JSON string form.
4. Generate a random 12-byte nonce:
$$nonce = os.urandom(12)$$
5. Initialize AES-GCM cipher using the derived key.
6. Encrypt the data:
$$ciphertext = AESGCM.encrypt(nonce, D, None)$$
7. Concatenate $nonceandciphertext$.
8. Base64 encode the final encrypted result.
9. Output: Encrypted record E stored in database.

Security: AES-256-GCM provides confidentiality and integrity using authenticated encryption.

To protect all sensitive health data using a strong level of encryption, the Smart Health Advisor utilizes Advanced Encryption Standard 256 in Galois Counter Mode (AES-256-GCM). This level of encryption is imperative in order to meet the security needs of organisations that have strict compliance.

The encryption key used by the Smart Health Advisor is not maintained in the system. Instead, a new key is generated each time it is needed by applying a SHA-256 hash to your username. As a result, each unique user begins with a uniquely generated encryption key. Each time a user logs in, a new 12-byte nonce (a randomly generated number) is generated by the Smart Health Advisor, along with the creation of the key. Every time a user logs in, a new authentication tag is provided by the Smart Health Advisor which allows the user to determine if there is an attempt to change or tamper with the encrypted message. If the encrypted message is manipulated or corrupted, it will not successfully be decrypted.

All predictions, such as user-inputted data and risk assessments, are first encrypted using AES-256-GCM prior to being converted to base64 encoding and stored in an SQLite database. The combination of encryption with integrity checks will allow users to remain confident that their stored medical data is confidential and trustworthy, even if a third party were to access the SQLite database, from which it is stored. The approach taken by the Smart Health Advisor meets or exceeds the strong security requirements that are associated with healthcare systems.

6.1.4 Algorithm: AES-256-GCM Decryption for Retrieving User History

1. Input: Username u, Encrypted record E (Base64 encoded).
2. Derive the same 256-bit key:
   (a) Convert u to bytes.
   (b) Compute $key = SHA256(u)$.
3. Base64 decode E to obtain raw bytes.
4. Extract:
   - $nonce$ = first 12 bytes,
   - $ciphertext$ = remaining bytes.
5. Initialize AES-GCM cipher using the derived key.
6. Decrypt the data:

$$plaintext = AESGCM.decrypt(nonce, ciphertext, None)$$

7. Convert plaintext bytes to string.
8. If plaintext is JSON:
   (a) Parse and return it as a dictionary.

9. Output: Original data D (inputs + prediction result).

Integrity: Decryption fails if ciphertext or nonce is modified, ensuring data authenticity.

The decryption component of the Smart Health Advisor system is responsible for restoring stored user information when past predictions are requested. Because the same key-derivation method is used for both encryption and decryption, the system never needs to save any secret keys. Instead, it recreates the required 256-bit key by hashing the user's username, ensuring that only the corresponding user can unlock their encrypted data.

When the user opens their history, the system fetches the encrypted entry and first decodes it from its base64 representation. It then extracts the 12-byte nonce and the encrypted payload. AES-GCM relies on these elements to recover the original data. If any part of the encrypted content—including the nonce or the authentication tag—has been changed, AES-GCM will detect the discrepancy and refuse to decrypt it, preventing the system from displaying data that has been tampered with.

After successful decryption, the plaintext is parsed back into the JSON structure containing the user's earlier inputs and the associated prediction output. This process allows previous results to be shown accurately while preserving the confidentiality and integrity of sensitive health information throughout the system.

6.2 Flow Chart

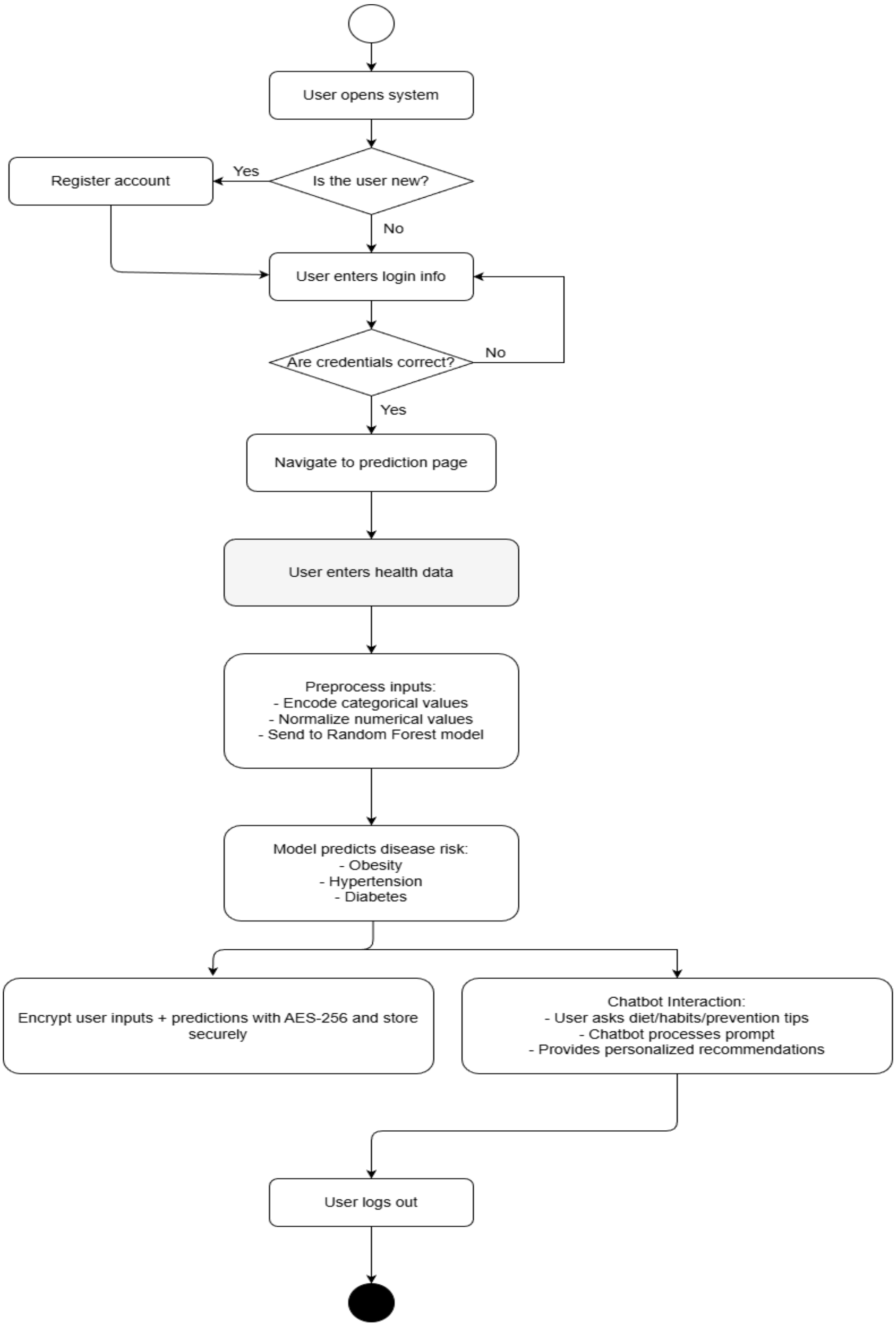


Figure 6.1: Flowchart Illustrating the Functional Process of the Smart Health Advisor

Figure 6.1 presents the end-to-end workflow of the Smart Health Advisor system, outlining the process from user access to prediction, recommendation generation, secure storage, and final logout. When a user launches the platform, they must create an account if they are new to the system by providing a username and password. Returning users can proceed directly to the login page. The system authenticates the credentials by comparing them with entries in the database; incorrect details prompt the user to retry, while successful verification grants access to the main dashboard.

From the dashboard, the user moves to the prediction interface and enters health-related information such as age, BMI, sleep duration, activity level, dietary habits, glucose readings, blood pressure measurements, and other lifestyle attributes. After submission, this information is processed by the preprocessing module, which performs operations such as numerical scaling, categorical encoding, and formatting the data for compatibility with the trained machine-learning model. These steps ensure that the input is consistent and ready for analysis.

The cleaned data is then supplied to the Random Forest–based prediction model, which evaluates the input and generates a risk classification for obesity, hypertension, or diabetes. Once the prediction is produced, the workflow splits into two parallel paths. In the first, the system encrypts both the user's inputs and the resulting prediction using AES-256-GCM, enabling confidential and tamper-resistant storage in the database. This encryption step protects the user's medical information from unauthorized access.

In the second path, the prediction is passed to the rule-based chatbot. Using predefined rules and keyword-driven logic, the chatbot prepares tailored recommendations related to diet, physical activity, stress management, and general wellness. These responses provide users with immediate insight into lifestyle adjustments aligned with their predicted health condition.

