DEVELOPMENT OF AN AI-BASED HYPERTENSION RISK PREDICTION AND ADVISORY SYSTEM USING XGBOOST AND GENERATIVE AI

BY

EDOGBANYA, SHADRACH UGBEDEOJO

GOU/U22/CSC/836

DEPARTMENT OF COMPUTER SCIENCE AND MATHEMATICS, FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY, GODFREY OKOYE UNIVERSITY, ENUGU.

IN PARTIAL FULFILLMENT OF THE AWARD OF BACHELOR OF SCIENCE (B.Sc), DEGREE IN COMPUTER SCIENCE.

SUPERVISOR: DR. U.F. OKEBANAMA

JULY, 2026.

APPROVAL PAGE

This research, titled "DEVELOPMENT OF AN AI-BASED HYPERTENSION RISK PREDICTION AND ADVISORY SYSTEM USING XGBOOST AND GENERATIVE AI" has been assessed and approved by the Department of Computer Science and Mathematics Committees in the Faculty of Computing and Information Technology, Godfrey Okoye University, Enugu, Nigeria.

	....	....
Dr. U.F. Okebanama	Signature	Date
(Project Supervisor)		
External Examiner	....	....
External Examiner	Signature	Date
Dr. U.F. Okebanama	....	....
HOD, Department of Computer Science and Mathematics	Signature	Date
Prof. I.A. Ajah	....	....
Dean, Faculty of Computing And Information Technology.	Signature	Date

CERTIFICATION

I **EDOGBANYA, SHADRACH UGBEDEOJO** certify that I completed the research included therein and that it was primarily based on my observations and investigation. If I am identified as having plagiarized on the assignment, I will completely accept the University's decision.

EDOGBANYA, SHADRACH UGBEDEOJO

GOU/U22/CSC/836

DEDICATION

This study is dedicated, first and foremost, to our creator God who has been my source of wisdom, strength, and grace in every step I took in our academic journey. I also recognize this milestone to my dear parents. To my father, CP Sylvester Adaji Edogbanya, thank you for your constant support, encouragement, and financial backing during my stay in Godfrey Okoye University. You have been my earthly anchor. To the loving memory of my mother, Mrs. Joylyn Unekwu Edogbanya, it was your early encouragement that got me to choose Computer Science for my degree and your unwavering trust in God to guide me that still continues to guide me today. You are no longer here to watch me finish, but I know your prayers carried me across the finish line.

ACKNOWLEDGEMENTS

Firstly, all the glory and all the thanks be to Almighty God for giving me life, wisdom and sustaining grace to see me through a successful completion of this research project and the whole degree program.

I gratefully acknowledge of the Head of Department of Computer Science and Mathematics, Dr. U.F. Okebanama who is also my project supervisor. Thanks to your great guidance, helpful technical tips and endless patience. Your mentorship played a key role in transforming "HyperSense" system from an idea to reality. I also extend my thanks to Prof. Ifeyinwa Angela Ajah, Dean, Faculty of Computing and Information Technology for her whole time academic support and administrative prowess.

I will gladly accept this as an answer to my family's longing for words to adequately express how grateful I am. Special thanks to my father and the ultimate pillar of all my dreams, CP Sylvester Edogbanya for all your never ending help, sacrifice and faith in me. I will forever remember the great influence of my loving mother of blessed memory, late Mrs. Joylyn Edogbanya who was my greatest motivator and it was through your encouragement that I was motivated to pursue this degree in Computer Science, know you are smiling down on this achievement.

I am also extremely grateful to my siblings; my elder brother, Mr. Benedict Edogbanya who never left me alone; my elder sister, Ms. Agatha Edogbanya who never failed to motivate and encourage me since my first year and my little sister Ms. Esther Edogbanya for the constant calls to check on me during long nights of coding and research.

Last but not least, to my friends, course mates and all those who helped in any form or shape to make this work, I say thank you. May God abundantly bless you all.

ABSTRACT

Hypertension, or high blood pressure, is a very common health condition in the world. One problem with hypertension is that many people don't know they have it as it can develop without any symptoms. People can therefore begin to function normally without knowing that their health is in danger until they develop complications like stroke, heart disease, or kidney trouble. In recent years a number of mobile and web-based health apps have been created that help people to manage their health records. These can be helpful in keeping track of the data, but many of them only collect data and do not communicate to users their risk of developing high blood pressure in the future. They also offer some assistance with regard to preventive health decisions. This project was specifically concerned with the development of HyperSense, a web-based decision support system to predict the risk of hypertension for the user and offer personalized health advice based on the user's result. This process analyzes various factors associated with hypertension and is done using the XGBoost machine learning algorithm, which is widely used for analyzing the system. These factors are analyzed to calculate a person's risk. Google Gemini was integrated into the platform to make it more functional and interesting. This enables the user to get recommendations based on their own risk profile, rather than only getting general health advice. The application was created to be responsive, allowing it to be easily accessed on a variety of devices. Once the model was developed and it was tested using the validation data. The results showed that the model was able to make predictions with a high level of accuracy. It has an accuracy of 98.24% and sensitivity of 98.35%, demonstrating its ability to accurately detect people who might be at risk for hypertension. This research shows that machine learning and generative AI can be leveraged to develop meaningful healthcare solutions. HyperSense can help guide early intervention and make more positive lifestyle choices, with both risk prediction and personalized recommendations. These systems can potentially help to promote health literacy and better control of hypertension in the general population.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Hypertension is commonly known as high blood pressure and it is one of the most critical public health challenges in modern day. It is often referred to as a silent killer since it can often grow unspotted, as it silently rupts and impairs the blood vessels, kidneys and the heart and a patient may not even know anything is amiss before it becomes too late. The World Health Organisation (W.H.O) estimates that more than 1.13 billion individuals across the world are affected with this disease and by 2030, the number is expected to soar high. The threat of hypertension is not only in the disease itself, but in the fact that the disease is a primary cause of serious cardiovascular diseases, such as stroke and heart failure, the number one causes of mortality across the world (Visco et al., 2023).

The situation is even more menacing in the case of Nigeria, and it should be addressed immediately. The recent research reports a rapid increase in the prevalence of hypertension, shifting about 8.6 percent in the middle of 1990s to more than 32.5 percent in 2020 (Adeloye et al., 2021). This drastic growth has been greatly associated with the rate of urbanization, changing life styles and an increase in the level of stress. Worryingly, the level of awareness is still critical; it is estimated that among hypertensive Nigerians, almost 30 percent are not aware of their condition, and a very small proportion receive proper treatment (Adeloye et al., 2021). This awareness gap can result in late diagnoses whereby the patient does not get medical attention until something disastrous such as a stroke occurs.

Traditional way of approaching this crisis is much dependent on the intermittent clinical visits- the patients are required to visit a hospital and have their blood pressure measured. This is a reactionary and not a proactive way. It does not reflect the changes in the health of the individual on the daily basis and it is frequently a strain to an already overstretched healthcare system. In most developing states, there are not enough medical professionals, and therefore preventative screening is not an issue of concern to an ordinary citizen until it is too late. It is thus urgent to have a system that will enable people to keep track of their own risk levels at the comfort of their homes (Nwevo et al., 2024).

Fortunately, the swift development of mobile technology and Artificial Intelligence (AI) can provide the possible solution that will help overcome this gap. It is said that modern smartphones are everywhere and may be a potent health monitoring instrument. With the help of the combination of Machine Learning (ML) and mobile apps, one can now calculate the increased risk of developing hypertension in a person based on analyzing various health characteristics (age, weight, and lifestyle) with high precision (Islam et al., 2022). These intelligent systems, in contrast to conventional, more rigid statistical techniques, have the capability to become informed by data, providing individual risk risk assessments, which persuades users to act early in the disease process before the situation escalates.

It has been found that Machine Learning models, including the Random Forest and Logistic Regression, may be used to successfully classify the risk of hypertension with the help of detecting the hidden patterns in patient data that may otherwise go unnoticed during a regular check-up (Eshiet et al., 2024). The given project is aimed at taking advantage of these technologies and creating an AI-based Hypertension Risk Prediction and Monitoring System. Through a user friendly mobile interface in conjunction with a powerful smart back end, this system will help to demystify health data to an average user, and translate the silent risks into actionable information that can be used to save lives.

1.2 Statement of the Problem

The absence of continuous and preventive monitoring system is still a serious health problem because hypertension is not monitored continually and low efforts to prevent its occurrence, particularly in developing countries like Nigeria. Most healthcare practice is still reactive, and reliant on visiting the hospital infrequently, which isn't enough to detect changes in a patient's health condition on a day-to-day basis.

The following are the main issues that were found:

1. Many people don't know if they have hypertension or not until they have serious problems.

2. There is no continuous real-time health monitoring with traditional hospital monitoring. Diagnosis is often delayed and severe health problems like cardiac arrest and stroke can result.
3. Current systems are not suitable for the analysis of past health information and early detection of risk. Manual monitoring techniques are more expensive and add to preventable deaths.
4. There are no intelligent and automated systems that can enable every day preventive healthcare.

Hence, an automated and intelligent system for hypertension monitoring is needed which can offer real time risk assessment, and early detection and prevention of severe health complications.

1.3 Aim and Objectives of the Study

The aim of this study is to Develop an AI-based Hypertension Risk Prediction and Advisory System using XGBoost and Generative AI.

The objectives of the study is to:

1. Establish a system for collecting physiological and lifestyle data on hypertension risk factors.

2. Train an XGBoost machine learning model to predict and classify hypertension.

3. Develop a web-based responsive interface to interact with the trained and integrate the prediction model model to monitor their health in real-time.

4. Evaluate the developed system using standard metrics to offer preventive health care.

1.4 Significance of the Study

This study will involve the design, deployment, and testing of a predictive health informatics tool specifically based on Essential Hypertension.

1. Study is an intelligent solution for early prediction and monitoring of hypertension risks before severe health complications, and thus helps in preventive healthcare.

2. The study showcases the application of AI and machine learning tools, such as XGBoost, in the field of healthcare decision support systems.

3. The functions can be accessed via a responsive web-based design that enables users to use the system on various different devices using common web browsers instead of a mobile application.

4. The system is customized to the conditions of environmental, dietary and socio-economic of Nigerians, and its predictions and health recommendations are more relevant and practical.

5. The study will be a useful academic resource for students, researchers and developers interested in artificial intelligence, machine learning, health informatics and web based healthcare systems.

1.5 Scope of the Study

The scope of this study primarily focuses on a hybrid machine learning approach based on Artificial Intelligence (AI) for early prediction of hypertension risk with focus on improving the accuracy of the prediction, using the system to assist preventive health care and making the system applicable in Nigeria.

The following areas are of importance:

1. **Data collection:** The study will involve structured health related data which include physiological data (e.g. blood pressure reading) and lifestyle risk factors associated with hypertension in Nigeria.
2. **Optimization of model performance:** The resulting model will be fitted to labeled data to identify patterns between input features and the outcomes of the hypertension risk and tested using performance criteria like the accuracy and the predictive power of the model.

3. **System Implementation:** The trained model will be incorporated into a responsive web-based decision support system built with HTML, CSS, JavaScript and Python, allowing for the prediction of risks in real-time and interaction with the user via secure API communication.

1.6 Limitations of the Study

Limitations to study include:

1. **Data Limitation:** The machine learning model was trained on publicly accessible medical data sets, meaning there is a possibility that it may not be fully representative of the local population data in Nigeria.
2. **Functional Limitation:** Not a clinical diagnostic system; rather it is a predictive and advisory system. The user must enter the blood pressure measurement value from the medical devices he or she used.
3. **Technology & Connectivity Limitation:** The system is a web-based application and requires modern web browsers and a stable internet connection to get real-time predictions and access to the cloud-based back-end services.

1.7 Operational Definitions of Terms

Within the scope of this study, the following terms are operationally defined to guarantee clarity and understanding of this study:

1. **Hypertension:** A condition of the blood pressure in which the Systolic Blood Pressure is 140 mmHg or higher and/or the Diastolic Blood Pressure is 90 mmHg or higher.

2. **Systolic Blood Pressure (SBP):** This is the pressure against artery walls during heart contraction and blood being pumped.

3. **Diastolic Blood Pressure (DBP):** Blood pressure in the arteries between beats.

4. **Artificial Intelligence (AI):** Computer systems that are able to carry out intelligent human functions like prediction, pattern recognition, and decision making.

5. **Machine Learning (ML):** One of the subfields of AI that allows systems to learn from past data and make predictions without being explicitly programmed.

6. **Supervised Learning:** Machine learning technique in which algorithms are trained with labeled examples consisting of input and output.

7. **XGBoost (Extreme Gradient Boosting):** A sophisticated machine learning model that is known to perform well on structured medical data, which is why it was selected for this study for predicting the risk of hypertension.

8. **Risk prediction:** Analyzing users' health information to determine the level of risk (Low Risk, Moderate Risk and High Risk) of hypertension.

9. **BMI (Body Mass Index):** A health indicator based on weight vs height to determine body fat & health risk.

10. **Large Language Model (LLM):** AI-driven language system to produce tailored health guidance and suggestions.

11. **Application Programming Interface (API):** Communication interface between frontend and backend of system for data exchange.

12. **Flask:** A light-weight Python web framework for building back-end and API services for the system.

13. **Dataset:** A specific set of medical and lifestyle information that can be used to train and test the machine learning model.

14. **Feature Selection:** The process of selecting most relevant features that will make the prediction model more accurate.

15. **Frontend/User interface:** The user interface of the web application.

16. **PostgreSQL:** Open source relational database management system for storing user and prediction data.

17. **RESTful API:** Web communication architecture that allows the secure transfer of data between the front and back end via web requests.

18. **mHealth (Mobile Health):** Use of digital and mobile technologies for provision of healthcare services and health monitoring.

19. **SHAP (SHapley Additive Explanations):** A model interpretation technique for elucidating the contribution of each feature to the prediction results.

20. **Silent Killer:** Hypertension is commonly referred to as the Silent Killer because symptoms are usually not present until serious complications.

21. **HyperSense:** AI-Based Hypertension Risk Prediction and Advisory System: A system designed to predict and advise the risk of hypertension using the HyperSense AI methodology.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The chapter discusses the extensive literature review of AI based Hypertension Risk Prediction and Monitoring System. It covers concepts like pathophysiology of hypertension, machine learning methods for disease prediction and mobile health (mHealth) use. The review gives the theoretical, technological background that is needed to understand the significance and complexity of the proposed system.

Moreover, the chapter reviews critically the existing research work, academic papers, and projects related to hypertension management systems in order to capture the successes and shortcomings of the current research. The review identifies problems including the lack of real-time feedback and poor user interface within the current solutions. All these are some of the key gaps that necessitate the need for the current study and make it more effective, accessible and intelligent to solve the issue of hypertension risk prediction and management.

2.2 Theoretical Background

In this section, the discussion of the core theories, clinical principles, and technical concepts that form the basis of the development of the AI-based Hypertension Risk Prediction and Monitoring System will be discussed in detail. It provides the required background of the study of the intersection of cardiovascular epidemiology and computational intelligence. This section supports the choice of a particular method of the medical domain knowledge combined with the use of the principles of advanced software engineering, proving their appropriateness in solving the problem of silent hypertension.

2.2.1 Summary of Hypertension: The Clinical Setting

The Pathology Hypertension is a complicated physiological disease, which is clinically defined as the persistent increase in the blood pressure (the circle of blood pressure) in the systemic tissue. Blood pressure is measured in two different measurements:

a. Systolic Blood Pressure (SBP): The level of pressure in the arteries during the contraction of the heart.
b. Diastolic Pressure of the Blood (DBP): This is the pressure in the arteries when the heart rests and is in the rest period between the heart beats.

Consequently, the normal reading as per the global medical standards is usually about 120/80 mmHg. The general diagnosis of hypertension is the consistent high readings of over 140 /90 mmHg but in recent years as American College of Cardiology and the International Society of Hypertension traditionally reduced the stage one diagnosis of Hypertension to 130/80mmHg to encourage intervention at a younger age.

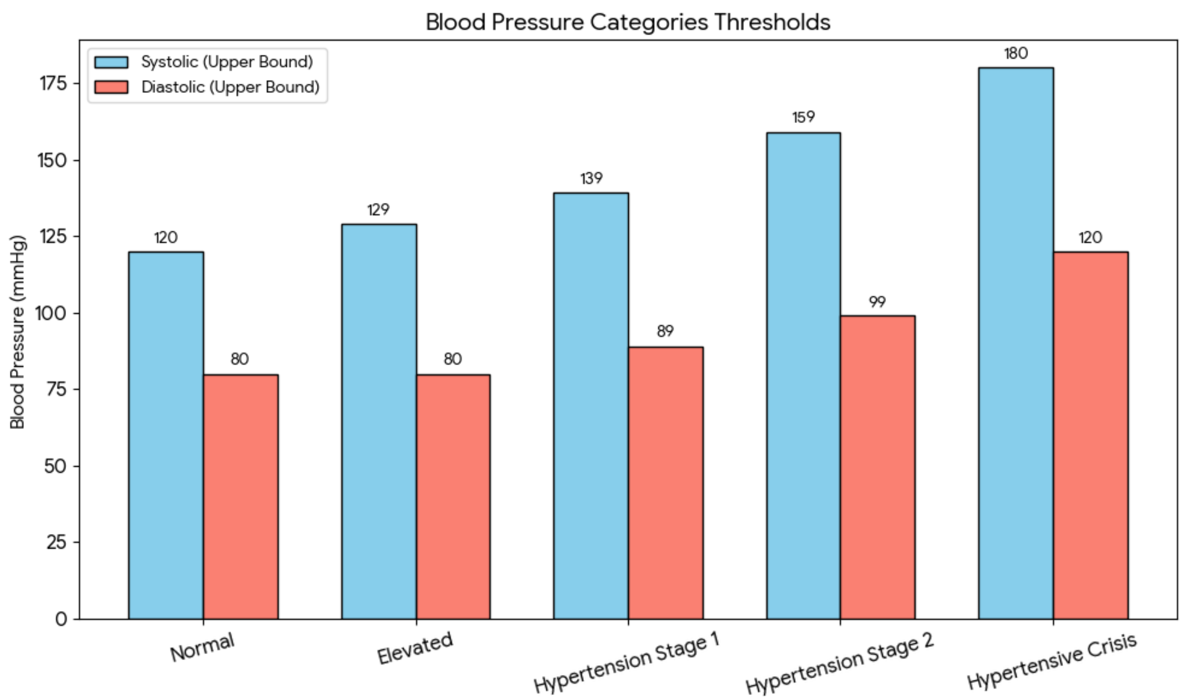

Figure 2.1 Chart Explaining Different Blood Pressure Categories Thresholds

The Phenomenon of the Silent Killer: The name is not only a colloquialism but also a clinical term to describe the fact that the disease is asymptomatic. There is a large segment of the population in a grey zone, as it is observed in the study of the prevalence of prehypertension in Nigeria by Bashir et al. (2021) where the vascular has been damaged, but has already shown no signs of damage in terms of headaches or dizziness. This latency is harmful in that untreated high blood pressure causes sustained pressure to be placed on the cardiovascular system which results in hypertrophy (thickening) of the heart muscle as well as damage to the delicate lining of the arteries (endothelium).