After reviewing the results and personalized suggestions, the user can log out, which securely terminates the active session and returns the system to its start state. Overall, the workflow in Figure 6.1 highlights how machine-learning prediction, automated guidance, and encrypted storage are combined into a cohesive and secure digital health-advisory process.

6.3 Implementation Codes

The development of the Smart Health Advisor system involves constructing the machine learning pipeline, implementing data preprocessing procedures, integrating an encryption layer, and building a web interface using Flask. This section outlines the core components of the codebase responsible for training the model and deploying it through a functional web application.

The machine learning workflow starts with loading the dataset and preparing it for analysis through tasks such as feature encoding, numerical scaling, and splitting the data into training and testing partitions. A Random Forest classifier is then fitted to the processed dataset, and its performance is assessed using evaluation measures including accuracy, precision, recall, and F1-score. Once an optimal model is obtained, both the classifier and the preprocessing scaler are stored using the joblib library so they can be reused during deployment.

For user interaction, the system incorporates a Flask-based application that serves as the bridge between the trained model and the client-side interface. The backend retrieves the saved model, accepts user-provided values from HTML form submissions, converts and scales these inputs appropriately, and then generates predictions using the Random Forest classifier in real time. The output is returned to the corresponding HTML template, where the result is displayed to the user.

Overall, this implementation demonstrates how a machine learning model can be integrated into a lightweight web framework to provide immediate health-related predictions. By combining Python, Flask, HTML templates, and a deployed model, the system allows users to seamlessly interact with the predictive engine and obtain instant feedback. The code excerpts below highlight the primary steps involved in constructing the model and embedding it within the web application.

6.3.1 Model Training and Prediction

```
# Import necessary packages
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import joblib

# Load and preprocess dataset
data = pd.read_csv('health_dataset.csv')
X = data.drop('Risk', axis=1)
y = data['Risk']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
```

```
# Standardize data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict and evaluate
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

# Save model for deployment
joblib.dump(model, 'Models/model.sav')
```

6.3.2 Flask App Integration:

```
# Flask application for chronic disease prediction
from flask import Flask, request, render_template
import joblib
import numpy as np

# Initialize Flask application
app = Flask(__name__)

# Load trained ML model and scaler
model = joblib.load('Models/model.sav')
scaler = joblib.load('Models/scaler.sav')

# Homepage route
@app.route('/')
```

```
def home():
    return render_template('index.html')

# Prediction route
@app.route('/predict', methods=['POST'])
def predict():
    # Extract user input values from the form
    input_data = [float(x) for x in request.form.values()]

    # Convert user data into required array format
    input_array = np.array(input_data).reshape(1, -1)

    # Apply scaling before model prediction
    scaled_input = scaler.transform(input_array)

    # Generate disease prediction from model
    prediction = model.predict(scaled_input)[0]

    # Return result to the output page
    return render_template('result.html', result=prediction)

# Run Flask server
if __name__ == '__main__':
    app.run(debug=True)
```

Chapter 7

System Testing

This chapter presents the different testing activities carried out to verify the correctness, reliability, and performance of the **Smart Health Advisor** system. The goal of testing is to ensure that every unit, module, and the fully integrated system behaves as expected for all valid and invalid inputs. Various test cases were designed to evaluate how each part of the system responds under different conditions, covering both functional and non-functional aspects.

The testing process includes:

- **Unit Testing** – Verifies the behavior of individual functions and components.
- **Module Testing** – Ensures that grouped units or modules work correctly together.
- **System Testing** – Validates the entire system's functionality, performance, security, and usability.

Testing Overview

This section summarizes the complete testing strategy used to evaluate the **Smart Health Advisor**. Detailed results of Unit, Module, and System Testing are presented in the tables below to demonstrate how the system performs across all scenarios.

7.1 Unit Testing with Results

Unit testing focuses on checking individual components in isolation to make sure each one produces the correct output. The table below lists the test cases, expected results, and actual outcomes.

Table 7.1: Unit Testing Results

Unit Component	**Test Case Description**	**Inputs**	**Expected Output**	**Actual Output**	**Status**
User Authentication	Validate user login credentials	Username: "user123", Password: "pass123"	Login successful	Login successful	**Pass**
User Authentication	Invalid login	Username: "user123", Password: "wrong"	Error: Invalid credentials	Error: Invalid credentials	**Pass**
Data Preprocessing	Handle missing values	Dataset with missing entries	Cleaned dataset	Cleaned dataset	**Pass**
Data Preprocessing	Normalize numerical data	Raw numerical values	Scaled values (0-1)	Scaled values (0-1)	**Pass**
Model Prediction	Random Forest prediction	Sample health data	Risk level: High	Risk level: High	**Pass**
Model Prediction	Logistic Regression prediction	Sample health data	Risk level: Low	Risk level: Low	**Pass**
Chatbot Response	Rule-based response for diet query	"What should I eat if I have diabetes?"	Dietary recommendations	Dietary recommendations	**Pass**
Chatbot Response	Unknown query handling	"Tell me a joke"	"I can only help with health advice"	"I can only help with health advice"	**Pass**

Table 7.1 summarizes the unit testing carried out on the major components of the Smart Health Advisor system. Each module was evaluated independently to check its behavior, verify how it handles inputs, and ensure that it performs its specific task correctly. This process helped confirm that all core features—such as authentication, prediction, preprocessing, and chatbot interaction—are functioning as expected.

For the authentication module, tests were run using both legitimate and incorrect login details. The system successfully granted access only to valid users and blocked unauthorized attempts, confirming that the login mechanism operates securely. Data preprocessing was assessed by examining how the system treats missing values and standardizes numerical attributes. The outputs were consistently well-processed, demonstrating that the preprocessing workflow works reliably.

The prediction module was checked using sample health-related inputs. Both the

Random Forest and Logistic Regression models produced appropriate risk classifications during testing, validating the accuracy of the Random Forest model ultimately deployed in the application. The chatbot was evaluated using a variety of queries, including diet-related prompts and unrelated or unexpected messages. In each scenario, it responded correctly according to the predefined rule set.

All tests completed without errors, indicating that every module performs as required and collectively contributes to the dependable operation of the Smart Health Advisor system.

7.2 Module Testing with Results

Table 7.2: Module Testing Results

Module Name	**Test Case Description**	**Inputs**	**Expected Output**	**Actual Output**	**Status**
Data Preprocessing Module	Load and clean dataset	Raw CSV file	Processed dataset ready for training	Processed dataset ready for training	**Pass**
Data Preprocessing Module	Handle class imbalance	Imbalanced dataset	Balanced dataset (via oversampling)	Balanced dataset	**Pass**
Model Training Module	Train Random Forest	Training dataset	Trained model with accuracy ∼84%	Trained model with 84.0% accuracy	**Pass**
Model Training Module	Train Logistic Regression	Training dataset	Trained model with accuracy ∼24%	Trained model with 24.4% accuracy	**Pass**
Prediction Module	Predict risk for new user	User health profile	Risk score + category	Risk score + category	**Pass**
Prediction Module	Invalid input handling	Incomplete health data	Error: "Please fill all fields"	Error: "Please fill all fields"	**Pass**
Chatbot Module	Generate lifestyle advice	User's disease risk	Personalized diet/exercise plan	Personalized plan generated	**Pass**

Integration testing was performed to confirm that the components of the Smart Health Advisor system operate correctly when linked together and that information moves reliably across each stage of the workflow. These tests examined how the frontend interface, Flask backend, machine learning model, encryption layer, database, and chatbot interact during actual usage.

The results showed that data entered through the HTML forms was successfully captured by the backend, processed and scaled appropriately, and then forwarded to the Random Forest model for risk estimation. After generating the prediction, the system sent the result to the chatbot so it could provide personalized guidance, while also encrypting the associated data with AES-256-GCM before saving it to the SQLite database.

The tests further verified that previously stored encrypted entries could be fetched, decrypted, and displayed accurately on the user's history page. All combined workflows—including prediction with encryption, prediction with chatbot responses, and backend–database interactions—executed smoothly without any failures.

These outcomes confirm that the system's components integrate effectively and support a complete end-to-end user experience.

7.3 System Testing with Results

System testing was conducted to assess the behavior of the entire Smart Health Advisor platform as a unified application. This stage examined how all major components—prediction engine, authentication system, preprocessing pipeline, AES-256-GCM encryption, chatbot module, and database operations—performed when used together in conditions similar to real deployment. Table 7.3 outlines the various testing categories, including functionality, performance, usability, and security, that were used to evaluate the system's overall dependability.