Risk Factors and Classification: In order to develop a prediction system, a person has to know the variables that are causing the disease. There are two major categories of hypertension:

a. **Primary (Essential) Hypertension:** This represents 90-95% of the causes and it does not have a single identifiable cause. Rather it is an outcome of a complicated combination of hereditary and environmental risk factors.
b. **Secondary Hypertension:** This is a condition that occurs because of another underlying problem e.g. kidney disease or thyroid problems.

To use the essential hypertension, in the case of an AI prediction model, we are mainly interested in the case due to its modifiable and non-modifiable characteristics that can be inputted into a Machine Learning model. Some of the major characteristics that are found in the literature are:

a. **Demographics:** Age (arterial stiffening due to old age) and Sex.
b. **Anthropometric Factors:** Body Mass Index (BMI) is an increase in excess weight which raises the amount of blood needed to transport oxygen to the tissues.
c. **Lifestyle Factors:** A high level of sodium (typical of the Nigerian diet), alcohol, and lack of exercise.
d. **History of the family:** An important genetic component.

Nutritional Epidemiology: Nutritional Context of Nigeria and the Risk of Hypertension

Although age and genetics have great impacts in the pathogenesis of high blood pressure, it is stated in recent literature that it is the environmental factors, in this case diet, that have caused the increasing prevalence of cardiovascular disease in West Africa. In Nigeria, the combination of the cultural dependence on particular high-risk food groups and the sudden shift in the nutrition pattern towards processed foods is a perfect storm of hypertensive epidemic. These are the key dietary subtleties that the Hypertension risk prediction system AI-based should understand, as those are the detailed lifestyle characteristics that the machine learning algorithm needs to factor in to estimate the risk correctly.

The Sodium Scourge: Bouillon Cubes and Hidden Salts

In Nigeria, hypertension is engendered by excess sodium intake, due to the use of seasoning agents in cooking which contain high levels of sodium, like bouillon cubes (Maggi and Knorr). Nigerian diets also tend to provide excessive sodium from local cooking ingredients whereas in the West, processed foods are a primary source of sodium. Research has demonstrated that there is a strong link between sodium intake and blood pressure (Batubo et al.)

Excessive sodium intake stressor the arterial walls as it is water retention that leads to increased blood volume. In many Nigerian dishes the amount of salt, crayfish and seasoning cubes used is far in excess of the recommended amount of sodium per day by the WHO. While consumers might not notice these foods as being "too salty" because of habituation, over time, too much sodium causes the walls of the arteries to stiffen, the endothelial cells to become more susceptible to injury, and raises the likelihood of hypertension.

The Role of Oxidized Fats

Palm Oil and Deep-Frying Another important ingredient of the Nigerian diet is the excessive intake of lipids, especially palm oil. Although fresh palm oil contains high levels of antioxidants (carotenoids and Vitamin E), the cooking tradition in Nigeria is to heat the oil until it turns white (bleaching) to make soups and stews. This oxidization process converts the chemical structure of the fatty acids to trans-fats and free radicals which are very atherogenic.

Moreover, the intake of deep-fried street foods (i.e. Akara, Puff-puff) are also considered risk factors. Batubo et al. emphasized that the high-fat diets with fried food, and red meat have a strong relationship with the risk of hypertension among the people of West Africa. These saturated and trans fats also help to form atherosclerotic plaques, which are fatty deposits that reduce the size of the arteries. A direct consequence of a narrowing of the arteries (stenosis) is that, to pump blood through the smaller aperture, the heart has to work harder and this directly leads to an increase in the Systolic Blood Pressure.

Carbohydrate Density and Metabolic Syndrome

Dense carbohydrates being consumed in the Nigerian diet is also known as swallow (e.g., Eba, Fufu, Pounded Yam, Amala). These are high-glycemic foodstuffs. Although it is not directly hypertensive like salt, the overconsumption of these staples leads to obesity and insulin resistance due to the excess consumption of calories. The belly fat (visceral obesity) has been known to cause hypertension since adipose tissue produces inflammatory cytokines that cause blood vessels to contract. Thus, in the suggested AI prediction model, the follow-up of the Body Mass Index (BMI) and the frequency of consuming carbohydrates of a user are more integrated risk factors, as compared to monitoring blood pressure.

A Comparative Analysis of the DASH Diet vs. The Nigerian Reality

In order to put into perspective how dangerous the Nigerian diet is, one needs to compare the current Nigerian diets with the standard gold standard of hypertension management worldwide: the DASH (Dietary Approaches to Stop Hypertension) diet.

The DASH diet is a medically tested nutrition plan that puts significant focus on the consumption of fruits and vegetables, whole grains and low-fat dairy, and puts a severe limit on sodium, red meat, and sweets. DASH works by the abundance of the electrolytes of potassium, magnesium, and calcium, which naturally counteract the effects of sodium and relax blood vessel walls (vasodilation).

The gap in my life that I have identified is the problem of low intake of fresh fruits and vegetables, as indicated by the article by Uyigue and Mohammadnezhad (2024) about adolescents in Nigeria. Economic factors and cultural avenues tend to reduce vegetables to a secondary garnish (e.g. a handful of leaves of ugwu in a soup) instead of a significant part of the diet. This leaves a deficiency in micronutrients. The hypertensive effect of the Nigerian diet is increased in the absence of adequate potassium intake in diseases in vegetables to compensate the high sodium content in the soups.

Consequences to the AI System

The knowledge of this contrast is critical to the Advisory aspect of the proposed system. A blank slap-on-the-forehead generic eat healthy suggestion will not do. The system has to offer culturally particular recommendations e.g. telling to use locust bean (Iru) as a substitute to Maggi to add flavor or to Steam your Plantain rather than frying it. Tuned to the desire to identify these particular local dietary habits, the AI model can be said to have a greater degree of relevance and accuracy to the Nigerian user base.

2.2.2 Conventional Methods of Hypertension Diagnosis and Management

The Manual Clinical Workflow Hypertension diagnosis is a reactive system that is usually performed manually. It is also dependent on the Sphygmomanometry, in which a medical practitioner applies an inflatable cuff and either a stethoscope (auscultatory method) or an automated digital machine (oscillometric method) in order to measure pressure. To clinically diagnose a patient, he or she generally needs several higher readings on different occasions.

Shortcomings of the Traditional Model: As precise when done properly, the traditional clinical model has certain inherent weaknesses, which require the creation of an automated monitoring system:

a. **White Coat Hypertension:** This occurs when the level of blood pressure of a patient soars only when in a hospital setting owing to the fear of visiting a hospital. The result of this is false positive and unneeded medication.
b. **Masked Hypertension:** This is the reverse situation where a patient records normal pressure in the clinic but high pressure at home because of the daily stressors. This is very fatal because it tends to remain unnoticed.
c. **Absence of Ongoing Data:** A doctor visit gives one a snapshot of the health of a patient. It does not reflect the dynamic changes of the blood pressure during the day.
d. **Resource Limitation in Developing Countries:** As it is noted in the International Journal of Hypertension (Ajayi et al., 2021), the number of specialists in Nigeria compared to the patient is disastrous. Relying on visiting only will lead to bottlenecks that will not allow millions of citizens to be detected early.

These constraints highlight the conceptual imperative of the use of Ambulatory Blood Pressure Monitoring (ABPM) and, more future-oriented, AI-assisted predictive monitoring, which allows the patient to evaluate the risk without necessarily having the clinical supervision.

2.2.3 Machine Learning and Uses in Healthcare

Definition and Core Principles: Machine Learning (ML) is a branch of Artificial Intelligence (AI) that deals with creating algorithms that enable computers to learn patterns based on the given data without being programmed to comply with the particular rule. Within the healthcare context, ML is shifting towards a paradigm shift where a rule-based system (e.g., if BP > 140 then High Risk) is replaced with a more data-driven system (e.g., the model learns that a combination of age 45, BMI 30 and slight smoking history predicts High Risk).

Types of Machine Learning: In order to make sense of how the proposed system would operate, one should identify three major pillars of machine learning:

a. **Supervised Learning:** The algorithm is trained using a labeled dataset (Input X and Output Y). This approach was selected in this project. The model is trained on historical patient records in which the status of hypertension is known.
b. **Unsupervised Learning:** It is applied to cluster unlabeled data. This can be applied to cluster related patients but it cannot be used to predict risks directly.
c. **Reinforcement Learning:** An agent learns through its interaction with an environment. This is not as widespread in rudimentary diagnostic applications.

Machine Learning Paradigm: Supervised Learning

Machine Learning (ML) is deeply different to conventional software engineering. In traditional programming, a human programmer has to write out the rules (e.g. should systolic exceed 140 then status = high). Machine Learning on the other hand is the process of building algorithms which enable computers to figure out these rules based on data. Within the definitions of the review by Jayatilake (2021) on healthcare decision-making, the machine learning process entails the maximization of a performance criterion with the help of example data or prior experience.

To comprehend the work of the proposed Hypertension Risk Prediction System, it is essential to describe the two main approaches to machine learning, Supervised Learning and Unsupervised Learning. These types characterize the interaction between the algorithm and the data and the character of feedback that it obtains in the course of training.

Supervised Learning

The Predictive Approach: The most common paradigm of medical diagnostics is Supervised Learning that is the methodology used in the present project. Here, a labeled dataset is given to the algorithm. That is to say, given an input vector, X (e.g., patient age, weight, blood pressure), the value of the output, Y (e.g., Hypertension Status: Yes/No) is already known and given to the system.

The Learning Process: The supervised learning aims to approximate a mapping function, f, such that, Y = f (X). The supervision is brought about by the fact that the answer is known right when training is going on. In hypertension studies, supervised learning can be categorized into two types:

a. **Classification:** This is where the objective variable is a category or a class.
b. **Application:** Predicting whether a patient is High Risk or Low Risk.

The main strength of supervised learning lies in the fact that it can be used to deliver its quantifiable metrics of success. Since the correct answers are known, there are rigorous tests that can be used to test the model in terms of measures such as Sensitivity (the ability to correctly respond to sick patients) and Specificity (the ability to correctly respond to healthy patients). This renders supervised learning, the industry standard of diagnostic support systems with which accuracy is the key factor.

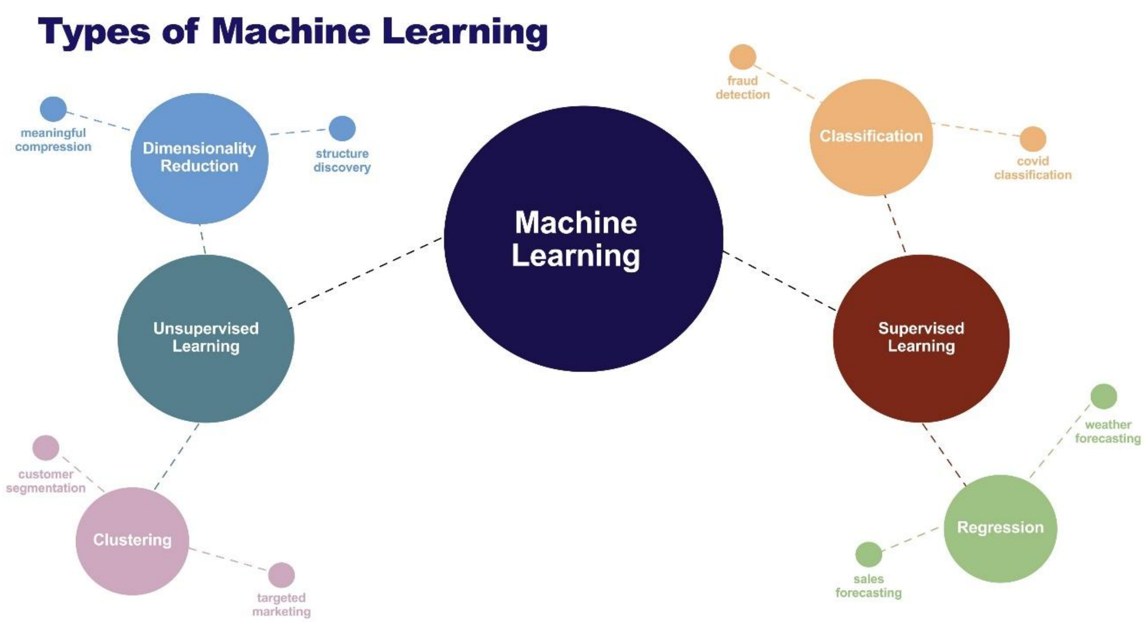

Figure 2.2: Types of Machine Learning

Limitations to Risk Prediction

Unsupervised learning is not ideally suited to the specific task of this project (Risk Prediction). According to Jayatilake (2021), without a labeling of historical data, an unsupervised model cannot be certain that it says This patient is at high risk. It can simply identify as a group of patients that this patient is comparable to. Thus, in case of a diagnostic advisory system, it plays an incidental role as opposed to supervised learning.

Relevance to Healthcare

According to the analysis made by Jayatilake (2021) in his review on ML in healthcare decision-making, the advantage of the method is that it can manage non-linear relations. This is not necessarily the case in biological systems where 1 +1 =2. As an illustration, a small amount of weight gain may be not dangerous to a young individual, but may be life-threatening to an old adult who has a family history of stroke. These are subtle, multi-dimensional interactions that can be missed by human doctors when one comes into a check-up room and thus are detected by ML algorithms. It can be used in medical imaging (finding tumors on X-rays), to predictive analytics (predicting disease outbreaks or personal risk profile), the field of this study.

2.2.4 Machine Learning Techniques for Identifying Hypertension

In this subsection, the particular computational methods applicable in the prediction of hypertension risk are reviewed, in order to justify the choice of the algorithms to be used on the given project.

Preprocessing and Selecting features: Prior to any algorithm being run, data has to be processed. Medical data is generally noisy or incomplete as highlighted by Pudjihartono et al. (2022) in the article titled Frontiers in Bioinformatics.

a. **Feature Selection:** Only not every point of data is useful. This is done by picking out the most appropriate variables (e.g. Systolic BP, Age, BMI) that lead to the greatest contribution to the predicted variable. This lowers the cost of computation and also eliminates overfitting (when the model recalls the training data but not on real users).
b. **Normalization:** This is done by scaling of the data (e.g., age 0-100 and weight 0-200) with another feature such that no feature will dominate the other.

The value domain of the imperative of data security in health care

Health information is considered Sensitive Personal Data in both international and local laws (HIPAA in the USA and NDPR in Nigeria). Contrary to the financial data which can be re-initiating (e.g. cancelling a credit card), the medical data is fixed and therefore, once it is leaked, a patient will not be able to alter his or her medical history or genetic predisposition.

Within the framework of this project, the system receives:

a. **Physiological Metrics:** Systolic/Diastolic Blood Pressure, Heart rate.
b. **Anthropometric Data:** Age, Weight, Height.
c. **Lifestyle Indicators:** Alcohol consumption, Smoking status.

This information will be used to unjustly discriminate people (e.g., health insurance or even work opportunities due to the possibility of developing hypertension). Thus, the system architecture should comply with CIA Triad of Information Security:

a. **Access Control:** Only the authorized users will have access to the data.
b. **Integrity:** The information is not distorted or manipulated in the process of transmission.
c. **Availability:** The data is available when it is necessary to the user.

2.3 Review of Related Works

This section will give the critical review of the past research works, scholarly articles and technical reports concerning the development of hypertension monitoring systems. It emphasizes the techniques used by the other researchers, the findings they have achieved, and the limitations they faced specifically and that this present study is meant to overcome them.

2.3.1 Prevalence, Risk Factors, and the Nigerian context

In order to realize the importance of an automated risk monitoring system, it is important to recognize the enormity of the issue in the local environment. Bashir et al. (2021) carried out a systematic review and meta-analysis in a pioneer study to establish the prevalence of prehypertension in Nigeria. The researchers were able to use a random effects model to estimate that 34.6% of Nigerian population is in prehypertensive stage- a warning zone where chronic disease can be prevented with early intervention by combining data of 21 separate studies with more than 25,000 participants. The paper presented that the North-West region was the worst hit and that men were more susceptible statistically compared to women. Nevertheless, a significant weakness that they found in their work was that they did not have longitudinal data; most of the studies were cross-sectional snapshots, i.e. there was no mechanism established to determine the rates at which these prehypertensive individuals moved to full hypertension, which would be addressed by a continuous monitoring application.

On a broader level, Olowoyo et al. (2025) examined the trends of hypertension in Africa within the past 20 years (2000-2023). Their results were disheartening, and they showed that even though there are many programs aimed at promoting the importance of health through the various avenues of public health, hypertension control rates are pathetically low. They claimed that fast urbanization is a more important cause of the disease in Africa than genes, because the rural populations relocate to the urban areas, and lead sedentary lives. The researchers observed that the existing control measures fail due to the fact that they are based on the hospital management, which cannot be accessible to an average citizen. This brings about a good reason why decentralized and mobile based solutions are needed. This disparity in the world was also highlighted by Schutte et al. (2023) in an article published by the International Society of Hypertension. They contrasted the results of blood pressure control in high-income and low-income countries and found that, in the West, the use of digital health tools is considered standard, whereas, in Sub-Saharan Africa, it is grossly underused because of the infrastructural neglect. According to their work, any intervention that is to be successful in this region should be low in price and be very accessible through smartphones.

Batubo et al. (2024) examined the particular biological causes of this epidemic carefully and paid attention to the diet peculiar to West Africa. After conducting a systematic review of observational studies, they found that the intake of sodium-rich bouillon cubes (as a part of the Nigerian soups) is directly and non-linearly correlated with high systolic blood pressure. They also singled out palm oil and red meat as major contributors of cardiovascular risk. This study is especially crucial to the "Advisory" aspect of the proposed system as this study shows that the generic Western dietary guidelines cannot be used to conduct effective preventive actions; to be effective, the prevention must involve culturally specific guidelines. Equally, Malik et al. (2022) examined the risk factors on the African continent more broadly, with alcohol consumption and high Body Mass Index (BMI) being the most relevant risks that are modifiable. Nonetheless, they disapproved of current interventions as being one-size-fits-all and not taking into consideration individual differences in risk tolerance, which an AI-based system can fix with a one-on-one modeling approach.

Breaching the stereotype about hypertension as the illness of the elderly, Uyigue and Mohammadnezhad (2024) concentrated on adolescents in Nigeria. The systematic review of school-based health screening showed that there is an alarming increase in high blood pressure among teenagers (ages 10-19), which is mostly due to the intake of fast food and physical inactivity. One of the weaknesses they found was that current screening instruments are not set to younger populations and tend to falsely categorize danger. This shows that a machine learning model is required to dynamically regulate risk baselines depending on the age of the user. Ajayi et al. (2021) unveiled the repercussions of not diagnosing this condition at an early stage because hypertension was identified as the main etiological agent of Chronic Kidney Disease (CKD) in Nigeria through a meta-analytic study. They discovered that the disease is silent hence very often it is diagnosed when target organ damage has already taken hold. They suggested a transition to an aggressive early detection, and it perfectly conforms to the goals of this project.