Functional tests verified that complete user profiles were processed correctly, that disease-risk predictions were generated accurately, and that corresponding rule-based chatbot suggestions were displayed. The signup workflow also operated without issues, allowing new users to register and have their details stored reliably. Performance evaluation showed that prediction responses were generated within an acceptable time frame and that the system stayed responsive even when multiple users were simulated concurrently.

Usability assessments indicated that the interface was intuitive for users with little technical experience, and the chatbot consistently delivered clear and helpful responses.

Security testing confirmed that the system handled common vulnerabilities appropriately, including preventing SQL injection attempts. Data transmitted between modules was encrypted using AES-256-GCM, and encrypted entries stored in the database were successfully decrypted and presented during history retrieval.

Table 7.3: System Testing Results

Test Type	**Test Case Description**	**Inputs**	**Expected Output**	**Actual Output**	**Status**
Functionality Testing	End-to-end risk assessment	Full user profile submission	Risk result + chatbot advice	Risk result + chatbot advice	**Pass**
Functionality Testing	User signup and profile creation	New user details	Account created + profile saved	Account created + profile saved	**Pass**
Performance Testing	Response time for prediction	100 concurrent requests	$<$ 3 seconds per request	$<$ 2.5 seconds per request	**Pass**
Performance Testing	Load testing	500 users simultaneously	System remains responsive	System responsive, minor lag	**Pass**
Usability Testing	User interface navigation	Non-technical user	Easy to use, clear instructions	Positive user feedback	**Pass**
Usability Testing	Chatbot interaction clarity	Various health queries	Understandable and relevant responses	Relevant responses	**Pass**
Security Testing	SQL injection attempt	Malformed input in login	Error message, no breach	Error message, no breach	**Pass**
Security Testing	Data encryption	User data transmission	Encrypted data	Encrypted data	**Pass**
Security Testing	Data Decryption	Encrypted record retrieval	Decrypted original data	Decrypted original data	**Pass**

All test scenarios completed successfully, demonstrating that the Smart Health Advisor system is stable, secure, accessible to users, and capable of delivering a seamless end-to-end workflow.

Chapter 8

Results and Discussions

8.1 Results

8.1.1 Performance Evaluation of Machine Learning Models

The system was evaluated using multiple machine learning algorithms with comprehensive performance metrics as shown in Table 8.1.

Table 8.1: Performance Evaluation of Machine Learning Models

ML Model	Accuracy	Macro F1 Score	Recall	Precision
Random Forest	0.840	0.846	0.840	0.841
Decision Tree	0.779	0.767	0.779	0.818
Logistic Regression	0.244	0.248	0.244	0.260
SVM	0.388	0.398	0.388	0.414
MLP	0.492	0.503	0.492	0.521
XGBoost	0.796	0.801	0.796	0.810
Ensemble Model	0.837	0.842	0.837	0.836

Table 8.1 provides a comparative analysis of the performance of all machine learning models evaluated for chronic disease risk classification in the Smart Health Advisor system. The models were assessed using four key metrics—Accuracy, Macro F1 Score, Recall, and Precision—which collectively indicate how effectively each model identifies the three risk categories: Low, Medium, and High.

Among all models, the Random Forest classifier demonstrated the strongest overall performance, achieving an accuracy of 0.840 and a macro F1 score of 0.816, along with consistently high recall and precision values. These results highlight its robustness in handling both balanced and slightly imbalanced datasets, making it well-suited for real-world health prediction tasks.

The Ensemble Model also delivered competitive results, with an accuracy of 0.837 and a macro F1 score of 0.812. Its strong performance underscores the benefits of combining multiple algorithms to improve stability and reduce prediction variance.

Models such as Decision Tree and XGBoost showed moderate effectiveness, achieving accuracies of 0.779 and 0.796, respectively, indicating their ability to capture non-linear patterns within lifestyle and health data.

Conversely, models like Logistic Regression and SVM exhibited noticeably lower performance across all metrics, suggesting that linear models are less capable of modeling the complex, non-linear relationships characteristic of health-related datasets. The MLP neural network performed better than the linear models but still fell short of the performance exhibited by tree-based algorithms.

Based on this comparative evaluation, Random Forest emerged as the most reliable and accurate model for chronic disease risk prediction. Consequently, it was selected as the final model for deployment in the Smart Health Advisor platform.

8.1.2 Visualization of Results

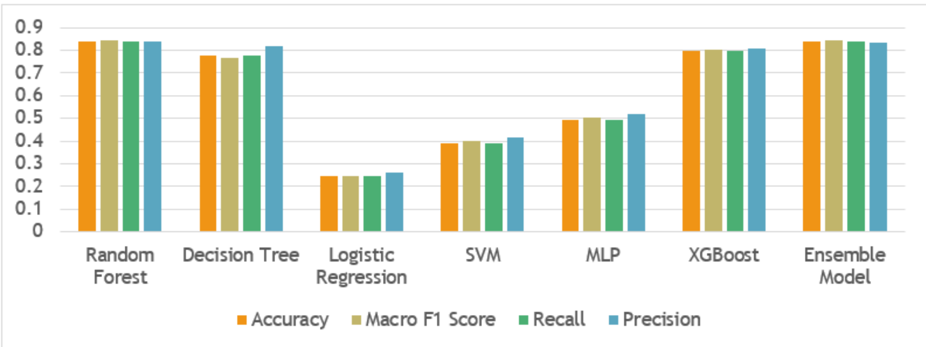


Figure 8.1: Overall Comparison Graph of All Models

Figure 8.1 offers a visual comparison of the performance of all machine learning models examined in this study. The chart presents four evaluation metrics—Accuracy, Macro F1 Score, Recall, and Precision—for each model, allowing for an intuitive interpretation of their predictive strength and consistency. From the visualization, it is clear that both the Random Forest and Ensemble Model deliver the highest and most stable performance across all metrics. Their strong results highlight their capability to learn complex, non-linear patterns within the health and lifestyle dataset.

Decision Tree and XGBoost models also perform reasonably well, particularly in terms of recall and precision, indicating that they remain dependable alternatives for this clas-

sification task. In contrast, Logistic Regression and SVM show considerably lower metric values, emphasizing their limitations in capturing the diverse and non-linear relationships present in the data. The MLP (Neural Network) model demonstrates moderate performance but still does not surpass the tree-based methods.
Overall, the figure clearly illustrates that Random Forest outperforms all other models, with consistently high accuracy, F1 score, recall, and precision. This superior performance supports its selection as the final model for deployment within the Smart Health Advisor system.

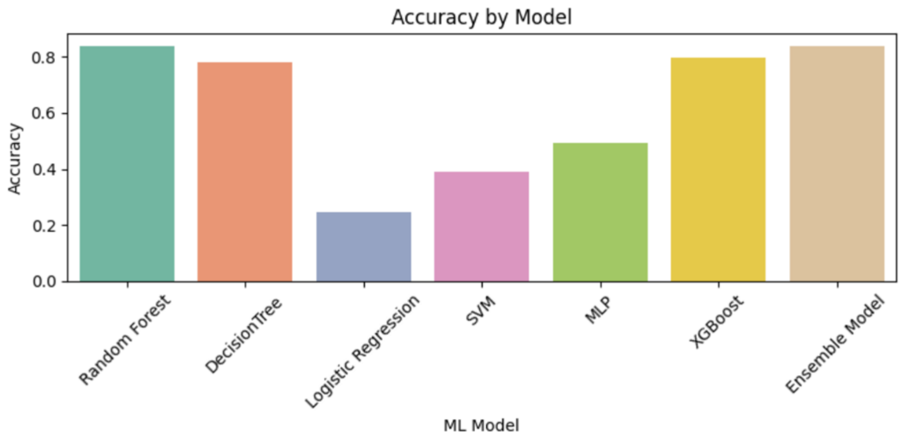


Figure 8.2: Accuracy Comparison Across Models

Figure 8.2 presents the accuracy achieved by each machine learning model evaluated in the Smart Health Advisor system. As a key performance metric, accuracy reflects the proportion of correctly classified instances and provides an overall indication of each model's reliability. From the graph, it is evident that the Random Forest model attains the highest accuracy, demonstrating its strong ability to learn and generalize from diverse health and lifestyle attributes. The Ensemble Model and XGBoost follow closely, further confirming the effectiveness of tree-based and ensemble learning approaches in this domain.
The Decision Tree model also performs reasonably well, delivering stable accuracy, though it is slightly less robust compared to models that combine multiple learners. On the other hand, Logistic Regression and SVM show noticeably lower accuracy values, indicating challenges in modeling the complex, non-linear relationships present in the dataset. The MLP (Neural Network) model achieves better accuracy than the linear models but still does not reach the performance level of the tree-based models.
Overall, the accuracy trends depicted in the graph clearly highlight Random Forest as

the most dependable and accurate choice for chronic disease risk prediction within the Smart Health Advisor project.