2.3.2 Healthcare machine learning algorithms

When passing on to the technological solution, Engda, Salau, and Ajala (2025) gave a critical analysis of the classical approaches of the machine learning in predicting hypertension. They comparatively evaluated older algorithms with newer Deep Learning models in their systematic review. Their findings indicated that in the case of tabular health data (structured data such as age, weight, and BP readings), ensemble algorithms such as Random Forest and XGBoost are seen to perform better than a more complex Neural Network in terms of accuracy and interpretability. They claimed that Deep Learning is an excessive step to take to solve this particular issue and it is too computationally intensive. Chowdhury et al. (2022) attempted to resolve the debate between AI and traditional statistics. They compared the results of Logistic Regression models to those of the machine learning (ML) models in 20 separate studies and discovered that ML models obtained higher scores on Area Under the Curve (AUC) (usually 0.85-0.90) than the regular regression (0.70-0.75). They explained this by the fact that ML algorithms can identify non-linear interactions among risk factors that standard statistics are unable to identify.

Nwamekwe et al. (2024) were able to prove the viability of the implementation of these advanced algorithms in the Nigerian healthcare infrastructure. Even though their research was on maternal mortality, it is very applicable since it was able to test prediction algorithms with local Nigerian data on hospitals. They obtained a high level of accuracy (more than 85), which shows that local datasets are good enough to be trained to strong models. Nevertheless, they have stated that bad data record-keeping in the hospitals is a significant challenge and patient-facing apps that receive data directly inputted by the patient may be more trusted than the digital hospital document. The review of Shahgholian et al. (2025) was focused on the particular execution of classification algorithms. Their claims were that Support Vector Machines (SVM) had high-dimensional space accuracy, but Decision Trees and Random Forests have better mobile implementation due to being computationally less heavy and consuming less battery power.

Continuing on the use of such tools, Nuryunarsih et al. (2023) tried to anticipate not only the existing risk, but also the progression of a blood pressure with the help of Artificial Neural Networks (ANN). Their model was highly sensitive, but they have pointed out that there was a major limitation namely, the model needed a tremendous amount of historical data per patient to operate, which poses a cold start problem when the model is used by new users who do not have years of medical history. The article by Cai et al. (2021) has reviewed the application of ML in the clinical care pathway of heart failure and hypertension. They discovered that ML is very good at risk stratification (placing patients in the Low, Medium, and High risk category), which allows physicians to prioritize their workload. They however criticized most of the available systems claiming that they are standalone models which are not compatible with real mobiles or doctor dashboards.

Liu et al. (2025) concentrated on the information sources of these models, and did a meta-analysis concerning the use of Electronic Health Records (EHR). They affirmed that incorporation of social determinants of health (including income level and location) in the ML model had a tremendous better fit than when medical data was utilized by itself. This justifies the fact that lifestyle questions should be incorporated in the proposed app. On the technical aspect of data preparation, Pudjihartono et al. (2022) conducted a review of the Feature Selection methods. They proved that feeding an AI model with excessive amount of data (noise) actually reduces the level of accuracy. The methodology of this project is justified by the fact that their work will only pick the most important 8-10 key health indicators (Feature Extraction), and not dozens of irrelevant questions that will overwhelm the user.

2.3.3 Adoption and Implementation of Mobile Health (mHealth)

The adoption of users is the key to the success of a technical solution. Aljohani and Chandran (2021) used the Technology Acceptance Model (TAM) on developing countries patients. A review of 22 papers established that in some areas such as Nigeria, the perceived ease of use is the main factor that leads to the adoption of the app, rather than its advanced features. They determined that those applications that need to be connected to the internet all the time tend to fail in rural environments which indicates that an offline-first architecture is essential. Hengst et al. (2023) examined the obstacles to the mHealth adoption among individuals with low socio-economic status. The key challenges pointed out by them include digital literacy and graphical complexity. Their investigation suggests that health apps in these groups need to be presented in simple forms (such as Red/Green traffic light colors in case of risk) instead of complicated medical charts-an aesthetic rule that is utilized in the given research.

Reviewing the existing market situation, Cucciniello et al. (2021) implemented a systematic investigation of the existing mobile applications in chronic conditions. Their research revealed a worrying discrepancy between their findings: on the Google Play Store, there are thousands of health apps, but not more than 5% of them comply with clinical recommendations or provide evidence-based information. The vast majority of them were discovered to be mere trackers that are recording numbers without providing any kinds of intelligent analysis or prediction. This is an indication of a blank market in the proposed system that intends to offer AI-based intelligence and not merely data storage. The article by Chiarito et al. (2023) provided a required counter-argument in the form of the article titled All That Glitters is Not Gold. They cautioned that AI models tend to be black boxes, which are not trusted by clinicians since they cannot say how they have arrived at a particular decision. According to them, an AI system needs to be ethically sound, which means it needs to be explainable: it needs to tell the user the reasons as to why they are at risk (e.g., Your risk is high intimately because of your BMI and Age), and not merely give a fearful score.

Lastly, Jayatilake and Ganegoda (2021) conducted a review of the purpose of ML in healthcare decision-making in terms of its legal and ethical aspects. They came to the conclusion that the power of algorithms is in the future of healthcare where the AI will propose a diagnosis, but the human will remain in control. They recommended that every predictive application should have easy to understand disclaimers to the effect that they are not a substitute to a doctor, but an advisory only-which is a critical finding in operationalizing this project. The latter was supported by Secinaro et al. (2021) in their systematic literature review, who stated that the most successful AI projects in the health field are based on preventive medicine and diagnostics of patient data, and not trying to substitute interventional surgery or acute care.

2.4 Summary of Related Works

Author(s)/Year	Journal Name	Contributions	Techniques/ Methodologies	Shortfall
Bashir et al. (2021)	Systematic Review & Meta-analysis on Prehypertension in Nigeria	Established that 34.6% of Nigerians are prehypertensive and highlighted regional/gender differences.	Systematic review, meta-analysis, random effects model, 21 studies with 25,000+ participants.	Mostly cross-sectional data; lacked longitudinal monitoring of progression to hypertension.
Olowoyo et al. (2025)	Hypertension Trends in Africa (2000–2023)	Showed that hypertension control rates in Africa remain poor despite public health programs.	Trend analysis and review of hypertension data across Africa.	Existing interventions depend too much on hospital-based care, limiting accessibility.
International Society of Hypertension / Schutte et al. (2023)	International Society of Hypertension Publication	Compared BP control outcomes between high-income and low-income countries; emphasized need for smartphone-based interventions.	Comparative global hypertension analysis.	Digital health tools are underused in Sub-Saharan Africa due to infrastructure limitations.
Batubo et al. (2024)	Systematic Review on West African Diet and Hypertension	Identified sodium-rich bouillon cubes, palm oil, and red meat as major contributors to	Systematic review of observational studies.	Generic Western dietary recommendations are ineffective for African populations.

		cardiovascular risk.		
Malik et al. (2022)	African Hypertension Risk Factors Study	Identified alcohol consumption and high BMI as major modifiable hypertension risks.	Continental risk-factor analysis.	Existing interventions are too generalized and fail to personalize care.
Uyigue & Mohammadnezhad (2024)	Adolescent Hypertension in Nigeria	Revealed rising hypertension rates among Nigerian teenagers due to fast food and inactivity.	Systematic review of school-based health screenings.	Existing screening tools are not tailored to adolescents and may misclassify risk.
Ajayi et al. (2021)	Hypertension and CKD in Nigeria	Identified hypertension as a leading cause of Chronic Kidney Disease in Nigeria.	Meta-analysis study.	Hypertension is often diagnosed too late after organ damage has occurred.
Engda, Salau & Ajala (2025)	Machine Learning for Hypertension Prediction	Demonstrated that Random Forest and XGBoost outperform Deep Learning for tabular health data.	Systematic review comparing ML and Deep Learning models.	Deep Learning models are computationally expensive and unnecessary for structured datasets.
Chowdhury et al. (2022)	ML vs Traditional Statistics in Healthcare	Showed ML models achieve better AUC scores than Logistic Regression.	Comparative review of 20 studies.	Traditional statistical models fail to capture non-linear interactions among risk factors.
Nwamekwe et al. (2024)	AI Prediction Models in Nigerian Healthcare	Proved that Nigerian hospital datasets can train accurate prediction	Predictive modeling using local hospital datasets.	Poor hospital data record-keeping affects reliability.

		models (>85% accuracy).		
Shahgholian et al. (2025)	Classification Algorithms Review	Found SVM highly accurate while Random Forest and Decision Trees are better for mobile deployment.	Review of classification algorithms.	SVM models are computationally heavier for mobile devices.
Nuryunarsih et al. (2023)	ANN-Based Blood Pressure Prediction	Developed ANN models capable of predicting BP progression over time.	Artificial Neural Networks (ANN).	Requires large historical datasets per patient, creating cold-start issues for new users.
Cai et al. (2021)	ML in Heart Failure and Hypertension Care	Demonstrated ML effectiveness in risk stratification for patient prioritization.	Review of ML applications in healthcare pathways.	Most systems are standalone and not integrated with mobile apps or clinician dashboards.
Liu et al. (2025)	EHR and Social Determinants Meta-analysis	Showed that combining social determinants with medical data improves ML prediction accuracy.	Meta-analysis using Electronic Health Records (EHR).	Many ML systems ignore lifestyle and socio-economic factors.
Pudjihartono et al. (2022)	Feature Selection Methods Review	Demonstrated that feature selection improves AI model performance by removing noise.	Review of feature extraction and selection methods.	Excessive input variables reduce model accuracy and increase user burden.
Aljohani & Chandran (2021)	TAM and mHealth Adoption in Developing Countries	Found that ease of use is the strongest factor affecting app adoption.	Technology Acceptance Model (TAM), review of 22 papers.	Apps requiring constant internet access perform poorly in rural areas.

Hengst et al. (2023)	mHealth Adoption Barriers Study	Identified digital literacy and interface complexity as major barriers to adoption.	Socio-economic and usability analysis.	Complex medical charts discourage low-literacy users.
Cucciniello et al. (2021)	Review of Chronic Disease Mobile Apps	Found that fewer than 5% of health apps comply with clinical guidelines.	Systematic review of mobile health applications.	Most apps only track data without intelligent analysis or prediction.
Chiarito et al. (2023)	"All That Glitters is Not Gold"	Emphasized the need for explainable AI in healthcare systems.	Critical review of AI ethics and trustworthiness.	Black-box AI systems reduce clinician trust and transparency.
Jayatilake & Ganegoda (2021)	ML in Healthcare Decision-Making	Discussed ethical and legal considerations of AI-assisted healthcare decisions.	Review of AI decision-support systems.	AI systems must clearly state they are advisory tools, not replacements for doctors.
Secinaro et al. (2021)	AI in Preventive Medicine Review	Found that the most successful healthcare AI projects focus on prevention and diagnostics.	Systematic literature review.	AI is less effective when attempting to replace acute care or surgical intervention.

Table 2.1: Summary of Related Works

2.5 Knowledge Gap

The literature studied shows that there is a huge gap between research and practice in relation to hypertension in Nigeria. Though it has been proven that machine learning models can predict hypertension, existing systems are limited to Western data sets and do not take into account other risk factors in the local area such as excessive intake of bouillon cubes (rich in sodium) and oxidized palm oil. In addition, most existing mHealth solutions are primarily passive data trackers that do not have intelligent and real-time risk predicting capability.

The review also calls for problems of usability and transparency. The constant need for internet connection and advanced digital skills in many health applications restricts access for people in low-resource settings. Moreover, the "black-box" engineering of many AI models decreases trust due to the lack of explanations for the risk scores that are given. To fill these gaps, this study introduces an AI-based system that is context-aware and explains the identified risks to give actionable health advice for hypertension prevention and management, using local risks to solve these problems, this study proposes an AI-based system that is context-aware and able to explain the identified risks so as to give actionable health advice for hypertension prevention and management using local risks.

CHAPTER THREE

SYSTEM ANALYSIS AND DESIGN

3.0 Introduction

This chapter outlines the methodology, system architecture and the analysis used in the design of the HyperSense Hypertension Risk Prediction and Advisory System. It measures the weaknesses of the current, standalone approaches to blood pressure monitoring, namely, the fact, that they do not give real-time, data-based lifestyle and dietary recommendations.

Moreover, the chapter provides the structural design of the proposed web-based decision support system, which includes the description of how a dynamic user interface and a powerful Python backend will be integrated. The most critical aspect of this design is the use of the XGBoost machine learning model that is programmed to handle multifactorial parameters of users, such as systolic and diastolic measurements, body mass index (BMI), stress levels, and local dietary patterns (such as the consumption of sodium) to create precise, automated risk forecasts. This section, ultimately, is a detailed technical plan, where the theoretical goals of the project are converted to a realistic, scalable software engineering form.

3.1 System Analysis

System analysis involves breaking down the proposed system and examining its components and their interaction. This section of the research work addresses the comprehensive system analysis of the proposed system, encompassing an evaluation of the already existing system and identification of the limitations. However, the analysis of the proposed system encompasses the detailed modelling activities using use case diagram, activity diagram, and class diagram.

3.1.1 Overview of the Problem Domain

Hypertension is a long-term health disorder that is typified by increased blood pressure and is sometimes referred to as a silent killer because it does not show symptoms until it progresses to extreme cardiovascular complications. Within the frames of the given research, the area of the problem is the constant level of blood pressure (Systolic and Diastolic rates) and the assessment of the interdependence of lifestyle factors, such as age, body mass index (BMI), stress, and certain diet patterns. One of the most important gaps in present health management practices, especially in developing countries, is the transformation of the data collection into the action-relevant, culturally applicable predictive analysis. People tend to fail to comprehend the correlation of their daily routine and their physiological parameters, which results in poor preventive health care.

3.1.2 Analysis of the Existing System

The current model of blood pressure control is mainly based on the reactive, manual and very fragmented system. Blood pressure is normally measured by patients on standalone digital sphygmomanometers or aneroid sphygmomanometers either at the clinic or at home.

The workflow of the existing system comprises of:

1. **Measurement**: Reading blood pressure at regular intervals.
2. **Manual Logging:** This is a form of recording these values, in hardcopy notebooks or simple computer notes to be reviewed by the physician at a later date.
3. **Generic Consultation:** This involves being given general medical advice by clinical visits that are few.
4. **Generic Digital Tools:** The use of generic mobile health apps which capture numerical data, but provide generalized and usually Western-centric dietary recommendations (e.g. suggesting a Mediterranean diet instead of talking about domestic culinary delights).

3.1.3 Weaknesses of the Exiting System.

The critical review of the current system shows that it has a number of strong weaknesses that do not allow efficient hypertension management:

1. **Lack of Predictive Analytics:** The existing systems are only diagnostic and reactive and only the user is alerted when the blood pressure has already reached dangerous levels, instead of calculating the trends to anticipate the risk in the future.
2. **Isolation of data:** The tools used currently assess the blood pressure in an empty chamber. They do not compare physiological measures with other important data on the environment and lifestyle, including daily stress levels, sleep time, and exercise.

3. **Cultural and Dietary Disconnect:** Standard applications do not have a local context. They fail to consider regional food issues that play a significant role in determining sodium levels e.g. the high frequency of taking high-sodium bouillon seasoning cubes or swallow meals that contain heavy amounts of carbohydrates.
4. **Inadequate Medication Adherence monitoring:** The manual logbooks and simple apps lack the ability to offer a user-friendly and visual tracking system to enable patients to stick to their daily intake of hypertension medications.
5. **Absence of Real-Time Decision Support:** Patients are not provided with an immediate and smart communication technology to make daily health decisions, since they are required to make sense of complex numerical data on their own, between doctor visits.

3.1.4 Analysis of the Proposed System

In order to address the encountered problems, a smart, localized decision support system is urgently required. The suggested HyperSense web application will attempt to substitute the status quo of reactiveness with a proactive and predictive framework. With the help of the XGBoost machine learning solution, the system will handle multi-dimensional and complex inputs (BP metrics, BMI, stress, and particular dietary habits) to produce real-time risk categories. Moreover, the system will fill the cultural divide by converting algorithmic predictions into actionable, localized lifestyle recommendations, and eventually transform the paradigm of hypertension treatment as a clinical check-up periodically to a personal health monitoring with the assistance of AI.

3.2 System Requirements Specification (SRS)

In this section, the specifications of a successful implementation and operation of the HyperSense application are described. These requirements are broken down into functional requirements and non-functional requirements that define the explicit functions the system will execute and the quality attributes, performance objectives, and security standards of the system, respectively.

3.2.1 Functional Requirements

Functional requirements identify the fundamental behaviour, processes and interactions that the decision support system should support the user. In order to fulfill the goals of this project, the system has to be able to do the following significant functions:

1. **User Data Acquisition and Profiling:** The system should include easy-to-use interface (modal forms) to record instantaneous physiological and lifestyle data, (Systolic and Diastolic blood pressure, age, weight/BMI, stress levels and local dietary habits) e.g., sodium/bouillon cube intake.
2. **Machine Learning Risk Prediction:** The system should send the user inputs to the Python (Flask) backend where the built-in XGBoost machine learning model will assess the parameters and automatically determine the level of risk of hypertension (i.e., Low Risk, Elevated Risk, High Risk).
3. **Culturally Relevant AI Advisory Generation:** Depending on the results of the prediction and specific lifestyle data provided by the user, the system is to use the Google Gemini AI API to produce dynamic, Nigerian-contextualized health advice (e.g., by recommending local spices such as Locust beans/Iru instead of high-sodium seasoning cubes).
4. **Multimodal Conversational AI (HyperSense SoberBot Chat):** The system should have its own chat interface where the user can post questions related to health. The system should be able to accept multimodal inputs, where users can post text or images (including pictures of meals or clinic results) so that the AI can analyze them and give them contextual feedback.
5. **Medication and Habit Tracking:** The system should include an interactive medication tracker that enables the user to record the daily medication intake, which will update a weekly calendar. The system should monitor and display lifestyle goals on a daily basis, e.g. the number of liters of water consumed, and stress management with dynamic SVG activity rings.
6. **Data Storage and Visualization:** The system should be able to store the past blood pressure measurements and daily habits safely. This historical data should be translated into visual representations by the system (e.g. sparkline graphs and trend indicators) to enable users monitor their cardiovascular health over time.

3.2.2 Non-Functional Requirements

Non-functional requirements are the standards of the evaluation of the system functioning and guarantee that it is stable, prompt, and safe.

1. **Performance and Efficiency:** Two-second hypertension risk prediction and efficient communication with outside AI systems.
2. **Usability and Accessibility:** User interface should be responsive on mobile, tablet and desktop devices, including on screen color-coded risk indicators to enhance user experience and light/dark mode.
3. **Security and Privacy:** The system should be able to protect user health data, validate all incoming data and securely store sensitive health data configurations like API keys.
4. **Scalability and Architecture:** The front and back end should be designed separately to ensure scalability, maintainability, and efficient processing.
5. **Reliability and Error Handling:** The system should be able to recover from errors and report user-friendly messages in case of invalid data entry or when services are unavailable.

3.3 System Design Methodology

The HyperSense Hypertension Risk Prediction and Advisory System development takes a hybrid form with Agile Software Development Methodology to the project lifecycle, whereas the system modeling takes place with Object-Oriented Analysis and Design (OOAD). Agile is used to define the iterative stages of coding, testing, and implementation, whereas OOAD is used to illustrate the structural paradigm of the application components, behaviours and data flows with the help of Unified Modeling Language (UML).

3.3.1 Justification for Methodology

The combination of Agile and OOAD was predetermined by the need to create a web-based machine learning application that will need both structure and developmental flexibility.

Why I used Agile: This project is being developed using the Agile methodology because of its iterative and incremental approach which is appropriate for complex systems involving the creation of machine learning models and AI APIs. The development process was segmented into various stages: Designing the responsive front-end, building the Python/Flask back-end, and integrating the XGBoost prediction model and the Google Gemini API. This allowed for flexibility, enabling the addition of new features, prediction variables (e.g., BMI, stress score) without impacting the overall development process, and testing and adaptation along the project's course.

Reason for using OOAD: OOAD is adopted due to its modularity, clarity and component reusability. HyperSense is composed of the user interface, Flask backend, XGBoost prediction model and Google Gemini API. With OOAD, these components can be developed as independent objects that communicate with each other via the JSON data exchange and keep the machine learning logic and the web interface separate. This modular design allows for easy testing, maintenance, debugging and scalability.

3.3.2 Method of Data Collection

A mixed-method approach to data collection was applied in order to make the system both technically sound and practically applicable to the target users:

1. **Questionnaires (Primary Data):** Information on users' blood pressure monitoring practices, challenges with taking medications and diet were obtained using questionnaires. The results guided the architecting of system functionality elements, like the medication tracker and the localized health advisories.
2. **Secondary Data Sources:** Secondary data was gathered from public medical data sets with variables like blood pressure, age, BMI, sleep time, and stress. XGBoost machine learning model was trained and tested using these datasets.
3. **Observation (Document Review):** The current state of mobile health applications and the traditional ways of keeping blood pressure records were reviewed to determine the strengths and weaknesses. During the review, it was discovered that there are gaps, including that there is no predictive analytics and localized health information, which is what the HyperSense system aims to fill.

3.4. System Modelling using UML

In this section, the Unified Modeling Language (UML) is used to give a standard, graphical outline of the HyperSense system architecture. These diagrams, based on the Object-Oriented Analysis and Design (OOAD) approach, transform the system requirements into logical, structural and behavioral models to ensure the software implementation phase has a clear roadmap.

3.4.1 Use Case Diagram of the Proposed System

The Use Case Diagram represents the system functionalities in the eyes of the users (actors), and the outside systems. It specifies what the system does but not how it does it.

The main participant of the HyperSense system is the Patient/User that communicates with the system and provides the vital signs, daily habits (medication and water intake), and historical health dashboard. Two secondary system actors are also incorporated into the system, namely: the XGBoost Model that is the predictive engine and the Gemini API that is the conversational and advisory engine. The figure demonstrates the interdependent nature of the responses of these external machine learning models in response to the user requesting a risk assessment.

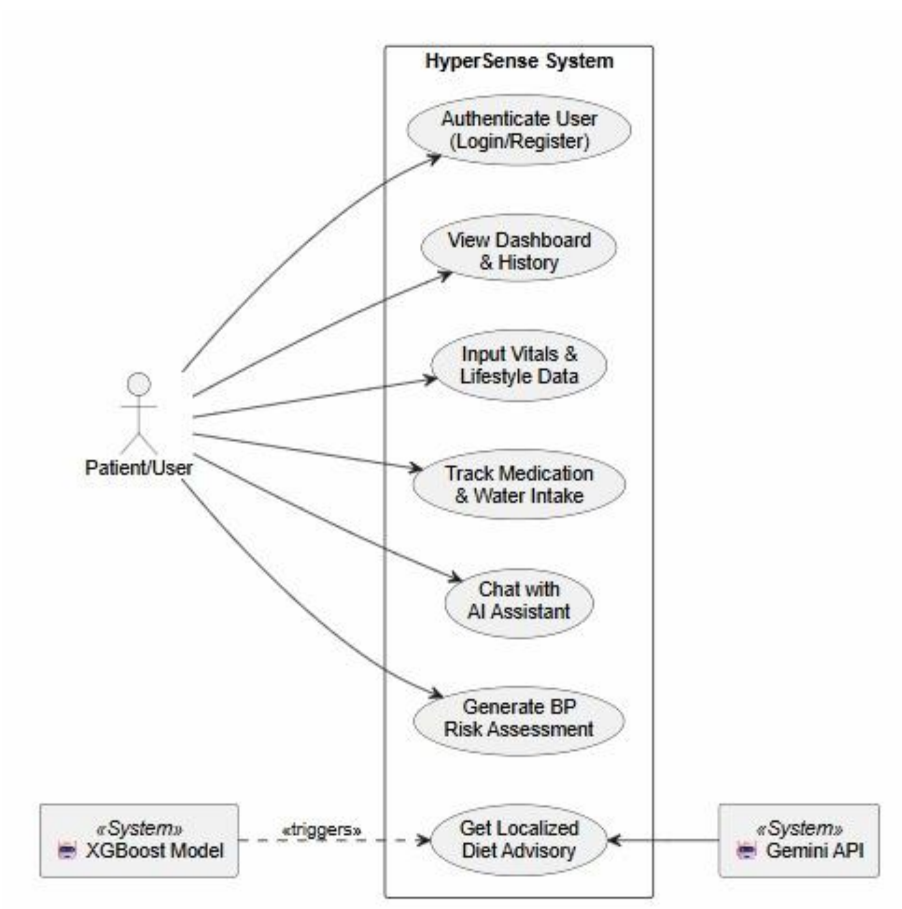

Figure 3.1: Use Case Diagram of HyperSense System

3.4.2 Activity Diagram of the Proposed System

The Activity Diagram is used to show the dynamic behavior of the system by charting the step-by-step process of the main functional process of the system: the Hypertension Risk Analysis.

The workflow starts with the user keying in his/her physiological information (Blood Pressure, BMI, Stress and Dietary inputs) into the frontend user interface. Client-side validation is done to validate data integrity in the system first. When the validation is successful, the workflow is passed to the backend server where the data is being parsed. The XGBoost algorithm is then provided with the control flow in order to calculate the risk classification. This is then sent to the Gemini API to produce a culturally contextualized dietary advisory according to the degree of risk. Lastly, the aggregated payload is sent back and displayed to the dashboard of the user and the activity path is closed.

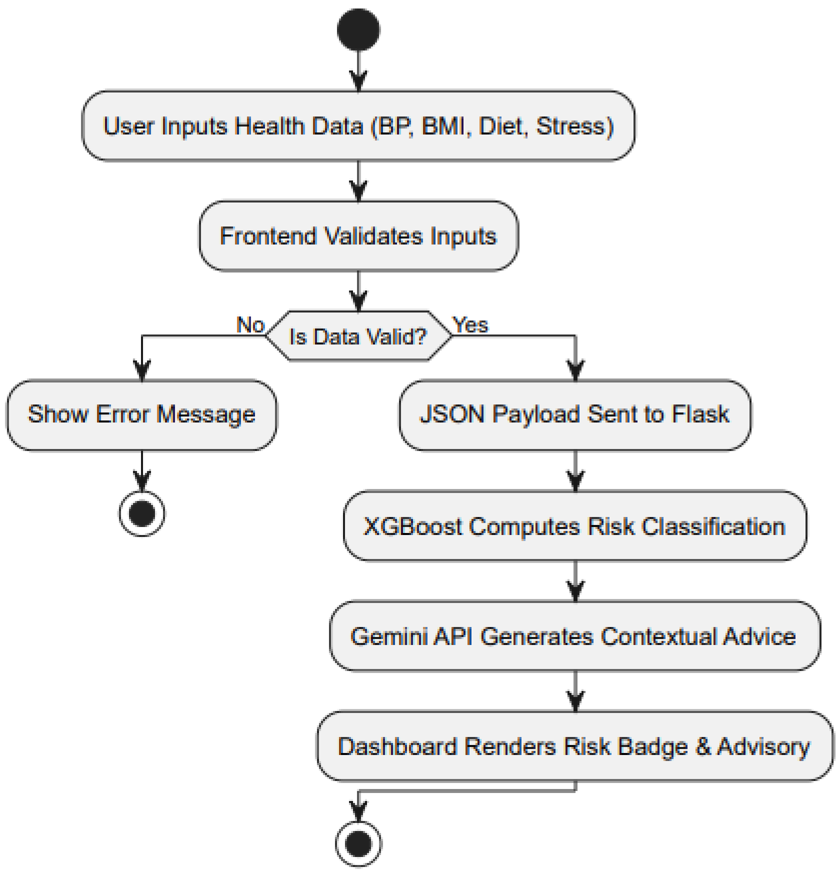

Figure 3.2: Activity Diagram showing the Prediction Flow

3.4.3 Sequence Diagram of the Proposed System

The Sequence Diagram gives a time-related perspective of how the various decoupled modules of the system interact with each other to perform a given task.

This model emphasizes the asynchronous flow of data when making a prediction request. It traces the history of the JSON payload that is being sent between the Frontend UI and the Flask Server. It shows how the server serves as a mediator of the control- take the organized array to the XGBoost Model and then retrieve the classification and then make a secure POST request to the Gemini API. The diagram is a visual demonstration of the reality that the application is decoupled as it demonstrates that the frontend is independent and that the backend is responsible to perform the intensive computational and API orchestration and then returns the end-user to the user interface with the final JSON response.

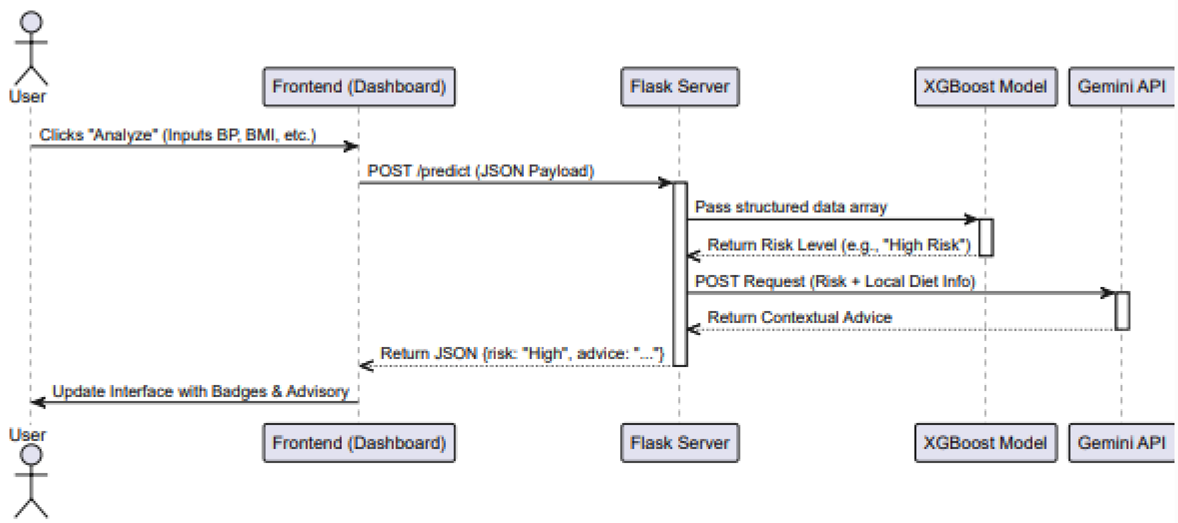

Figure 3.3: Sequence Diagram of the Prediction and Advisory Request

3.4.4 Class Diagram of the Proposed System

The Class Diagram is a static object-oriented architecture of the application. It specifies the essential data blueprints (classes), their inner properties, how they perform and the rigid connections among them.

The model is broken down into logical entities. The User class manages demographic information and BMI computation and is in a one-to-many relationship with the HealthRecord class, which safely stores the daily systolic, diastolic, stress and sodium intake values of a user. The structural association depicts that the cases of the HealthRecord are transmitted to the PredictionEngine class and are evaluated. In addition, the AIAdvisory class is also a separate integration object, which has the methods needed to process the XGBoost output into natural language prompts to the conversational interface.

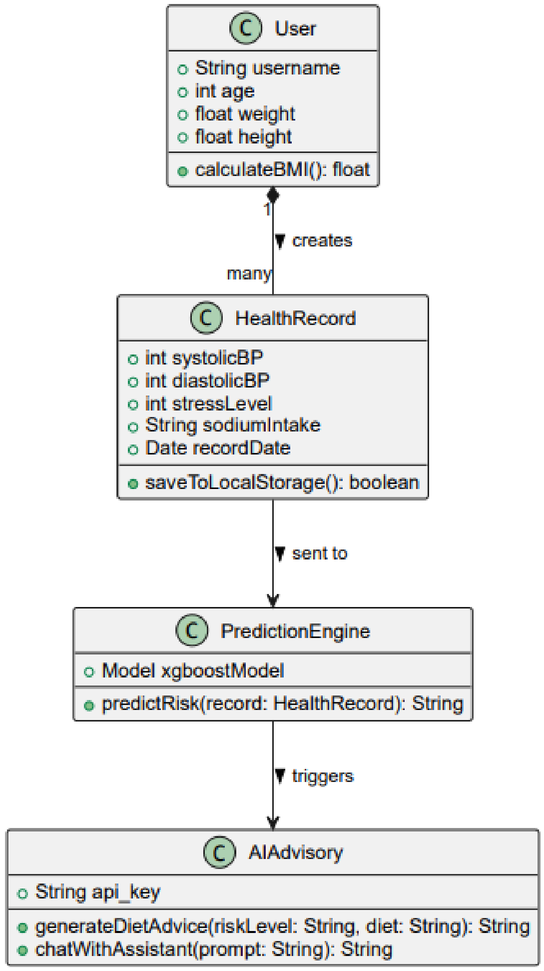

Figure 3.4: Class Diagram showing Entities and Relationships

3.4.5 Deployment Diagram of the Proposed System

The Deployment Diagram is a graphical representation of the physical architecture and the execution environment that the HyperSense application is implemented in, showing the mapping of the software artifacts to the hosting nodes.

Since HyperSense is a recent, decoupled web app, the deployment is spread across more than one physical environment. The Client Node is the mobile or desktop browser of the user, with LocalStorage to store information in an isolated, client-side fashion. The presentation (HTML/CSS/JS) is hosted on a high-performance network of static web server (like Vercel). The backend API, which consists of the Flask Server and the XGBoost model (.pkl file) in a serialized form, is hosted on a separate cloud computing service (like Render) specialized in running Python code. Lastly, the system is linked to the Google Cloud node with the Gemini LLM, via the internet, creating a complete distributed network of the system.

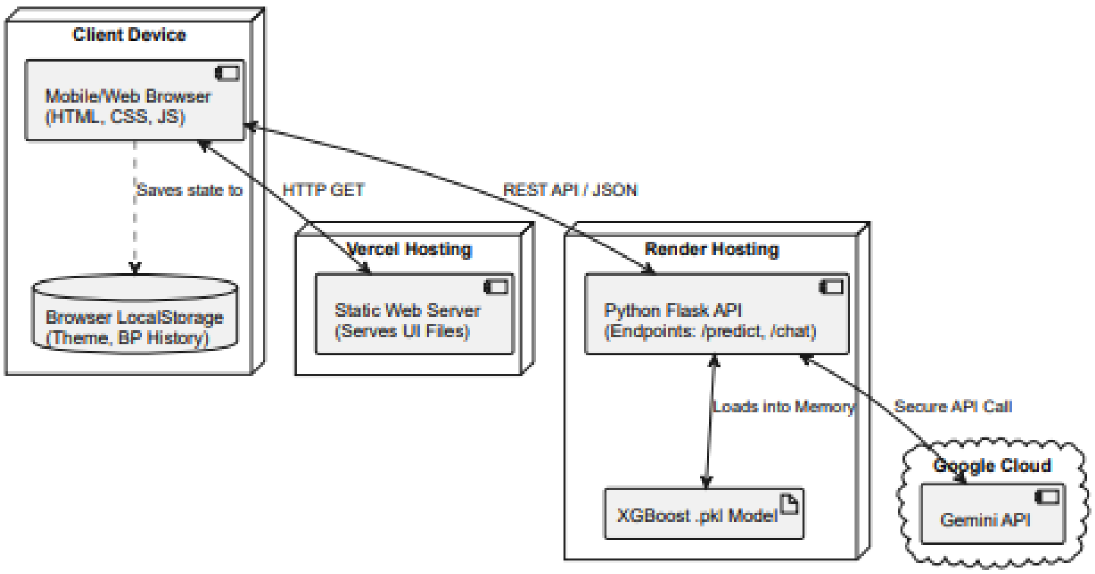

Figure 3.5: Deployment Diagram of the Distributed Web Architecture

3.5 System Design

System design is the transition phase of the system where the requirements developed are converted to a logical and physical blueprint of software construction. In the case of the HyperSense application, a Client-Server (Decoupled) Architecture was followed. This is a design pattern that makes the separation of concerns between the presentation layer (what the user sees) and the business logic layer (where the machine learning computations are done) strict.

There are three basic levels of the system architecture:

1. Presentation Tier: Built using HTML5, CSS3 and JavaScript, with a mobile-first design. A user-friendly, responsive UI on all devices, client-side validation and asynchronous API calls.

2. Application Tier: Using Python and Flask, it operates as a controller, sending data from the front end to an XGBoost model for risk classification, and then to the Google Gemini API for contextual health advice.

3. Storage Tier: Stores user preferences (such as light/dark theme) and previous blood pressure readings using PostgreSQL.

3.5.1 Input and Output Analysis

1. Input Design

The design of input in software engineering, especially machine learning, is a crucial stage since predictive model accuracy is strictly dependent on the quality of entered by the user data (the so-called Garbage In, Garbage Out rule). The HyperSense input interfaces are easy to use, which reduces cognitive load and eliminates errors in inputting data.

The system has the following main input mechanisms:

1. **The Risk Assessment Form:** This is a modal based, dynamic form activated by the main dashboard. It safely fields the multidimensional parameters needed in the XGBoost model. The fields include:
   a. **Physiological Metrics:** Systolic and Diastolic Blood Pressure (mmHg), Age, Weight, and Height (calculates automatically Body Mass Index - BMI).
   b. **Lifestyle Metrics:** Sliding scale of everyday Stress Levels (1-10) and dropdown lists of localized eating habits (e.g., how often you eat a high-sodium bouillon cube).
2. **The AI Chatbot Interface:** a chatbox that enables users to search natural language queries about their health, medication, or diet and send them to the Gemini LLM.
3. **Input Validation:** To curb the backend on handling invalid data, strict client-side validation is imposed with the use of JavaScript. To illustrate, the system will reject entry when a user inevitably enters a lower Systolic BP than his Diastolic BP, or when the numerical figures do not make any medical sense.