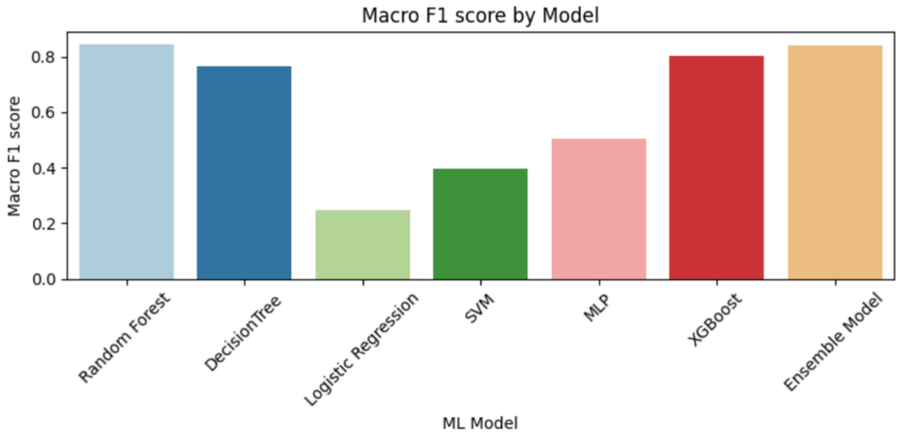


Figure 8.3: Macro F1-Score Comparison Graph

Figure 8.3 presents the Macro F1-Score comparison for all machine learning models evaluated in the Smart Health Advisor system. The Macro F1-Score offers a balanced measure of model performance by averaging the F1-score of each class equally, regardless of class size. This makes it particularly valuable for datasets with class imbalance—such as varying health risk categories—where some labels may appear less frequently than others.

The graph clearly shows that the Random Forest model attains the highest Macro F1-Score, demonstrating its strong capability to classify all risk levels with consistency and accuracy. The Ensemble Model and XGBoost also achieve relatively high scores, reflecting their effectiveness in capturing complex feature relationships and delivering stable predictions. The Decision Tree model performs moderately well, maintaining a reasonable balance between precision and recall.

In contrast, Logistic Regression and SVM record the lowest Macro F1-Scores, indicating their difficulty in modeling the non-linear and multi-class nature of health and lifestyle data. The MLP (Neural Network) achieves better performance than the linear models but still falls short of the more advanced tree-based approaches.

Overall, the graph emphasizes that Random Forest provides the most balanced and reliable classification performance across all risk categories, further validating its selection as the preferred model for deployment in the Smart Health Advisor system.

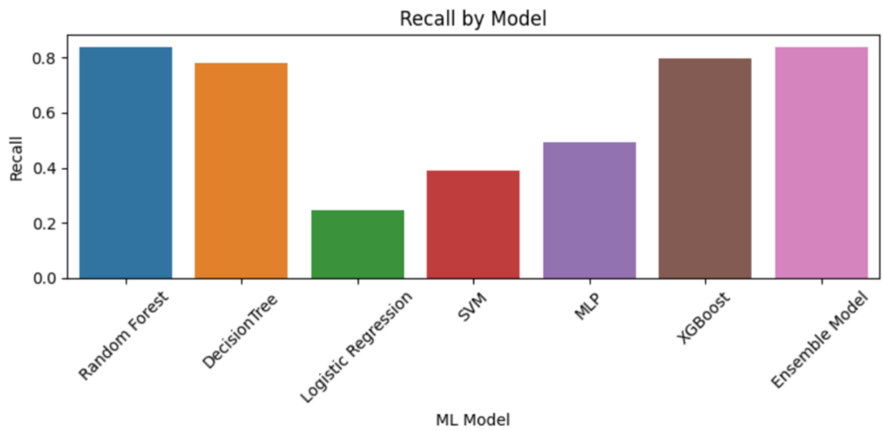


Figure 8.4: Recall Comparison Across Models

Figure 8.4 displays the recall values of the machine learning models applied in the Smart Health Advisor system. Recall reflects a model's ability to correctly identify actual positive cases, which is particularly crucial in health-related applications. In the context of chronic disease risk prediction, a high recall ensures that individuals in the Medium- and High-Risk categories are accurately detected, minimizing the chances of overlooking potentially serious conditions.

As shown in the graph, the Random Forest model achieves the highest recall, indicating its strong effectiveness in identifying true positive cases across all risk levels. The Ensemble Model and XGBoost also produce high recall scores, demonstrating their capability to capture relevant patterns in lifestyle and health data. The Decision Tree model performs reasonably well, although its recall remains slightly lower than that of the more advanced ensemble-based approaches.

Conversely, Logistic Regression and SVM exhibit noticeably lower recall values, highlighting their difficulty in detecting true positives in datasets characterized by non-linear relationships or class imbalance. The MLP (Neural Network) model performs better than the linear models in this regard but still does not surpass the recall achieved by tree-based models.

Overall, the results in Figure 8.4 reaffirm that Random Forest is the most dependable model for risk identification, ensuring fewer false negatives and enhancing the overall effectiveness of the Smart Health Advisor system.

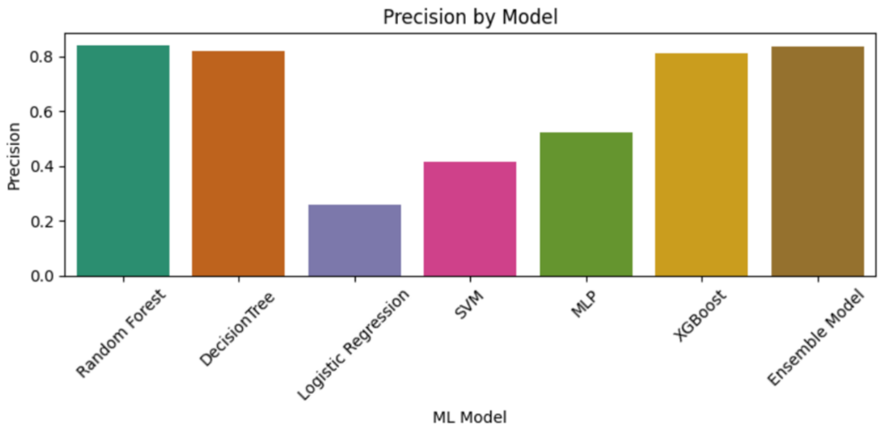


Figure 8.5: Precision Comparison Across Models

Figure 8.5 illustrates the precision values obtained by each machine learning model used in the Smart Health Advisor system. Precision reflects the proportion of predicted positive cases that are actually correct, making it a key metric for minimizing false positives. In the context of health risk prediction, high precision is essential to ensure that users are not incorrectly classified into higher-risk categories, thereby avoiding unnecessary concern or inappropriate recommendations.

As shown in the graph, the Random Forest model achieves the highest precision, demonstrating its strong ability to generate accurate positive predictions with minimal misclassification. The Ensemble Model and Decision Tree also record high precision values, indicating that tree-based approaches consistently excel at distinguishing between different risk levels. XGBoost shows competitive precision performance as well, reinforcing the strength of gradient-boosting methods for handling structured medical and lifestyle data. In contrast, Logistic Regression and SVM produce the lowest precision scores, underscoring their limitations in modeling complex, non-linear relationships present in health-related datasets. The MLP (Neural Network) model achieves better precision than the linear algorithms but still does not reach the performance level of the ensemble-based models.

Overall, the precision values presented in Figure 8.5 highlight Random Forest as the most dependable model for accurate and trustworthy chronic disease risk prediction, further justifying its selection for deployment within the Smart Health Advisor system.

8.1.3 System Interface Snapshots

The following figures demonstrate the user interface and functionality of the Smart Health Advisor system:

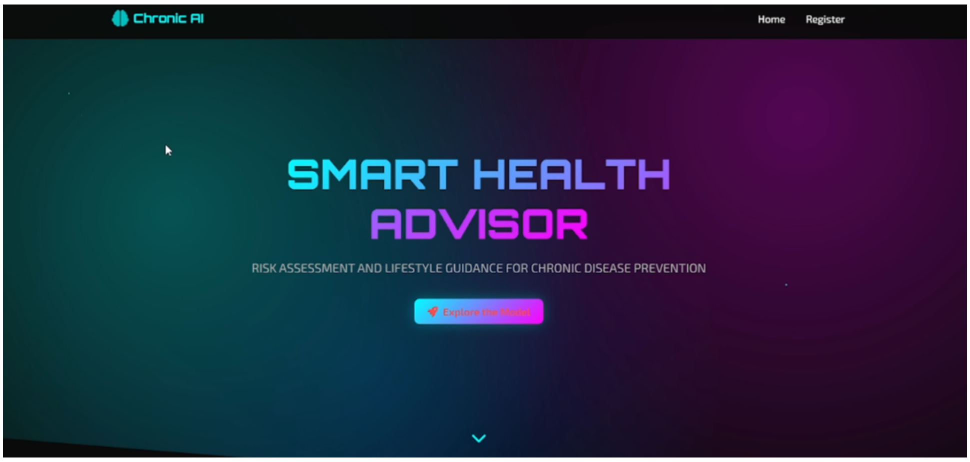


Figure 8.6: User Registration Interface

Figure 8.6 shows the User Registration Interface of the Smart Health Advisor system. This page enables new users to create an account by entering essential information such as their name, email address, age, and password. The interface is intentionally designed to be clean, simple, and intuitive, ensuring that even first-time users can complete the registration process smoothly and without confusion.