3.5.2 Output Design

HyperSense output design is designed to map the complex mathematical classifications of XGBoost model and natural language generation of AI into simple, practical, and visual feedback that can be understood by a non-medical user. The idea is to educate the user about his or her health condition without making him or her panic needlessly, and give him or her actionable measures to do so.

The system delivers the results using the following interfaces:

1. **Dynamic Risk Badges:** Instant, color coded (Green = Normal, Yellow = Elevated, Red = High).
2. **Contextual AI Advisories:** The culturally relevant health advice will be displayed below the badge, and avoid using technical jargon and instead propose locally available alternatives (e.g., locust beans, Iru, Ogiri).
3. **Visual Tracking Dashboard:** Interactive graphics such as SVG activity rings for daily water use and a tracking calendar with a grid to reinforce medication adherence with positive visual cues.
4. **Historical Trend Graphs:** Users can view past blood pressure readings in a list or sparkline graph and use them to monitor cardiovascular trends and then share with their physicians.

3.5.3 Database Design

HyperSense system uses PostgreSQL, a state-of-the-art, enterprise-scale Relational Database Management System (RDBMS) to guarantee the integrity and scalability of data and to ensure the secure management of users. The database will be built so that it will constantly store user demographics, past physiological readings, risk categories, and advisories provided by AI.

Data Storage Structure and Access Method

Structured organization of information is achieved by the use of a relational storage model with tables, rows, columns, data types and constraints. It uses Flask-SQLAlchemy ORM to abstract the raw SQL queries from the Python (Flask) backend, and connects to the PostgreSQL database. This way, blood pressure readings can be accessed or inserted safely via Python functions instead of the potentially insecure direct method. This design is consistent with the OOAD principles, and the database access is also safe and efficient, as it also has built-in protection against SQL injection attacks.

Relational schema (Table and Field)

The database is normalized in order to remove the redundancy of data and consists of three major tables:

1. users Table

The demographic and baseline physiological data of the patient is saved in this table.

Field	**Data Type**	**Description**
user_id	UUID (PK)	It is a unique identifier for every user.
username	VARCHAR	It displays the name of the user.
age	INT	Stores age of the user in years.
weight_kg	FLOAT	Stores users weight in kilograms.
height_m	FLOAT	Stores users height in meters (for BMI)
created_at	TIMESTAMP	Stores date and time when user account was created.

Table 3.1: users Table

2. health_records Table

The table records all the individual prediction requests that the user makes, including the exact parameters that were passed to the XGBoost model and the risk that was returned.

Field	Data Type	Description
record_id	UUID (PK)	A unique identifier for the particular assessment.
user_id	UUID (FK)	It links the records to the particular user.
systolic_bp	INT	Stores higher value of blood pressure.
diastolic_bp	INT	Stores lower value of blood pressure.
stress_level	INT	Stores numerical values of stress score.
sodium_intake	VARCHAR	This stores local diets as string values.
computed_risk_level	VARCHAR	Shows results of the XGBoost model.
recorded_at	TIMESTAMP	Shows exact time the prediction was made.

Table 3.2: health_records Table

3. AI_advisories Table

This table stores the local health and dietary recommendations produced by the Google Gemini API given a particular health record.

Field	Data Type	Description
advisory_id	UUID (PK)	A unique identifier for the AI-text block.
record_id	UUID (FK)	Connects advisory to the specific health block that triggered it.
advisory_text	TEXT	Full natural language recommendation from the LLM.
generated_at	TIMESTAMP	Timestamp of when the API generated the advice.

Table 3.3: AI_advisories Table

Entity Relationships

The database has strong relationships to ensure referential integrity:

1. **One-to-Many (1:N):** A User may have numerous HealthRecords. The latter enables the application to query history and draw trend graphs of the blood pressure of a patient in a few months.
2. **One-to-One (1:1):** One HealthRecord produces a single AIAdvisory. This guarantees that a particular dietary feedback is inseparably linked to the precise physiological parameters which required it.

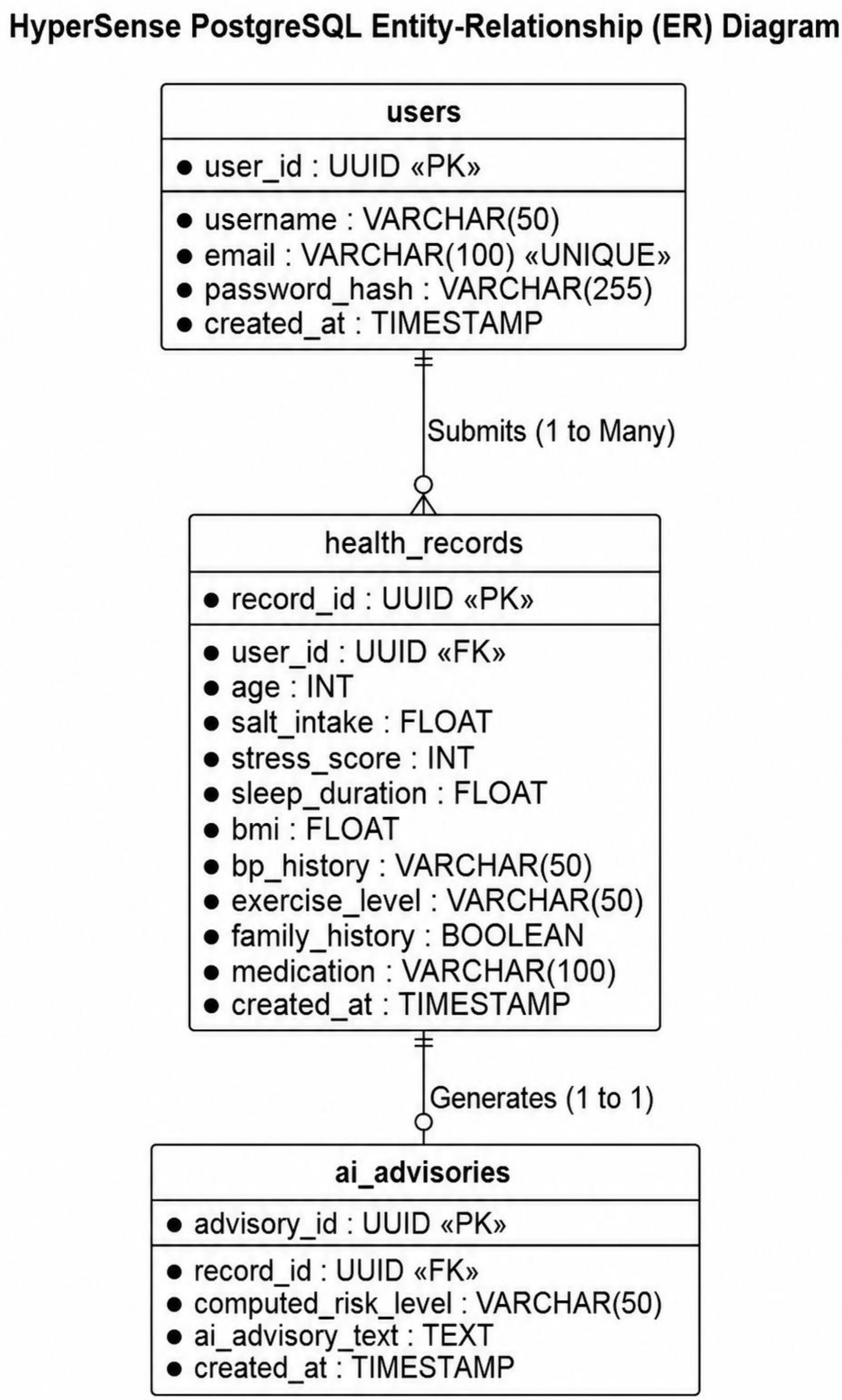

Figure 3.6: HyperSense PostgreSQL Database entity-relationship diagram.

3.6 Dataset Description and Preprocessing

The quality, structure and cleanliness of the data that the XGBoost algorithm is trained on are fundamentally important to the predictive accuracy of the XGBoost algorithm. This section describes the features of the dataset that has been used to run the HyperSense predictive engine and the systematic transformations that have been made on the dataset to prepare it to be fed into a machine learning system.

3.6.1 Overview of the Dataset

In order to train the hypertension risk classification model, a structured, tabular medical dataset (hypertension_dataset.csv) was used. This data set has multidimensional records of patients, a holistic picture of their physiological conditions and lifestyle.

The dataset consists of 10 independent features column and 1 dependent target column, which are as follows:

1. **Demographic & Physiological:** Age, BMI (Body Mass Index), and Family_History (genetic predisposition).
2. **Clinical Features:** BP History (historical baseline) and Medication (current prescriptions such as ACE Inhibitors or Beta Blockers).
3. **Lifestyle & Dietary Features:** Salt_Intake (critical in localized dietary modeling), Stress_Score (rated numerically), Sleep_Duration, Exercise_Level and Smoking_Status.

4. **Target Variable:** Has_Hypertension, a dichotomous categorical variable that shows whether the patient will ultimately be classified as hypertensive (Yes or No).

3.6.2 Data Preprocessing Steps

Machine learning algorithms, such as XGBoost, are able to calculate mathematical equations, and cannot directly interpret raw alphabetical text (strings). Thus, intensive data preprocessing was performed with the help of Pandas and Scikit-Learn libraries in Python in order to convert the set of data into a purely numerical format.

The following critical steps were involved into the preprocessing pipeline:

1. **Binary Encoding, Target Variable Mapping (Binary Encoding):** The dependent variable (Has_Hypertension) was transformed into binary integers by changing the text string labels into binary values. The Yes instances were assigned the value of 1 (positive class), and those of the No instances were assigned the value of 0 (negative class).
2. **Dealing with Categorical Variables (One-Hot Encoding):** The data set had various nominal categorical variables, including Smoking_Status (Smoker/Non-Smoker) and Medication types. With pd.get_dummies() function, these text columns were converted to distinct binary columns (0s and 1s). This is to make sure that the model does not make false assumptions about mathematical hierarchy between the various text categories.
3. **Data Type Standardization:** Although XGBoost is a tree-based algorithm, and does not inherently require the extreme normalization or scaling (such as Min-Max scaling) of the data to which it is applied, all numerical columns (Age, BMI, Salt_Intake) were strictly cast to float or int data types to prevent the execution errors during the model fitting stage.

3.6.3 Training and Validation Split

To make sure that the model was able to generalize to new unseen patients, instead of just memorizing the dataset (a weakness called overfitting), the preprocessed data set was split using scikit-learns train test split function.

The ratio of 80:20 split was used:

1. **Training Set (80%):** Eighty percent of the preprocessed data was used for training the XGBoost model, which helps build the relationships between physiological and lifestyle factors and hypertension risk.
2. **Testing Set (20%):** The rest 20% of the data set was used to test the model with unseen data. This set is employed to evaluate the predictive performance and to create evaluation measures like the Confusion Matrix and the ROC Curve.

To prevent overfitting and ensure the model's generalizability to unseen data, the dataset was divided into 80% training and 20% test sets. The dataset was partitioned into an 80% training set and a 20% test set to enhance the model's ability to apply to new cases and to minimize overfitting. To enable the results to be reproduced consistently, a fixed random state was used.

3.6.4 System Architecture Design

The system architecture is a high-level structural description of the HyperSense application, which shows how the different components in the application, sub-systems, and external services interrelate to achieve the functional requirements. The project is based on a web-based, client-server model based on a modern N-Tier (Multi-Tier) Decoupled Architecture.

This architectural paradigm is crucial so that the computational complexity of the machine learning algorithms does not become a bottleneck of the user interface that would lead to a scalable, modular, and highly responsive system. The architecture can be broadly divided into the following levels:

1. The Presentation Tier (Client/Frontend):

This is the front-end of the application which is a mobile-first, responsive web interface. It is developed purely based on the standard web technologies (HTML5, CSS3 and modern Vanilla JavaScript). This level will render the dashboard, receive user inputs (vital signs and lifestyle metrics), and do initial data validation on the client-side, as well as present the obtained AI recommendations and badges of risk. It asynchronously communicates with the backend through RESTful API calls through JavaScript fetch API.

2. Application / Logic Tier (Server/Backend):

As the main player, this layer is driven by Python and a lightweight web framework called Flask. It deals with the system business logic and routing logic.

- **API Endpoints:** It provides specific endpoints (e.g., /predict, /chat) where the frontend sends out payloads in a safe and secure format (JSON).
- **Machine Learning Integration:** At this level, the trained XGBoost model (in the form of a serialization pkl file) is loaded into the memory of the server. The Flask controller transmits the cleaned-up patient information into the model and obtains the mathematical risk category.

3. API Services Integration:

The application tier links to external cloud services smoothly to surmount the divide between numerical classification of risk and practical patient care. After receiving a risk score of the

XGBoost model, the Flask server authenticates a prompt and sends remote HTTP POST request to Google Gemini API. The Gemini Large Language Model (LLM) understands the context and sends back to the server a culturally-localized, natural language health advisory.

4. The Data /Storage Tier:

This layer offers long-lasting and secure data storage of the application. It comprises PostgreSQL relational database management system. Application Tier This tier is the one that communicates with this database through the Flask-SQLAlchemy Object-Relational Mapper (ORM) to provide safe transactions with the database without exposing raw SQL queries. This layer safely stores information about users, past blood pressure records and artificial intelligence recommendations to analyze longitudinal trends.

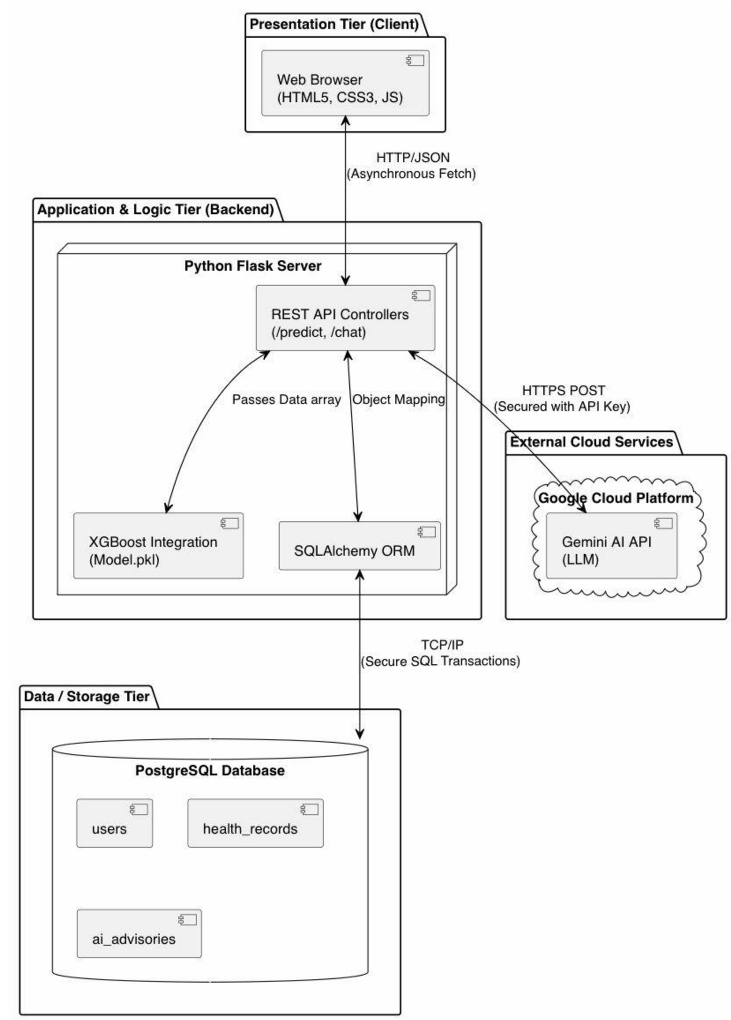

Figure 3.7: HyperSense Application N-Tier System Architecture at high level.

CHAPTER FOUR

SYSTEM IMPLEMENTATION

4.1 Introduction

System implementation is the provision of a fully tested system into production for its daily operation. This chapter connects the theoretical perspective, object-oriented models and architectural blueprints developed in Chapter Three with the actual implementation and deployment of the HyperSense application. It gives a holistic view of system design to system realization. In particular, this chapter outlines the software engineering tasks that were performed during the development of the web-based hypertension risk prediction and advisory system, so that the final product will comply with the specified functional and non-functional requirements to the latter.

Finally, this chapter can be taken as the evidence of concept, as it shows that the conceptualized predictive models and localized health tracking mechanisms were converted into a working, user-friendly software solution successfully.

4.1 Development Environment and Tools

The effectiveness of the HyperSense implementation depended on a well-chosen set of the recent programming languages, libraries and development platforms. The selected technology stack was also able to promote a decoupled architecture, which meant the computational intensity of the machine learning algorithms did not cause the web-based user interface to be responsive.

4.1.1 Programming Languages and Libraries Used

The development of the system can be splitted into the two main language layers as Python as the back-end logic and machine learning, and HTML5,CSS3,JavaScript as the presentation layers.

Why did I Use Python:

There was no doubt about Python being the main programming language to be used in the backend controller and the predictive engine. The choice is supported by the fact that it has the best data science and machine learning ecosystem. Python already has complex mathematical operations needed by decision trees, and at the same time, it includes lightweight web frameworks, which could be used to support API endpoints. In addition, Python has the most powerful and well-maintained Software Development Kits (SDKs) to connect with third-party cloud AI services, including the Google Gemini API.

Libraries and Frameworks Used:

In order to boost the development process and provide stability of the system, there were some open-source Python libraries that were incorporated:

1. **Flask**: A non-blocking Web Server Gateway Interface (WSGI) web application framework that it uses to implement the backend RESTful API, route web traffic, and accept web traffic in the form of JSON.
2. **Flask-SQLAlchemy**: Object-Relational Mapper (ORM) to connect the Python code with the PostgreSQL database so that the database tables can be accessed in a secure and object-oriented style.
3. **XGBoost (Extreme Gradient Boosting)**: The machine learning library that was used to train the classification model. It was selected in comparison to the traditional models (such as Logistic Regression) because of its better ability to process tabular medical data and the speed of its execution.
4. **Scikit-Learn**: It is used when training the model, to preprocess data, and divide it into training/testing sets, as well as to create evaluation metrics (Confusion Matrix and ROC AUC scores).
5. **Pandas & NumPy**: Data manipulation libraries that are essential in cleaning, parsing, and structuring the hypertension data in hypertension dataset.csv before being fed into the predictive algorithms.
6. **Matplotlib & Seaborn**: Data visualization tools used in the research stage to plot the learning curves, precision-recall graphs, and feature importance charts.
7. **SHAP (SHapley Additive exPlanations)**: SHAP may be used to explain Explainable AI as a visual representation of the influence of each patient variable (e.g., Sodium Intake vs. Age) on the final decision of the XGBoost model.
8. **google-generativeai-sdk Google Generative AI SDK (google-generativeai)**: The official library to safely send the XGBoost risk classifications to the Gemini LLM to generate localized dietary advice.

4.1.2 Development Platform and Hardware Requirements

The development life cycle employed local and cloud-based Integrated Development Environments (IDEs) to manage the different stages of software engineering and machine learning.