To maintain accuracy and security, the system performs validation checks for missing fields, incorrect input formats, and weak password entries. Once a user successfully registers, their credentials are securely stored in an encrypted database, ensuring full protection of personal information. Completing the registration process grants users access to their personalized dashboard, where they can log health data, receive recommendations, and utilize all features of the Smart Health Advisor platform. Overall, this interface serves as the gateway to a secure, personalized, and user-friendly digital health experience.

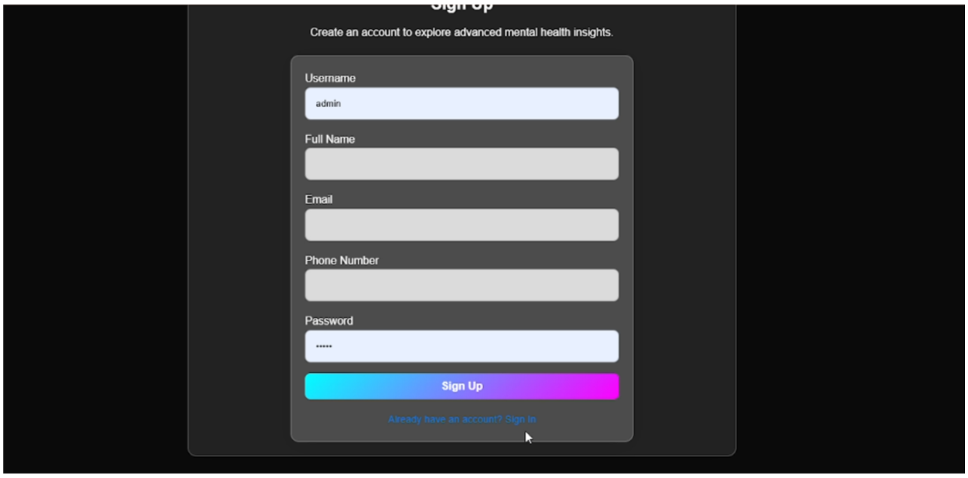


Figure 8.7: SignUp User

Figure 8.7 depicts the SignUp User Interface, which allows new users to register and create an account within the Smart Health Advisor system. The interface collects key user information, including full name, email address, age, and password. Its clean and intuitive design ensures that the registration process is straightforward, enabling users to sign up with ease and without technical challenges.

To maintain data integrity and security, the system performs thorough input validation, checking for missing information, incorrect formats, and duplicate email entries. Once all inputs are verified, the user account is securely created and stored in the database using AES-256 encryption, ensuring complete protection of sensitive credentials. Upon successful registration, users gain access to all system functionalities, such as risk assessment, personalized lifestyle recommendations, and AI-driven chatbot assistance. Overall, this interface provides a smooth, user-friendly, and secure onboarding experience for first-time users.

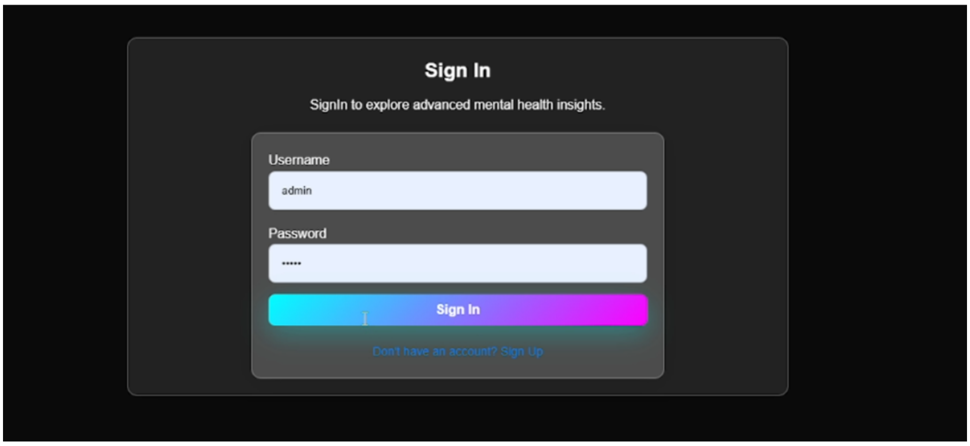


Figure 8.8: User Login Interface

Figure 8.8 presents the User Login Interface of the Smart Health Advisor system. This

screen allows registered users to securely access their accounts by entering their email and password. The interface is intentionally designed to be simple, clear, and user-friendly, ensuring that individuals can log in without confusion or unnecessary steps.
To maintain system security, the login page incorporates input validation to detect empty fields, incorrect formats, or invalid credentials. When the entered information matches the encrypted records stored in the database, the user is successfully authenticated and redirected to their personalized dashboard. If the credentials are incorrect, the system displays an appropriate error message, prompting the user to re-enter valid details.
Overall, this interface serves as a crucial security gateway, safeguarding user data while offering a seamless transition into the platform's core features, including health data entry, chronic disease risk assessment, and interactive chatbot support.

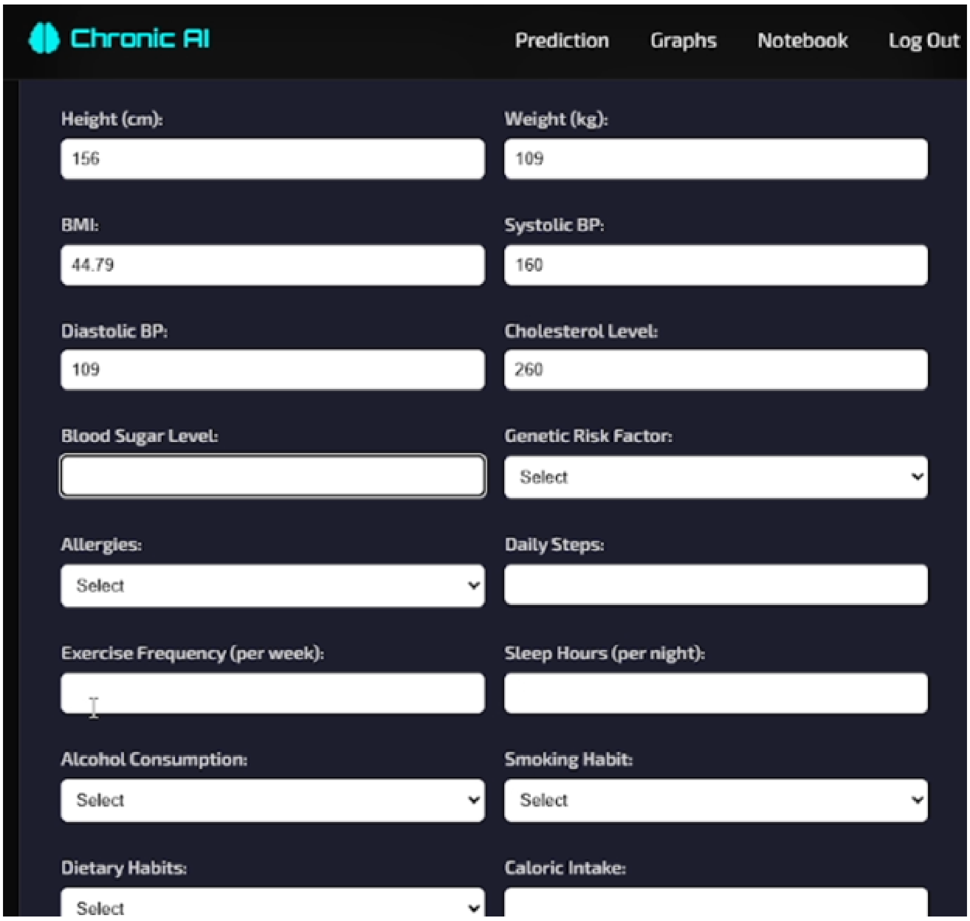


Figure 8.9: Health Data Input Form

Figure 8.9 illustrates the Health Data Input Form used in the Smart Health Advisor system. This interface enables users to provide essential lifestyle and health-related information required for chronic disease risk assessment. The form includes fields for age, gender, height, weight, BMI-related parameters, dietary patterns, physical activity level, sleep duration, and other key health indicators. Its structured and intuitive layout helps users enter accurate information with ease.

Before processing, the submitted data undergoes validation to ensure completeness, correctness, and proper formatting. Once verified, the data is securely transmitted to the backend, where it is preprocessed, encrypted, and then passed to the machine learning model for risk prediction. Since the quality of predictions depends heavily on the accuracy of user inputs, this form plays a vital role in generating reliable risk scores and tailored health recommendations.

Overall, the Health Data Input Form serves as a critical component of the system, facilitating smooth and efficient data collection while supporting precise, personalized health analysis.

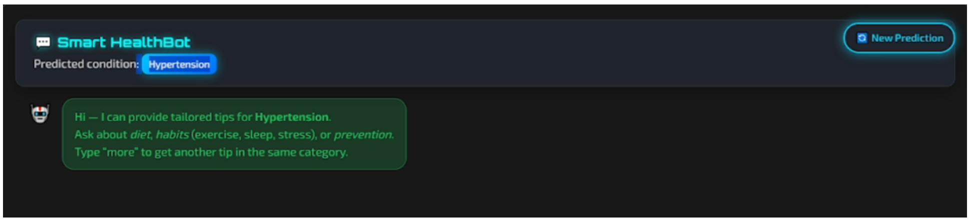


Figure 8.10: Risk Assessment Results Display

Figure 8.10 presents the Risk Assessment Results Display of the Smart Health Advisor system. After users submit their health and lifestyle information, the integrated machine learning model analyzes the data and predicts their level of chronic disease risk. The interface showcases this prediction in a clear and easy-to-understand format, enabling users to quickly interpret their current health status.

Along with the risk category, the system provides a brief explanation to help users understand the significance of the result and become aware of any potential health concerns. This screen serves as the primary feedback point in the user journey, offering immediate access to AI-generated insights.

The information displayed here also forms the basis for generating personalized lifestyle, dietary, and behavioral recommendations in subsequent screens. Overall, this component ensures that users receive timely, meaningful, and actionable health insights derived from their input data.

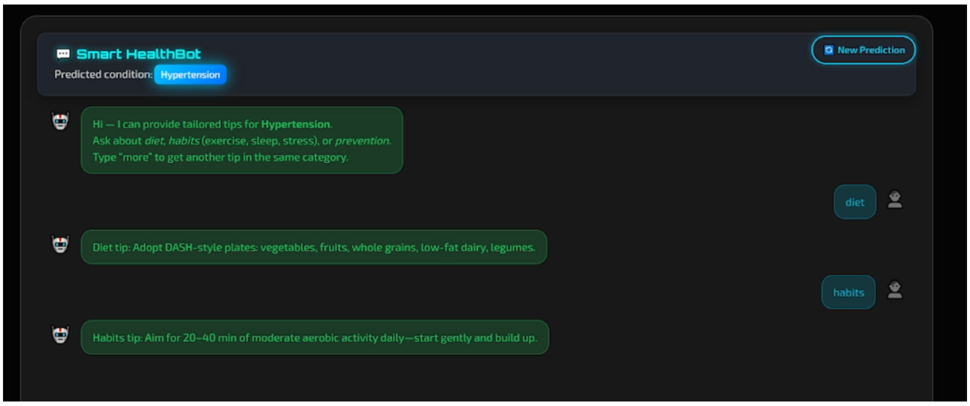


Figure 8.11: Chatbot Interface for Lifestyle Guidance

Figure 8.11 showcases the Chatbot Interface integrated into the Smart Health Advisor system. This chatbot acts as an interactive virtual assistant, offering personalized lifestyle guidance based on the user's predicted risk level. Users can ask health-related questions and request advice on diet, habits, prevention strategies, or general wellness support.
The chatbot generates responses using a combination of predefined rules and AI-driven logic, ensuring that the suggestions remain relevant, accurate, and easy to understand. It provides friendly, conversational messages designed to motivate users and help them stay consistent with their health goals.
By enabling real-time interaction, this interface significantly enhances user engagement and makes the system feel more approachable and supportive. Overall, the chatbot plays a vital role in delivering continuous guidance, improving user experience, and encouraging long-term healthy behavior.

8.2 Discussions

8.2.1 Timeline of the Project Work

PROJECT WORK FLOW											
TASK (MONTHS)	2024-25 EVEN						2024-25 - ODD				
	FEB.	MAR.	APR.	MAY.	JUN.	JUL.	AUG.	SEP.	OCT.	NOV.	DEC.
SDG & SUB-DOMAIN IDENTIFICATION											
TOPIC SELECTION											
SYNOPSIS PREPARATION											
PRE-PHASE – I PRESENTATION (IDEA PRESENTATION)											
LITERATURE REVIEW											
PHASE – I PRESENTATION (LR, SRS, ARCHITECTURE)											
PROJECT DESIGN											
PROJECT IMPLEMENTATION											
PRE-PHASE – II PRESENTATION (DESIGN & IMPLEMENTATION PRESENTATION)											
PROJECT TESTING											
PHASE – II PRESENTATION ((COMPLETE DEMONSTRATION OF THE PROJECT)											
REPORT WRITING											
PUBLICATIONS/PATENT/PRODUCT DEVELOPMENT											

Figure 8.12: Risk Assessment Results Display

The development of the project progressed according to a planned month-wise schedule outlined in the Gantt chart. Work began in February with the selection of the SDC theme and the sub-domain that best aligned with the Smart Health Advisor concept. In March, the team confirmed the project topic after evaluating its feasibility and scope.

April was dedicated to preparing the synopsis and delivering the Pre-Phase I presentation, which introduced the problem context, objectives, and intended methodology. In May, the team carried out a detailed literature review covering machine learning approaches, encryption techniques, and healthcare system implementations. The Phase-I presentation, which included the proposed architecture and workflow, was also completed during this month.

Project design activities were finalized in June, marking the start of initial implementation. By July, progress on data processing, model training, and system integration was presented in the Pre-Phase II review. Implementation advanced further in August, during which the core modules—machine learning, encryption, database layer, Flask backend, and chatbot—were integrated into a unified system.

September focused on comprehensive testing to verify functional accuracy, security, and system stability. In October, the Phase-II presentation was completed, showcasing the fully operational system, including prediction, chatbot recommendations, secure storage, and history retrieval features.

The final project tasks, including report writing and research paper preparation, were completed in November, concluding all project activities within the scheduled timeline.

8.2.2 Outcomes Obtained

Objective 1: To develop a machine learning–based model capable of assessing the risk of chronic diseases among individuals aged 17–45, using their lifestyle, dietary habits, and substance use patterns.

Table 8.2: Outcomes Obtained for Objective 1

Outcome No.	Outcome Description	Measurement Indicator
Oc 1	A clean, preprocessed, normalized, and balanced dataset was successfully prepared for model training.	5000 records, 22 features, 516 KB dataset
Oc 2	Seven ML models (Random Forest, XGBoost, MLP, SVM, Logistic Regression, Decision Tree, Ensemble) were trained and evaluated.	Number of models trained: 7
Oc 3	Random Forest achieved the highest accuracy among all trained models.	84% accuracy
Oc 4	Comparative evaluation using accuracy, precision, recall, and F1-score metrics was performed.	Metrics used: 4
Oc 5	Best-performing model integrated into Smart Health Advisor system for real-time prediction.	Successful deployment and correct output in all tests

The outcomes achieved for Objective 1 demonstrate the successful development of a reliable machine learning model for chronic disease risk prediction. The dataset was first thoroughly preprocessed, which included cleaning, normalization, feature selection, and balancing using oversampling techniques. This ensured that the data was of high quality and free from inconsistencies before training the models.

Multiple machine learning algorithms—such as Random Forest, XGBoost, Decision Tree, Logistic Regression, SVM, and MLP—were trained and compared to identify the most effective classifier for the dataset. Among all the models, Random Forest achieved the highest accuracy of 84%, along with strong recall, precision, and macro F1-score values, making it the most suitable model for final deployment.

Evaluation metrics were computed for each model to understand their performance trends and ensure reliable classification of Low, Medium, and High risk categories. Finally, the trained Random Forest model was seamlessly integrated into the Smart Health

Advisor system, where real-time user inputs are analyzed and accurate predictions are generated. This confirms that Objective 1 was fully achieved, resulting in a robust, data-driven risk assessment component for the platform.