Development Platforms (IDEs):

1. **Visual Studio Code (VS Code):** Main Development Environment for Frontend and Backend Development, providing Code Management, Version Control, and Project Organization.
2. **Google Colab:** For data preprocessing, model training, testing, and evaluation of the XGBoost algorithm before exporting the trained model to the Flask application, during the machine learning phase.

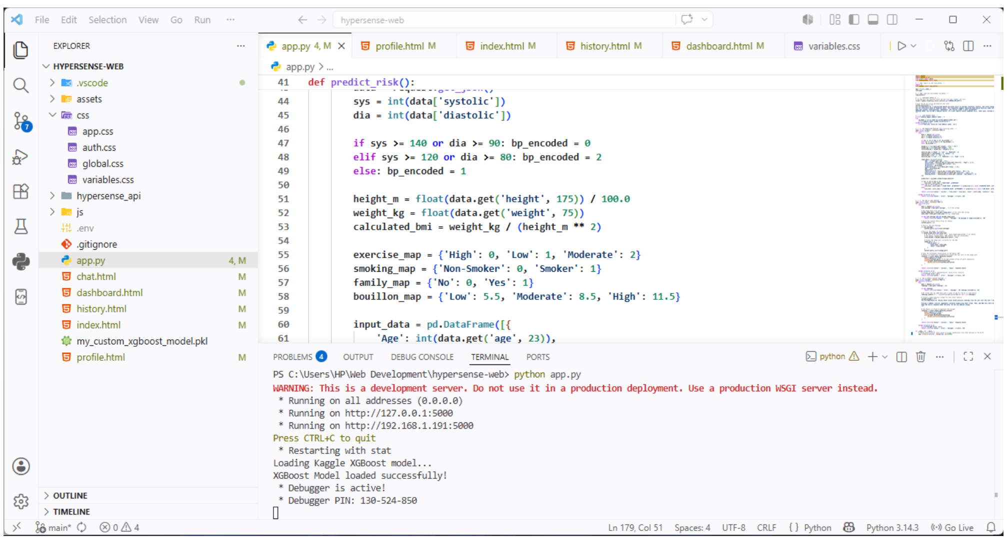

Figure 4.1: VS Code IDE showing HyperSense Backend Architecture

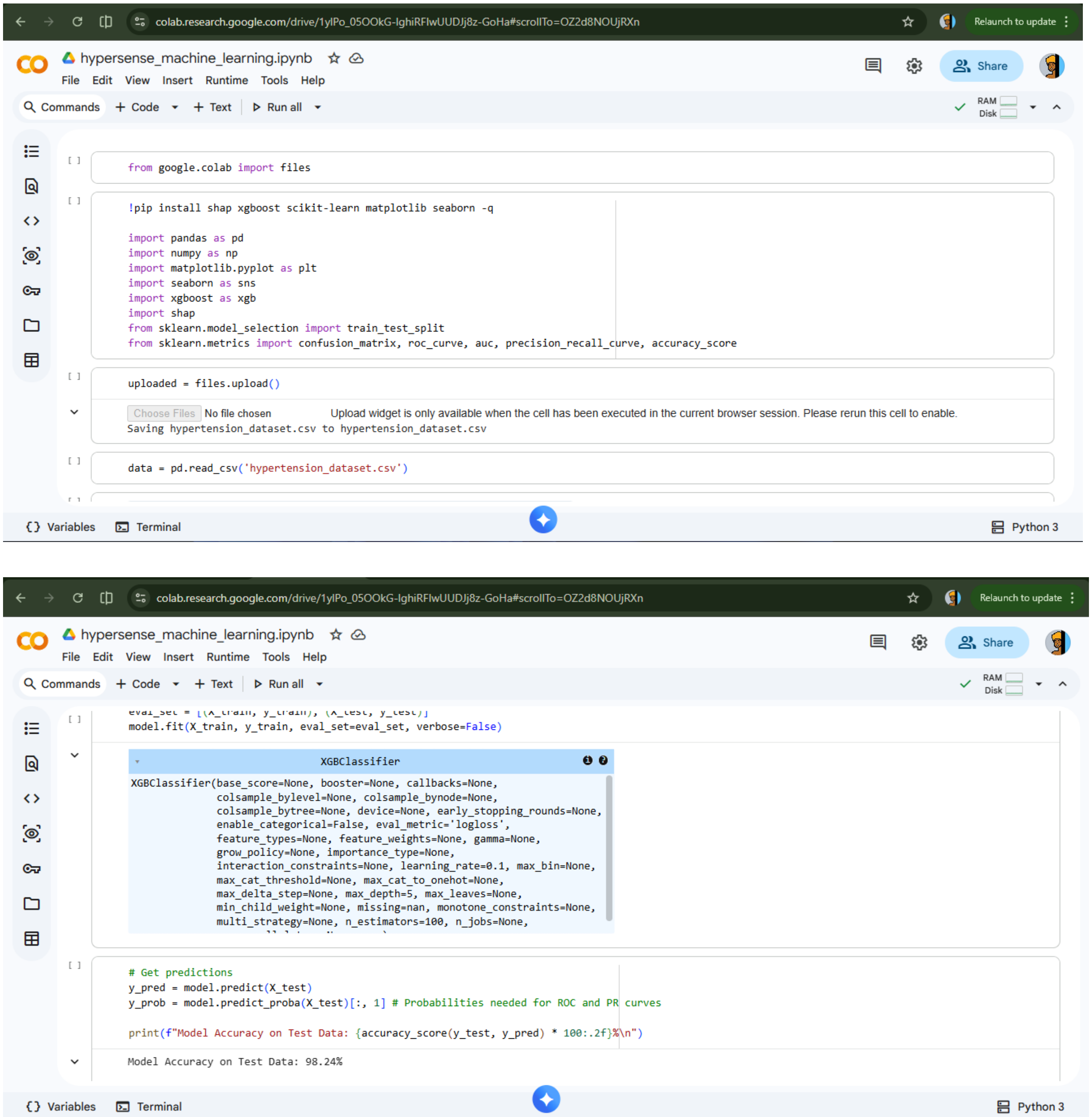

Figure 4.2: Google Colab Environment used for XGBoost Model Training and Evaluation

Hardware Specifications:

The application was locally developed and tested on a personal computer with the following hardware specifications, to make the local Flask server and browser rendering run smoothly:

1. **processor:** Intel Core i3/AMD Ryzen 3 or newer to enable asynchronous API threading.

2. **RAM (Memory):** 8 GB (16 GB Recommended) to be able to run the IDE, local postgreSQL server, browser debugging tools and Python environment simultaneously.

3. **Storage:** 256 GB Solid State Drive (SSD) which can be retrieved in a timely fashion through reading and writing of the database and software requirements.

4. **GPU (Graphics Processing Unit):** The XGBoost tree-building step of the web application took place on a virtual machine with the NVIDIA T4 Cloud GPU to speed up the learning phase of the training process.

4.1.3 System and Hardware Requirements

Since the HyperSense system is implemented as a decoupled, cloud-based web application, the intensive computational power, including executing the XGBoost machine learning model and querying the Gemini LLM, is outsourced to off-the-shelf backend servers. As a result, there is a low amount of hardware and resource requirements on the device of the end-user.

Nevertheless, the baseline requirements have to be met in order to guarantee the maximum performance, flawless asynchronous API communication, and the correct display of the interactive dashboard. These specifications can be divided into Client-Side (End-User) and Server-Side (Deployment) requirements.

Client-Side (End-User) Requirements

To effectively use and communicate with the HyperSense application, the user needs to have the following requirements in his or her device:

1. **Internet-Enabled Device:** A working smartphone, tablet, laptop, or desktop computer.
2. **Modern Web Browser:** A modern browser like Google Chrome, Mozilla Firefox, Microsoft edge, or Apple Safari.
3. **JavaScript Enabled:** The browser is required to support JavaScript execution, the frontend needs it to do client-side validation and to render dynamic SVG activity rings and to make asynchronous fetch requests to the backend.

4. **Reliable Internet Connection:** A stable Internet connection (at least 3G, though preferably 4G/Wi-Fi) is absolutely necessary to avoid blocking out upon errors of connection to the external Google Gemini API to retrieve real-time dietary advice.

Server-Side (Deployment) Requirements

To successfully host, run and maintain the system by the administrator, the software requirements of the backend environment (e.g., Render or a local development machine) must satisfy the following requirements:

1. **Python Environment:** Python 3.9 or later is required to run the backend logic, and modern machine learning libraries.
2. **Web Framework:** Flask web framework (not Django) should be installed and correctly configured to serve the API endpoints.
3. **Memory (RAM):** The hosting server should have at least 512MB to 1GB of RAM to store and load the serialized XGBoost model (.pkl file) in memory to make quick inferences.
4. **Database Instance:** An operational instance of a PostgreSQL database to store the persistent data on user profiles and historical health records.

4.2 Model Architecture and Training

A machine learning classification algorithm powers the predictive core of the HyperSense app. The proposed system processes structured medical data (in tabular format) while Convolutional Neural Networks (CNNs) are usually used for unstructured image processing. So, it was decided to go with ensemble learning architecture as it has been seen to perform better on tabular data, is faster to train and more interpretable in medical scenarios.

4.2.1 Details of Model Architecture

It adopts the algorithm of Extreme Gradient Boosting (XGBoost). XGBoost is an optimized distributed gradient boosting library, designed to be highly efficient and flexible. XGBoost is based on the architecture of Decision Tree Ensembles. The model builds upon multiple sequential “weak learners” (shallow decision trees) rather than a single large predictive tree.

Gradient Boosting Mechanism: The models are created sequentially in the architecture. The new trees are trained to overcome the remaining inaccuracies (mistakes) of previous trees.

Inference: When a patient enters their health information (e.g., Systolic BP, Sodium Intake, Stress, etc.) into the HyperSense dashboard, that information flows down all of the trees in the ensemble. All of the individual trees combine their prediction to create a very accurate and strong "strong learner" that is the final hypertension risk classification.

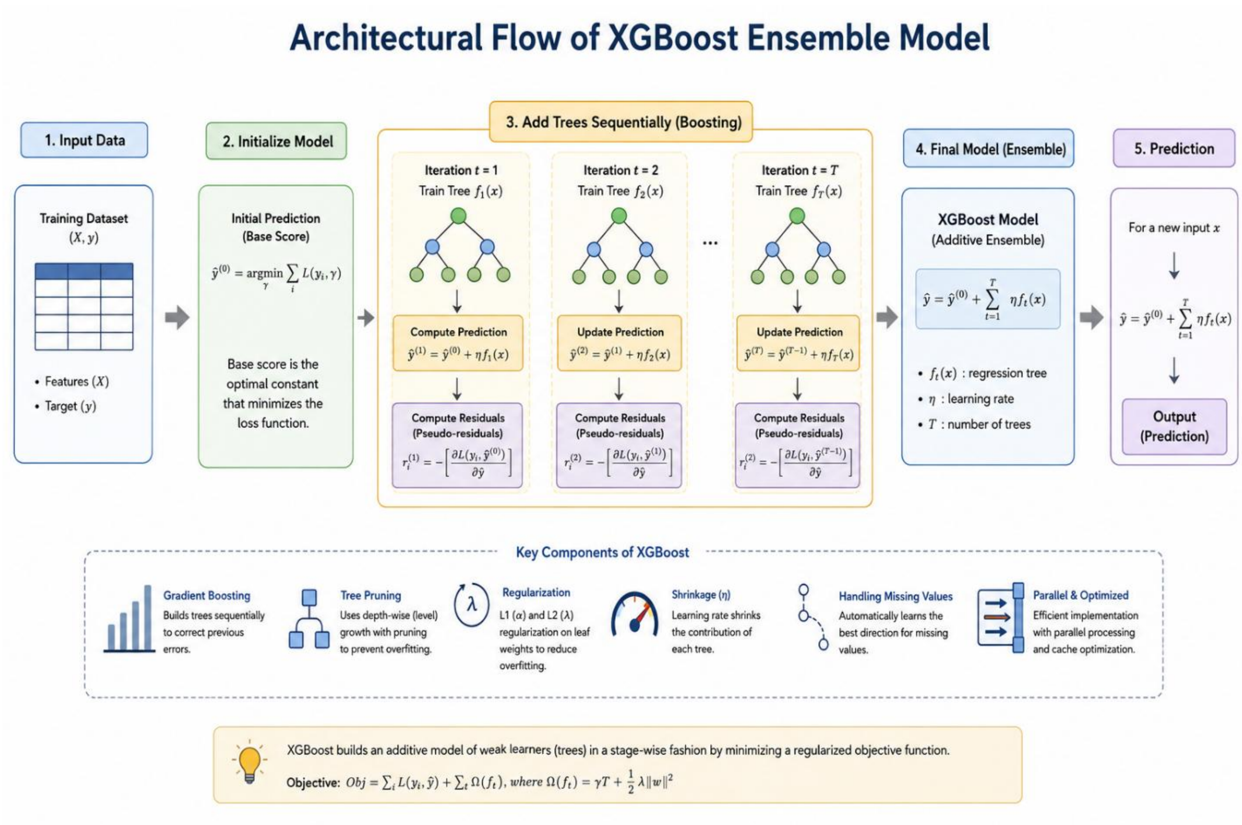

Figure 4.3: Architectural flow of XGBoost Ensemble Model

4.2.2 Training Parameters (Hyperparameter Tuning)

In order to maximize the accuracy of the model without overfitting, some hyperparameters were tuned during the training phase, using the Python API of the XGBoost model and the Scikit-Learn Python library. The following training parameters were used:

1. **Number of Estimators (n_estimators = 100):** similar to the number of epochs in a neural network. Specifies the limit on the number of decision trees to construct in sequence. The value of 100 was selected, as a compromise between speed and accuracy.
2. **Learning Rate (learning_rate = 0.1):** Also called eta, this is a parameter that reduces the weights of the features after each step. To make the model stronger, by slowing down the learning process, it was decided to use a lower learning rate, 0.1, so that the 100 trees were able to learn the medical correlations carefully and not going to fast conclusions.
3. **Maximum Tree Depth (max_depth = 5):** This is the maximum depth of each individual decision tree. To avoid capturing ‘noise' and very particular edge cases in the data used to train the model, a depth of 5 was enforced to prevent this.
4. **Loss Function (eval_metric):** The evaluation value which will be optimized when training. This means that if the model is quite certain that the patient is "High Risk" but they are actually "Normal", then that is a serious error and it will be heavily penalized.

4.2.3 Modeling Serialization and Integration

After a successful training and validation process in the Google Colab environment, the completely trained model was then serialized (saved) to a binary file format in Python (model.pkl).

This was a serial architecture, which was then exported and coupled with the Flask back-end of the HyperSense application. This way, the web server won't have to retrain the algorithm every time a user logs in, but it can just load the pre-trained .pkl file into the web server's RAM to generate risk predictions on the fly.

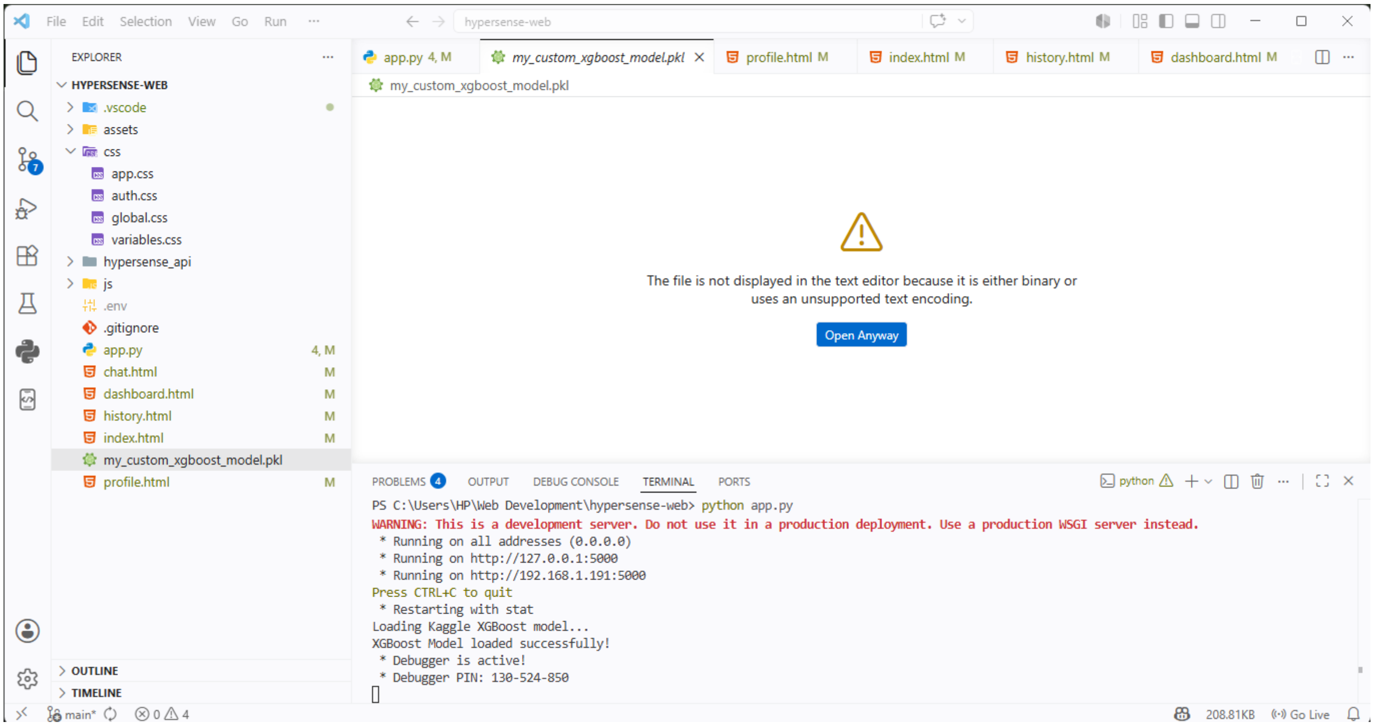

Figure 4.4: Integration of the Serialized XGBoost Model into the Flask Backend

4.3 System Implementation and Modules

Chapter Three laid the groundwork for the HyperSense application's theoretical framework, but this section outlines the actual development of the functional modules of the system. The application was designed in a very modular and decoupled fashion. The machine learning engine, the user interface and the conversational AI are implemented as independent operation modules, allowing for quick response times and high fault tolerance.

The implemented system has four main interacting modules, focusing on the core functionality.

4.3.1 Patient Data Acquisition Module

It is the primary means of interaction between the system and user. It is based on responsive HTML5, CSS3 and JavaScript and is used as the main data collection platform.

Implementation

The module uses interactive modal forms to safely gather in real-time, physiological data (Systolic/Diastolic BP, Age, Weight) and categorical lifestyle data (Stress Levels, Sodium intake/Bouillon cube).

Interaction

The JavaScript code on the client side is very strict and will not allow empty or impossible values to be submitted to the server (e.g. diastolic bp greater than systolic bp). This means that the backend cannot crash when the data is being transmitted, and it ensures that there are no problems in transmitting the data from the user's device.

4.3.2 The XGBoost Classification Engine (Backend Controller)

This is the main processing of the app written in Python with the Flask module. It connects the world of web communications to machine learning inference.

The Flask routing module catches the JSON payload of the user's health data sent from the frontend by a RESTful POST request. It transforms this data into a format, that is 2D array, necessary for the use in the Scikit-Learn pipeline.

Then the module loads the trained XGBoost algorithm (model.pkl) into memory which is serialized.The module then loads the serialized model.pkl (trained XGBoost algorithm) into memory. It feeds the structured array into the model's predict() function, which creates complex decision trees and provides an instantaneous categorical risk classification (such as "Normal", "Elevated" or "High Risk").

4.3.3 The AI Advisory and Conversational Module

This module sought to design and implement a system that was able to convert the numerical risk outputs into practical, culturally relevant health advisories, rather than a mere mathematical "black box. This module aimed to design and implement a system that was able to translate the numerical risk outputs into practical, culturally relevant health advisories, not just a mathematical "black box.

Implementation: This module is based on the official Google Generative AI Python SDK. It dynamically builds an unseen and user personalized text prompt with his/her particular dietary practices, such as eating local Nigerian seasoning, and their recently calculated XGBoost risk level.

Interaction: The Flask backend sends an HTTP request to the Google Gemini Large Language Model (LLM), which is an asynchronous call that happens securely. The LLM is used to process the contextual prompt and generate a personalized, natural language paragraph recommending lifestyle changes that the user should undertake, along with the existing risk classification.

4.3.4 Dashboard and Results Display Module

The final module is responsible for displaying the processed information to the end-user.

Implementation: It is a JavaScript module that works totally on the client side and parses the JSON response coming back from the Flask server. It uses DOM (Document Object Model) manipulation to change the user interface without refreshing the web page.

Interaction: It presents the output of the system in user-friendly visual elements, such as Risk Badges (Green, Yellow, Red) according to the standard medical triage indicators, and presents the text advisory generated by the Gemini system in a beautiful way. In addition, this module is connected to the browser's local storage and the PostgreSQL database (through the API) to store the session and return to it historical SVG activity rings and previous medication tracking data.

4.4 Testing and Evaluation

A thorough testing procedure was performed to make sure that the HyperSense application is scientifically accurate, computationally strong, and easy-to-use. This section documents the performance of the machine learning model in terms of classification metrics, after which it is thoroughly tested with user interface and integration tests and unit tests to assess the software architecture.

4.4.1 Evaluation Metrics

The isolated 20% of the validation data (not seen during training) was used to assess the performance of the XGBoost classification engine. The evaluation of only the accuracy was not enough, since medical diagnostic tools are extremely reliable, so precision, recall and F1-score were also measured.

1. **Accuracy (98.24%):** An overall accuracy of 98.24% was obtained which suggests that the model has a very good accuracy in the classification of the vast majority of patient records into the correct risk categories (Normal, Prehypertension, or Hypertension).

2. **Precision:** The percentage of correct positive identifications. The high precision will not trigger false alarms in healthy patients.
3. **Recall (Sensitivity):** Measures the percentage of actual cases of hypertension that were identified. In healthcare, high recall is essential to make sure that the patients with a high risk do not escape notice (minimizing False Negatives).
4. **F1-Score:** The harmonic mean of Precision and Recall, uses to balance both measures when there is a class imbalance in the set.
5. **Confusion Matrix:** A graphical representation that clearly describes the number of False Positive, False Negative, True Positive and True Negative results generated by the model.

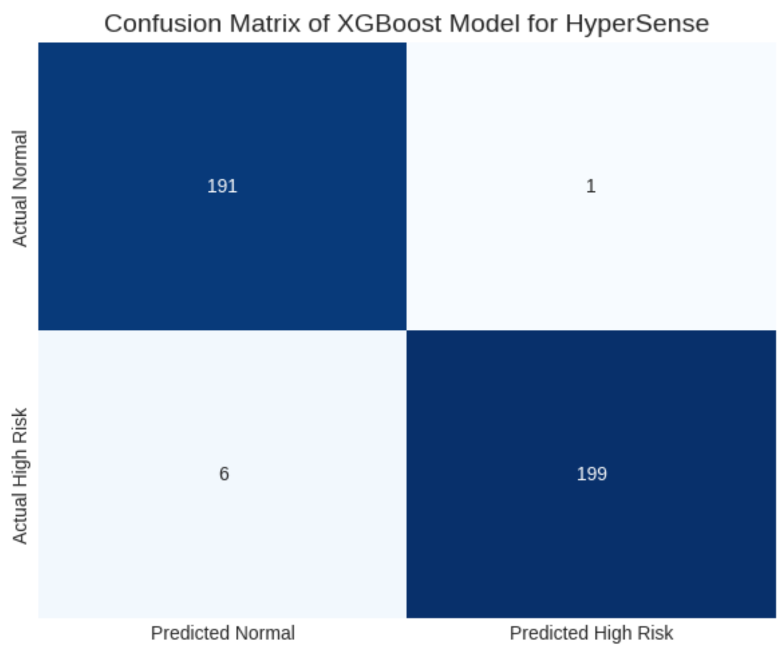

Figure 4.5: Confusion Matrix showing True and Predicted Classifications in an XGBoost Model

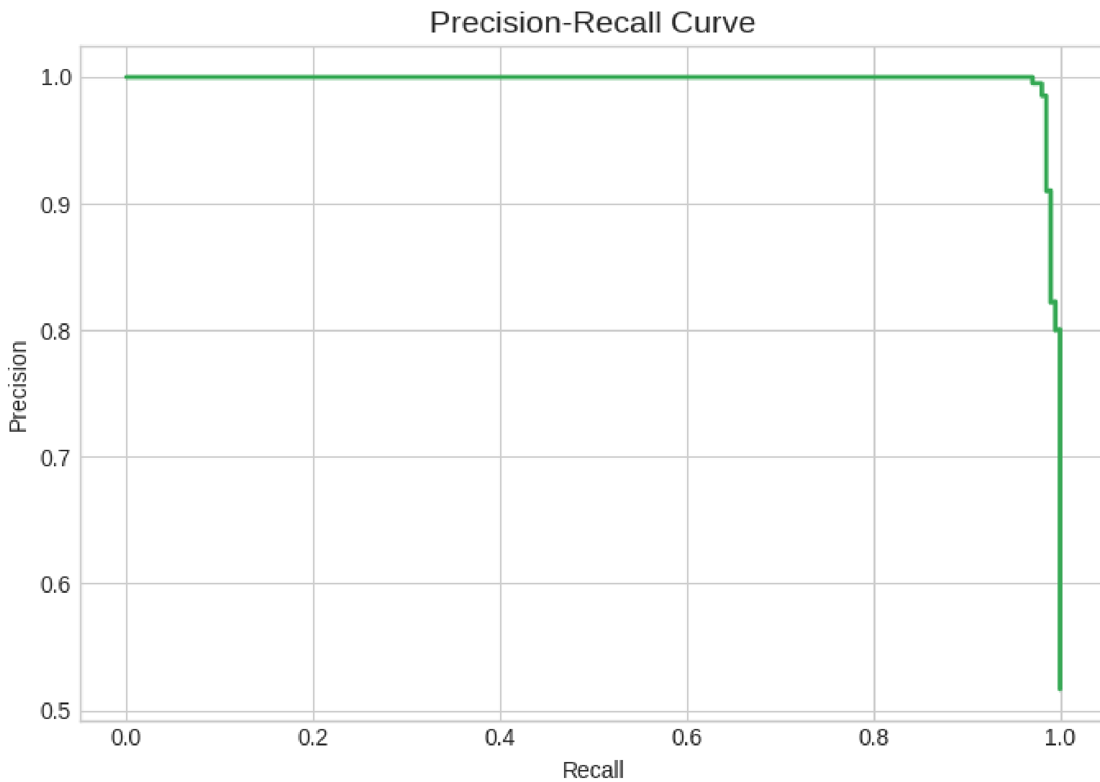

Figure 4.6: Precision Curve showing the model's reliability

4.4.2 System Testing

System testing was performed to ensure an integrated software architecture worked properly. This was split into Unit Testing (testing single components) and Integration Testing (testing communication between different components).

Unit Testing by Module:

i. **Frontend Validation Module:** Intentionally entered incorrect data (letters in the Blood Pressure box, for example, or a higher Diastolic than Systolic). The system passed the test and blocked the submission and gave warning to the client side if there were errors.

ii. **XGBoost Engine Module:** The Python predictive function was tested independently by using a static dictionary of well-known health values and feeding it to loaded model.pkl file. It also managed to get the string result (e.g., 'high risk', etc.) and did not crash the Flask server.

Integration Testing:

i. **Frontend to Backend Integration:** Tested by invoking fetch API for JavaScript. This test was successful, showing that the JSON payload made it across the network to hit the Flask /predict route and received a 200 OK (http) response.

ii. **Backend to Gemini API Integration:** Latency and Response Format of Google Generative AI API were tested. The integration test proved successful: Flask server was able to append the XGBoost risk level to the hidden dietary prompt, securely send to Google's servers with an API key in an environment (.env) file, and successfully parse the returned natural language text without timed out.

4.4.3 User Interface Testing

User Interface (UI) testing focused on visual design, responsiveness, and User Experience (UX) of the HyperSense app for varying device sizes (mobile and desktop viewports).

Risk Assessment Input Form: HyperSense needs precise numerical and categorical input, not image uploads, to be risk assessed. The UI was tested to ensure that the slider to adjust "Stress Levels" and the localized dropdown menus for "Dietary/Sodium Intake" (e.g. Bouillon cube use) were intuitive and easy to use on touch screens.

Dashboard and Results Display: The result rendering was tested to make sure that the numerical predictions were transformed into easily understandable visual aids. The system is now tested to immediately generate the correct Color-Coded Triage Badge (e.g., Red = High Risk) and to nicely format the Markdown text that the Gemini AI returns into readable paragraphs without disrupting the CSS layout.

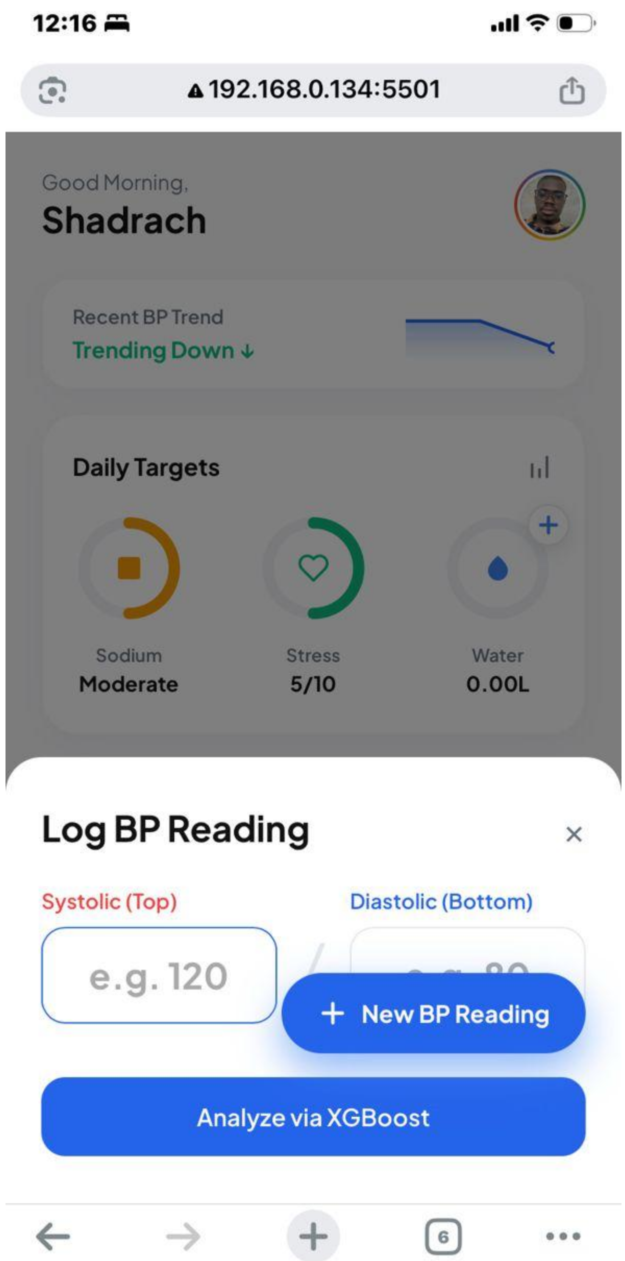

Figure 4.7: The Patient Data Input Form features text boxes that are placeholder-labeled

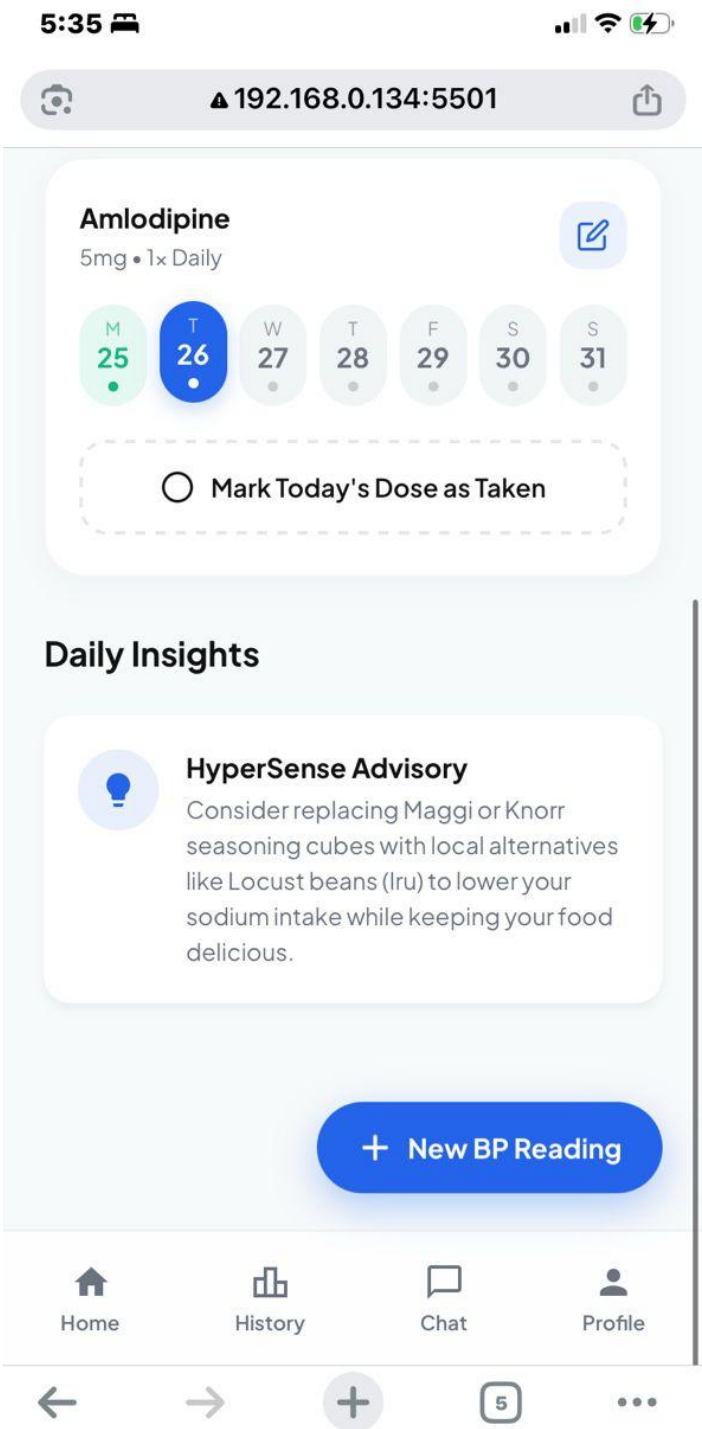

Figure 4.8: Labelled UI Component of the Results Display Page with Triage Badge and Gemini Advisory