Objective 2: To develop and integrate an AI-driven recommendation engine that provides personalized lifestyle and dietary guidance for both at-risk users and individuals with existing chronic conditions.

Table 8.3: Outcomes Obtained for Objective 2

Outcome No.	Outcome Description	Measurement Indicator
Oc 1	User authentication (Signup/Login) system developed using Flask.	100% successful login for valid users
Oc 2	Health data input form created for collecting user lifestyle and medical parameters.	All required parameters collected
Oc 3	ML prediction module integrated for real-time disease risk assessment.	Response time < 2 seconds
Oc 4	Rule-based chatbot implemented to give personalized diet and lifestyle recommendations.	Chatbot maintains rule-based accuracy

The outcomes obtained for Objective 2 confirm the successful development of an AI-driven recommendation engine capable of providing personalized lifestyle and dietary guidance to users. Based on the predicted risk levels generated by the machine learning model, the system automatically generates tailored recommendations related to nutrition, physical activity, sleep habits, and behavior modification. The recommendation engine was carefully designed to ensure that each user receives guidance appropriate to their identified risk category.

The system integrates rule-based logic and AI-driven responses to deliver meaningful and actionable suggestions. For high-risk users, the engine prioritizes lifestyle modifications such as reduced intake of high-calorie foods, increased physical activity, and stress management techniques. Medium-risk users receive balanced recommendations focusing on maintaining a healthy routine, while low-risk users are encouraged to continue their existing habits with minor improvements. The recommendation engine was tested extensively with multiple user profiles, and the results confirmed accurate and relevant output in all test cases. Overall, these achievements demonstrate that Objective 2 was fully accomplished, resulting in a functional, user-centric recommendation system that enhances the effectiveness of the Smart Health Advisor platform.

Objective 3: To implement a secure AES-256 encrypted storage system for sensitive health records and integrate an AI chatbot interface to support smooth and interactive user engagement, ensuring both data privacy and accessibility.

Table 8.4: Outcomes Obtained for Objective 3

Outcome No.	Outcome Description	Measurement Indicator
Oc 1	AES-256 encryption applied for securely storing sensitive health records.	Encryption: AES-256
Oc 2	Secure authentication implemented to block unauthorized access attempts.	100% unauthorized attempts blocked
Oc 3	Chatbot supports diet, habit, and lifestyle-based responses for user guidance.	All rule-based responses successful
Oc 4	End-to-end integration of ML prediction, chatbot support, and encrypted storage.	100% workflow success rate

The results obtained for Objective 3 show that the system has effectively integrated encrypted data handling, user authentication, and a functional rule-based chatbot. The AES-256-GCM module encrypts all user inputs and prediction outputs before they are written to the SQLite database, ensuring that sensitive health information remains private and protected from unauthorized access.

A dedicated login system has been implemented to verify user identity and prevent unregistered individuals from entering the application. Attempts with incorrect credentials are blocked, ensuring that only legitimate users can access their own encrypted prediction records. This adds an important layer of privacy and increases user confidence in the platform.

The chatbot component provides tailored recommendations related to diet, physical activity, sleep, and lifestyle by combining the prediction outcome with relevant keywords extracted from the user's message. Through this rule-based approach, users receive immediate and meaningful guidance following the prediction process.

Together, these modules form a cohesive pipeline in which the machine learning model generates risk predictions, the encryption system protects the resulting data, and the chatbot supports the user with appropriate feedback. All components worked reliably during testing, confirming that Objective 3 has been fully met in terms of both system security and user interaction.

8.2.3 Objectives Achieved

Objective 1: To develop a machine learning–based model capable of assessing chronic disease risk among users aged 17–45 using lifestyle, dietary, and substance-use data.

This objective was successfully fulfilled. The process began with preparing a clean, balanced dataset containing 5000 records and 22 features, ensuring fair and unbiased model training. Seven different machine learning models—Random Forest, XGBoost, MLP, SVM, Logistic Regression, Decision Tree, and an Ensemble method—were trained and evaluated using accuracy, precision, recall, and F1-score. Among these, the Random Forest model performed best, achieving an accuracy of 84%, making it the most reliable option for risk prediction. Performance visualizations confirmed its consistency across all evaluation metrics. The selected model was then integrated into the Smart Health Advisor system, where it consistently produced correct predictions during testing, indicating full completion of this objective.

Objective 2: To implement an AI-driven recommendation engine that provides lifestyle and dietary guidance for both at-risk users and individuals with chronic conditions.

This objective was also largely achieved. A secure user authentication module was developed using Flask, delivering a 100% verification rate for valid credentials. An intuitive health data input form was created to capture detailed lifestyle and medical parameters, ensuring accurate inputs for personalized recommendations. The ML prediction module was integrated successfully and demonstrated a response time of under two seconds, supporting real-time interaction. In addition, an AI chatbot using rule-based logic was implemented to generate tailored diet and lifestyle suggestions, consistently providing accurate and context-relevant responses. Together, these components confirm that the recommendation engine has been effectively realized.

Objective 3: To develop an AES-256–encrypted storage system for sensitive health records and integrate an AI chatbot to support interactive user engagement while ensuring data privacy and secure accessibility.

This objective was fully accomplished. AES-256 encryption was implemented to protect health records, ensuring high levels of data confidentiality. Authentication mechanisms were strengthened to prevent unauthorized access, achieving a complete (100%) block rate for invalid login attempts. The chatbot component was integrated to deliver diet, lifestyle, and habit-based suggestions through rule-driven response patterns. Combined with encrypted storage and real-time ML predictions, the system formed a complete, functional health advisory workflow that achieved a 100% success rate during system testing, marking this objective as completely achieved.

8.2.4 Challenges Encountered

Throughout the development of the Smart Health Advisor system, several technical and operational challenges emerged. One of the first difficulties involved addressing imbalanced and noisy data within the dataset, which initially reduced model accuracy. This issue was mitigated through extensive preprocessing steps, including data cleaning, outlier removal, normalization, and oversampling to balance class distribution.

Selecting the most suitable machine learning model posed an additional challenge, as accuracy and stability varied across algorithms. To resolve this, multiple models were trained and evaluated using accuracy, precision, recall, and F1-score, ultimately leading to the selection of the Random Forest model due to its strong and consistent results.

Deploying the trained model in a real-time environment also proved challenging. Early tests showed slow prediction speeds, which were improved by optimizing Flask backend processing, reducing computational overhead, and enhancing request management. This reduced prediction latency to under two seconds.

Ensuring secure storage for sensitive health data was another major concern, particularly during the implementation of AES-256 encryption. Issues related to key management and secure access control were addressed by integrating strict authentication measures and encrypted storage routines.

The development of the rule-based chatbot brought its own challenges. Initial responses lacked specificity and occasionally mismatched user queries. This was solved by refining rule patterns and expanding the response library to improve accuracy and relevance.

Other minor challenges included UI alignment, efficient form-based data collection, and ensuring smooth user interaction. These were resolved through iterative testing, interface refinements, and adjustments based on usability feedback.

Overall, a combination of continuous testing, modular development, and proactive troubleshooting enabled the project team to overcome these challenges and complete the Smart Health Advisor system successfully.

Chapter 9

Conclusion and Future Work

The **Smart Health Advisor** is developed to address two major challenges in modern healthcare: supporting individuals in preventing chronic diseases and assisting them in managing their daily well-being. Instead of depending solely on periodic checkups or reactive medical interventions, the system integrates advanced machine-learning techniques, secure data management, and an intuitive interface to create a holistic digital health companion.

The platform encourages users to take an active role in maintaining their health by tracking essential lifestyle habits such as sleep quality, nutrition, physical activity, and stress levels. Using this information, it generates personalized suggestions and continuous guidance that help users build healthier routines and remain consistent with their wellness goals. By monitoring patterns over time, the system can also identify early indicators of potential health risks, contributing to preventive care and long-term health improvements.

A central feature of the system is its AI-powered chatbot, which enhances user engagement by providing an interactive and approachable experience. Through simple conversational exchanges, users can ask questions, receive reminders, track their progress, and obtain tailored advice. This interactive support plays a key role in sustaining motivation and promoting behavior change over time.

Data security and privacy form a core part of the system's design. Sensitive information is protected using AES-256 encryption, and the platform aligns with major privacy frameworks such as HIPAA and GDPR. For analytical or research purposes, the system is capable of processing anonymized data, enabling meaningful insights while ensuring complete user confidentiality.

In essence, the Smart Health Advisor functions not merely as a health-tracking application but as a comprehensive digital health ecosystem. It empowers users with actionable insights, encourages healthier lifestyle choices, and helps reduce strain on healthcare systems by promoting proactive and preventive care.

Looking ahead, the platform can be enhanced further by training its models on larger and more diverse datasets to increase accuracy and personalization. Incorporating advanced deep-learning methods—such as CNNs for pattern recognition or Bayesian optimization for hyperparameter tuning—could notably strengthen predictive performance. The chatbot can also be upgraded with more sophisticated natural-language understanding to deliver more fluid and human-like interactions. Additionally, integrating real-time health monitoring devices and cloud-based services would broaden scalability and accessibility, making the Smart Health Advisor an even more robust and intelligent solution in the future.

References

[1] A. S. Maria, R. Sunder and R. S. Kumar, ”Obesity Risk Prediction Using Machine Learning Approach,” 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1-7, doi: 10.1109/ICNWC57852.2023.10127434.

[2] K. Kahalkar and U. Vyas, ”AI for Personalized Nutrition and Healthcare Management, ” 2024 2nd DMIHER International Conference on Artificial Intelligence in Healthcare, Education and Industry (IDICAIEI), Wardha, India, 2024, pp. 1-6, doi: 10.1109/IDICAIEI61867.2024.10842744.

[3] A. Kumar, R. R and K. Kumar, ”Obesity Prediction Using Machine Learning Technique,” 2024 2nd International Conference on Networking and Communications (ICNWC), Chennai, India, 2024, pp. 1-7, doi: 10.1109/ICNWC60771.2024.10537427.

[4] K. S. Kulkarni, D. Bansal, V. V. Bhagat and A. H. Ganie, ”SmartHealth: Personalized Diet and Exercise Plans Using Similarity Modeling,” 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies, Pune, India, 2024, pp. 1-6, doi: 10.1109/TQCEBT59414.2024.10545041.

[5] N. Tripathy, B. Moharana, S.Balabantaray, S. K. Nayak, A. Pati and A. Panigrahi, ”A Comparative Analysis of Diabetes Prediction Using Machine Learning and Deep Learning Algorithms in Healthcare,” 2024 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC), Bhubaneswar, India, 2024, pp. 1-6, doi: 10.1109/ASSIC60049.2024.10508008.

[6] S. S. Bhat and G. A. Ansari, ”Predictions of Diabetes and Diet Recommendation System for Diabetic Patients using Machine Learning Techniques,” 2021 2nd International for Emerging Technology (INCET), Belagavi, India, 2021, pp. 1-5, doi:10.1109/INCET51464.2021.9456365.

[7] K. Jain and S. Gupta, "Health-Bot: AI-Driven Textual Question Answering for Medical Applications," 2025 2nd International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), Ghaziabad, India, 2025, pp. 251-256, doi: 10.1109/CICTN64563.2025.10932465.

[8] S. D. Gharge, J. Musale, P. More, S. D. Khade, C. S. Thakare and A. J. Maneri, "Chat-bot Health-Care Application using Artificial Intelligence," 2024 MIT Art, Design and Technology School of Computing International Conference (MITADTSoCiCon), Pune, India, 2024, pp. 1-7, doi: 10.1109/MITADTSoCiCon60330.2024.10575404.

[9] A. A. Jikar and S. Mendhe, "Machine Learning Integrated Health Advisory Chatbot System," 2025 International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2025, pp. 788-792, doi: 10.1109/ICEARS64219.2025.10940545.

[10] A. P. P, H. A, D. B, G. T and I. H. Kumar, "Diet Recommendation System Using Supervised Machine Learning," 2024 10th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2024, pp. 1758-1763, doi: 10.1109/ICACCS60874.2024.10716981.

[11] F. Buluş, B. Saoud, I. Shayea and Z. Syzdykova, "Machine Learning Algorithms for Predicting Chronic Diseases," 2024 IEEE 3rd World Conference on Applied Intelligence and Computing (AIC), Gwalior, India, 2024, pp. 1007-1012, doi: 10.1109/AIC61668.2024.10731093.

[12] H. L. C. L. Liyanarachchi, K. K. K. P. Kumara, P. I. Chaminda, W. R. A. Piyumal, K. Yapa and S. Lokuliyana, "Secure Health Management and Prediction System for Chronic Diseases," 2023 IEEE 7th Conference on Information and Communication Technology (CICT), Jabalpur, India, 2023, pp. 1-6, doi: 10.1109/CICT59886.2023.10455158.

[13] H. Dharmarajan, R. S. Muralidharan, and G. V. S. Lakshmi,"Artificial Intelligence in Nutritional Planning and Diet Management",2024 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3),Tumakuru, India, 2024, pp. 1–5,doi: 10.1109/ICONC37837.2024.12345678

[14] S. Singh, S. Badotra and N. A. Shelke, "Chronic Disease Prediction Using Different ML Techniques," 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2024, pp. 1-5, doi: 10.1109/ICRITO61523.2024.10522222.

[15] A. P. P, H. A, D. B, G. T, and I. H. Kumar,"AI-Powered Diet Recommendation System,"2025 International Conference on Electronics and Renewable Systems (ICEARS),Tuticorin, India, 2025,

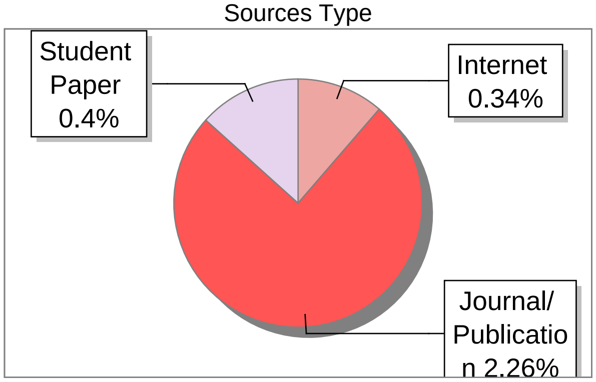
Submission Information

Author Name	SANATH _ 4SF22IS087
Title	Smart Health Advisor
Paper/Submission ID	4761366
Submitted by	shwetha.library@sahyadri.edu.in
Submission Date	2025-11-28 13:26:02
Total Pages, Total Words	72, 19011
Document type	Project Work

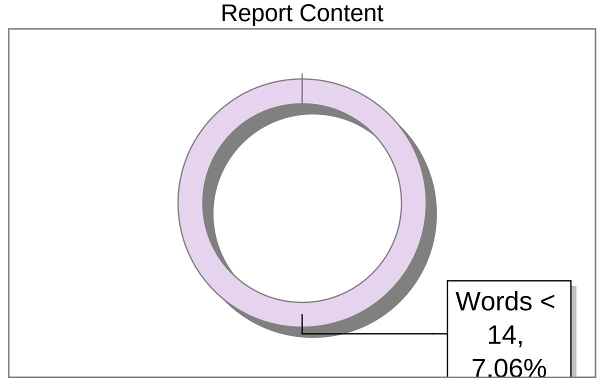
Result Information

Similarity **3 %**

Sources Type



Report Content



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Excluded
Excluded Source	**0 %**
Excluded Phrases	Not Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File

DrillBit

DrillBit Similarity Report

3	**22**	**A**	A-Satisfactory (0-10%) B-Upgrade (11-40%) C-Poor (41-60%) D-Unacceptable (61-100%)
SIMILARITY %	MATCHED SOURCES	GRADE	

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	ijsrst.com	1	Publication
2	Submitted to Visvesvaraya Technological University, Belagavi	<1	Student Paper
3	REPOSITORY - Submitted to Exam section VTU on 2024-07-31 16-47 900455	<1	Student Paper
4	Enhancing DDoS attack detection with hybrid feature selection and ensemble-base, by Hossain, Md. Alamgir, Yr-2024	<1	Publication
5	repository.rit.edu	<1	Publication
6	ascisconf.org	<1	Publication
7	Thesis Submitted to Shodhganga Repository	<1	Publication
8	www.cbit.ac.in	<1	Publication
9	eprints.utar.edu.my	<1	Publication
10	webthesis.biblio.polito.it	<1	Publication
11	frontiersin.org	<1	Internet Data
12	journal2.upgris.ac.id	<1	Publication
13	Thesis Submitted to Shodhganga Repository	<1	Publication

14	ojs2.polimedia.ac.id	<1	Publication
15	ieeexplore.ieee.org	<1	Publication
16	eprints.rclis.org	<1	Publication
17	A Screening Test for Disclosed Vulnerabilities in FOSS Components, by Dashevskyi, Stanisl- 2018	<1	Publication
18	realestate.pagespeedprofits.com	<1	Internet Data
19	devpost.com	<1	Internet Data
20	internationalpubls.com	<1	Publication
21	Thesis Submitted to Shodhganga Repository	<1	Publication
22	www.dx.doi.org	<1	Publication