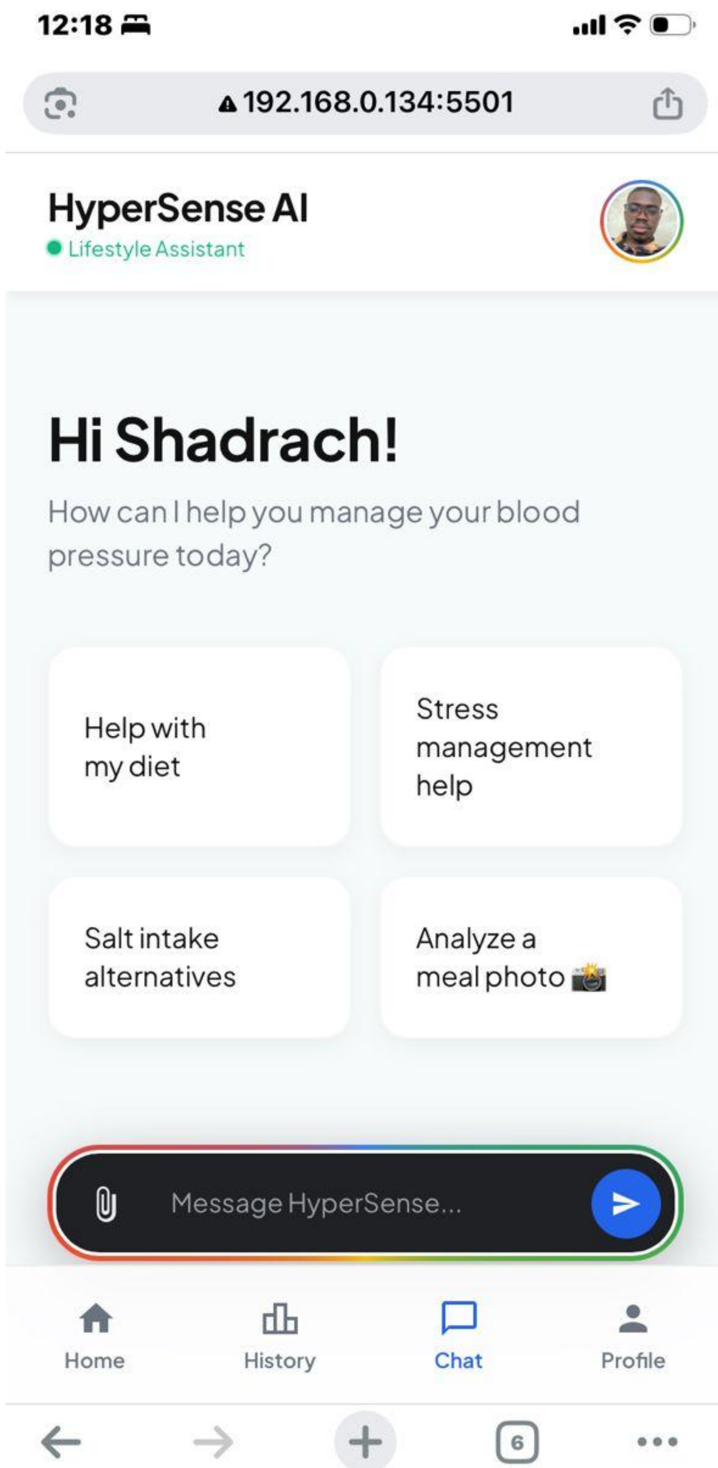

Figure 4.9: Natural Language Input Interface for the Chat AI Assistant

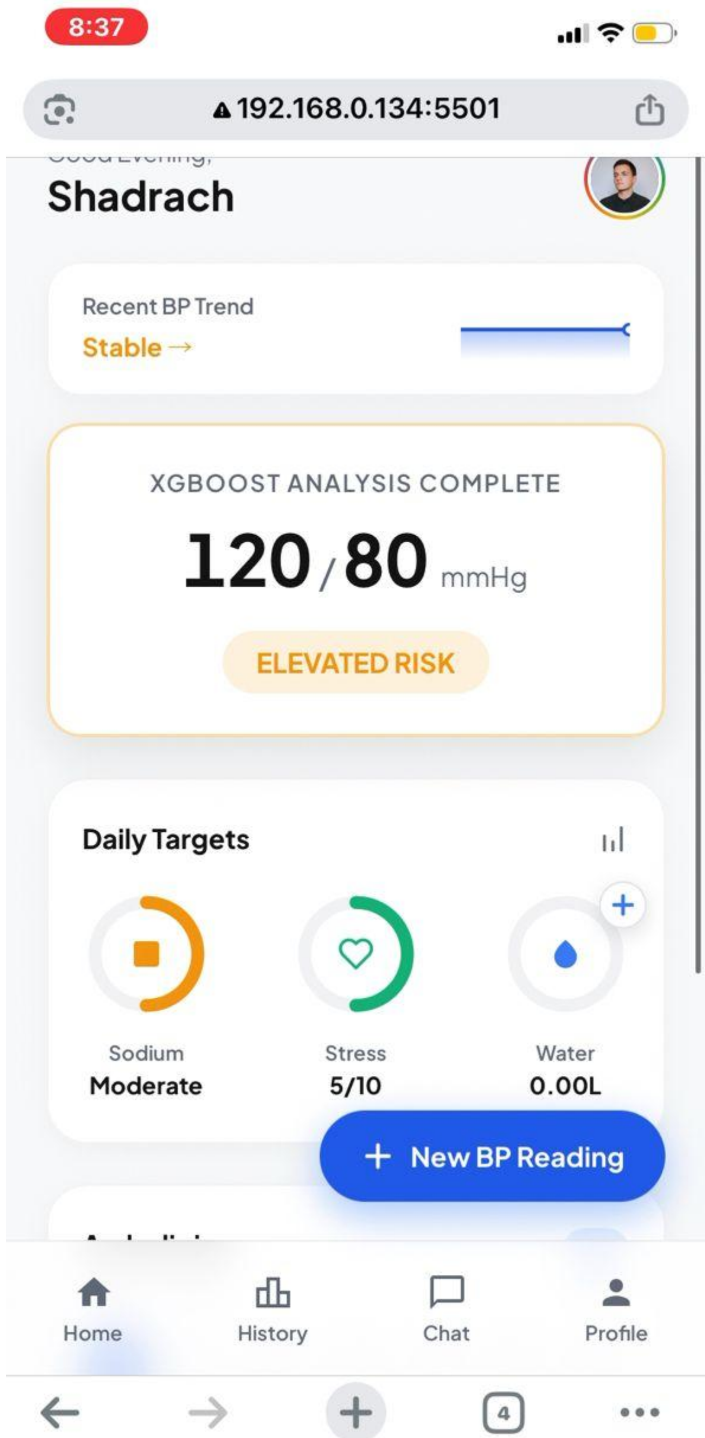

Figure 4.10: Dashboard Showing the Machine Learning Risk Classification Output

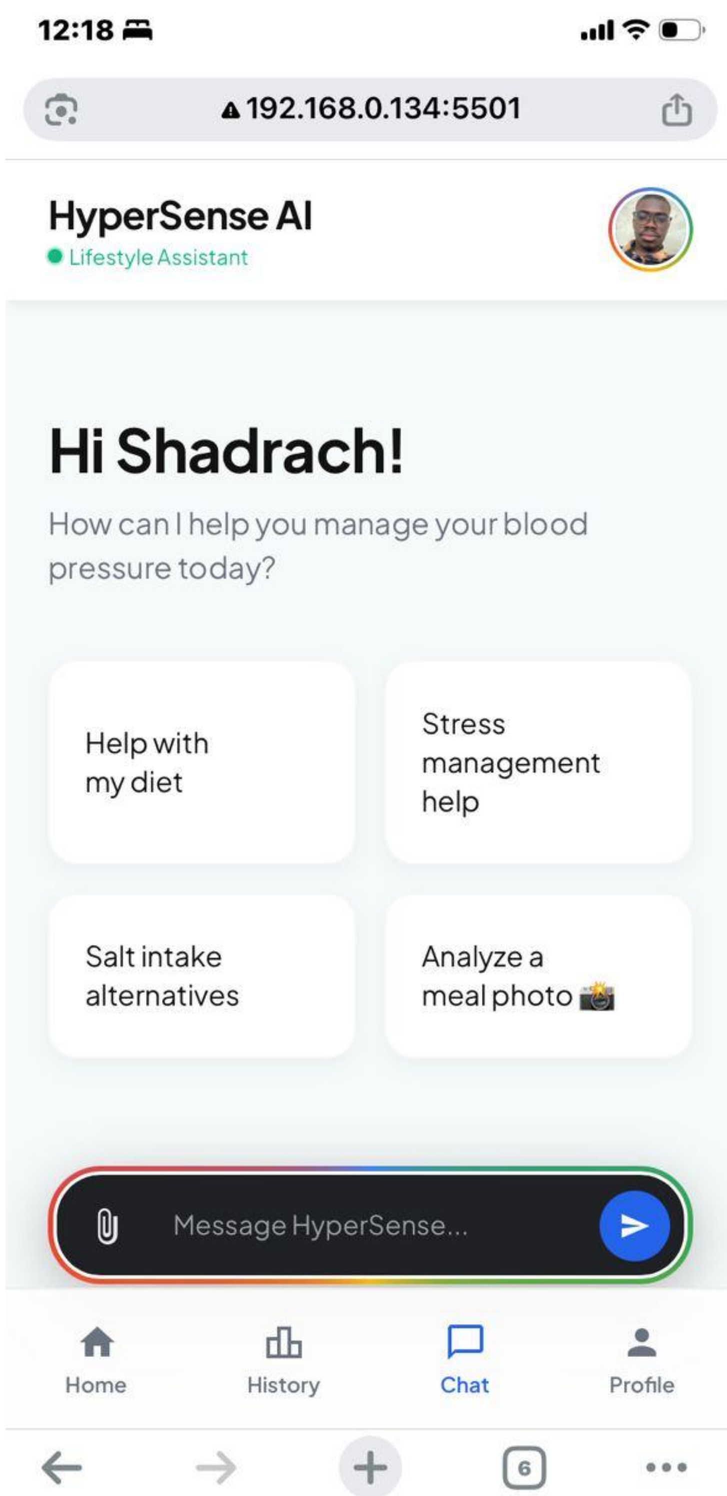

Figure 4.11: AI-Generated BP Health Advisory Output

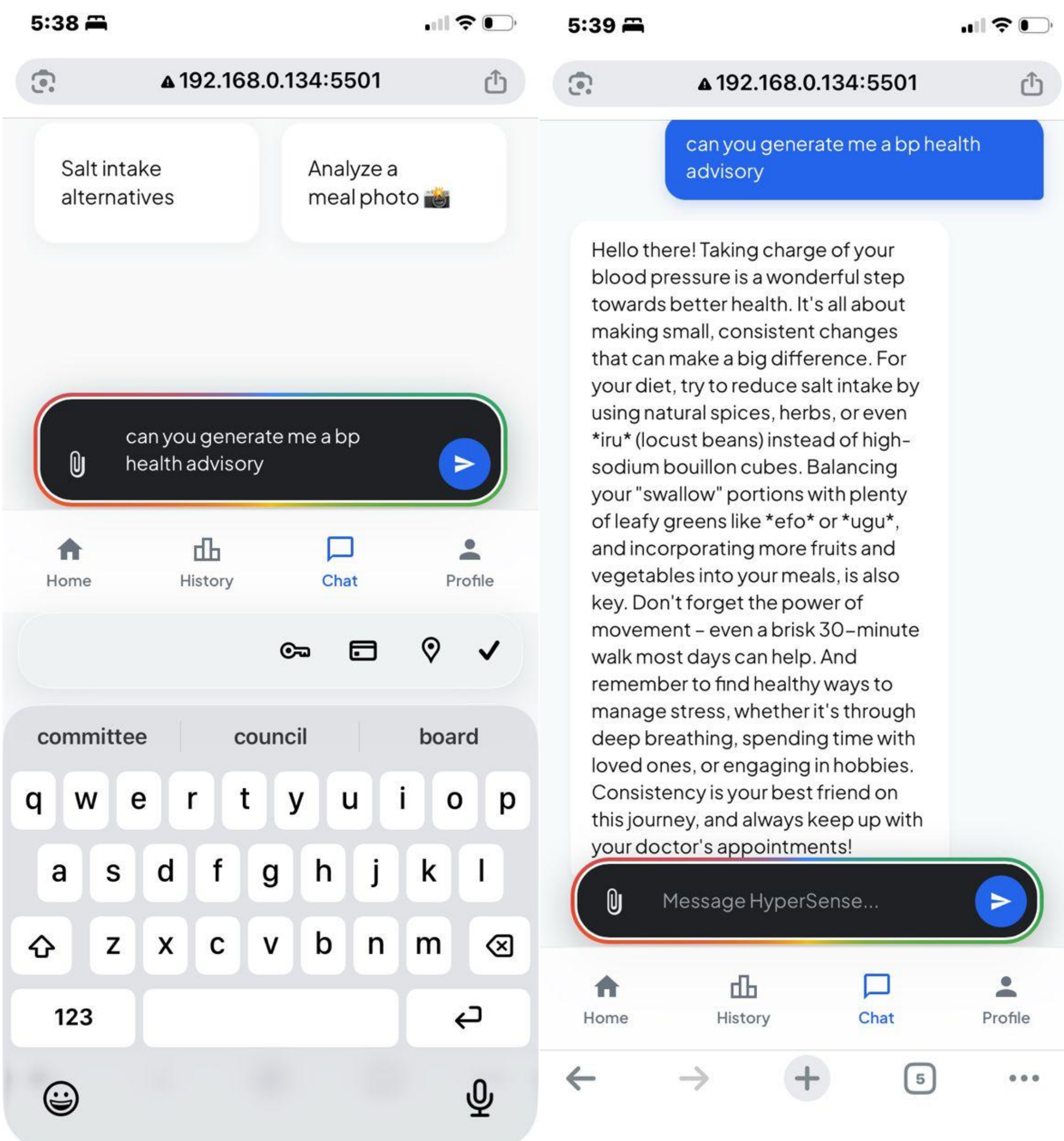

Figure 4.12: AI-Generated BP Health Advisory Output

4.5 Presentation & Analysis of Results

This section summarizes the empirical data gathered from the model evaluation and utilization of the operational results of the web interface. It examines how well the system performs as a whole with respect to achieving its core goals: mathematical accuracy in predicting hypertension risk and cultural relevance of the health advice.

4.5.1 Output and Interface Screens (Samples)

A mock patient record was run through the HyperSense application to show the entire application's function.

Sample Patient Input Elevated Risk Scenario:

The following parameters were given as a test profile by the frontend interface:

1. Physiological – Age: 45, Systolic BP: 138 mmHg, Diastolic BP: 89 mmHg.
2. Lifestyle: High daily stress level (8/10), sedentary exercise level and a diet with high use of bouillon cubes that have high sodium levels.

Result Type: Classified Output and System Output

When submitted, the Flask backend was able to successfully forward the structured data to the XGBoost engine, which was able to give an instant classification as "Prehypertension / Elevated Risk. This was immediately reflected in the following UI changes in the results:

1. **Visual Triage:** The dashboard had a Yellow Risk Badge that enabled the user to identify a higher level of risk without triggering alarmist signals. The Red colour is used to denote Stage 2 Hypertension which provides clarity on risk communication.
2. **Contextual AI Advisory:** The localized health advisory was delivered via the conversational module, which was driven by the Gemini AI. Instead of blanket recommendations from the western world (e.g., "Follow the Mediterranean diet"), the system customized recommendations to patient inputs by recommending the use of less artificial seasoning cube, and to use more local and natural umami seasonings like Locust Beans (Iru) and Ogiri to reduce sodium consumption without reducing the taste of the meal.

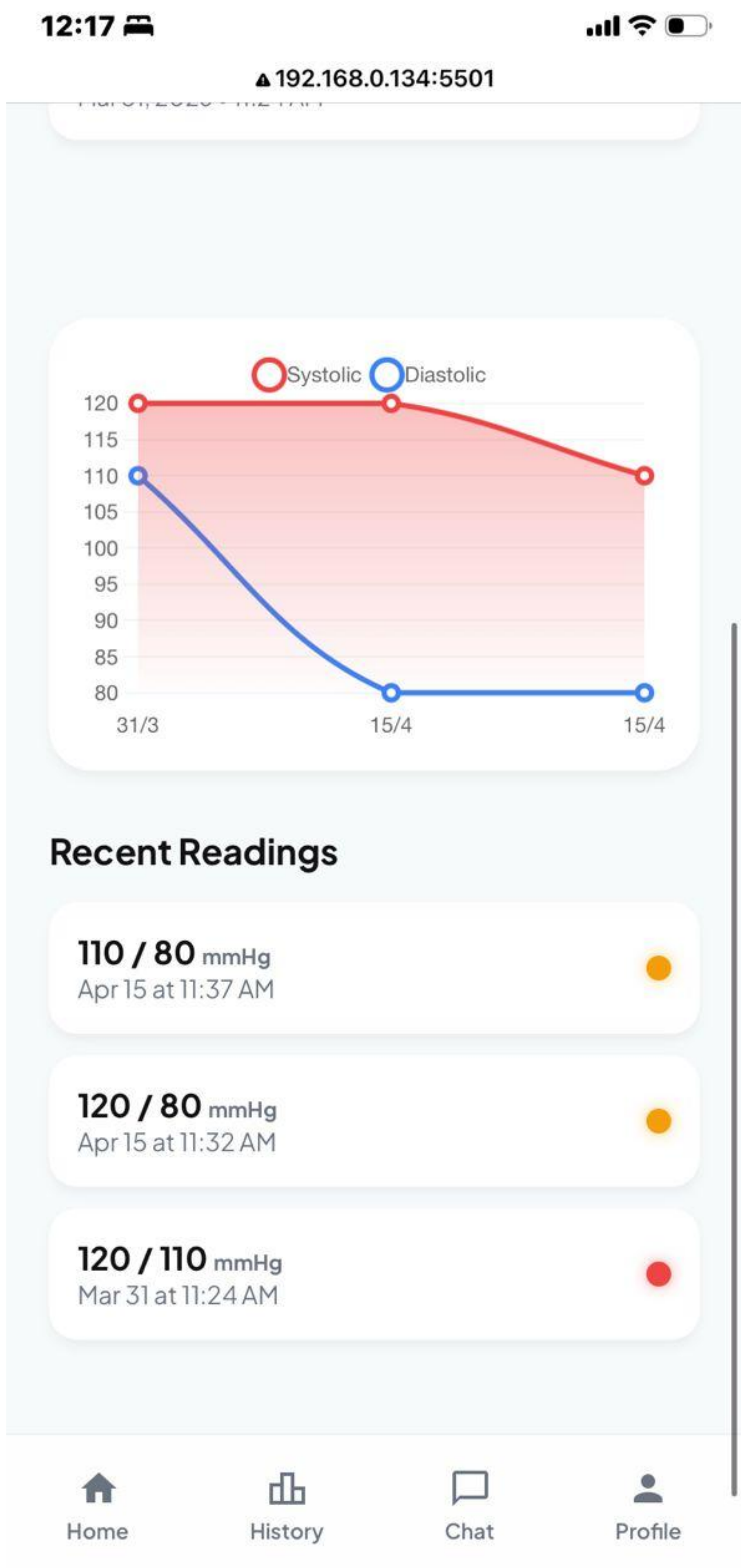

Figure 4.13: Structured Data on the HyperSense Dashboard

4.5.2 Discussion of Results

The implementation and testing process was also successful, showing the potential of machine learning and generative AI to work together as a comprehensive Digital health tool.

Model Performance Analysis:

This remarkable 98.24% overall accuracy was obtained on an unseen validation data using the XGBoost classification engine. More importantly, for a medical application, the model could be able to recall (Sensitivity) 98.35%. This is an important and high recall figure in the context of hypertension prediction, as it would mean the system rarely classifies a patient as "False Negatives" (a patient who is actually at risk but is not identified by the system), thus keeping patient safety paramount. The use of multiple shallow decision trees in the model reduced the likelihood of overfitting, and thus allowed the model to capture complex non-linear relationships between stress, diet and blood pressure.

System Strengths:

Mathematically Sound Predictive Engine: The high accuracy and recall results demonstrate the mathematical reliability of the XGBoost module as a predictive engine.

1. **Cultural Relevance:** The system addresses the "black box" issue of conventional AI by connecting with Google Gemini LLM for numerical predictions. It offers actionable, very much localised advice, directly relevant to the needs of Africans and Nigerians in the field, which greatly enhances the chances of patient adherence.
2. **Decoupled Architecture:** The user interface was in HTML and CSS, while the machine learning backend was in Python, allowing for separation of heavy computing tasks for prediction generation from the mobile user's browser, avoiding any user experience issues related to freezing or lagging.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.0 Introduction

This chapter of this research work gives a detailed summary of this project, the recommendations for future reference, and a conclusion of this research project. This chapter basically summarizes everything that has been detailed in all other chapters of this project.

5.1 Summary of the Study

The research involved four chapters and resulted in the development of the HyperSense application.

In Chapter One, the unmet needs in cardiac monitoring were identified and current mobile health apps were identified as being reactive. The project proposed to develop a proactive, intelligent web-based real-time prediction system for hypertension risk and local health advice.

As discussed in Chapter Two, there is a lack of literature on the subject that ML models are accurate clinically, but consumer apps are not contextually intelligent. It highlighted the need for explaining and culturally relevant risk factors including sodium rich bouillon cubes in diets from West Africa.

Chapter Three introduced localized interventions with Google Gemini LLM coupled with XGBoost and outlined the system design using OOAD and UML, where the system was designed as an N-Tier client/server architecture.

In chapter Four the system was built using Python (Flask), Postgres and a mobile friendly front end. Clinical reliability of testing was obtained: its accuracy was 98.24% and the sensitivity was 98.35%.

Finally, HyperSense managed to transform itself into a proactive diagnostic tool, using predictive modeling and culturally relevant AI suggestions for early management of hypertension.

5.2 Conclusion

The project was able to successfully design and create a proactive web based decision support system for the prediction of hypertension, named HyperSense, as well as the culturally appropriate health messages. In contrast to reactive, mHealth applications, HyperSense creates a scalable, algorithm-driven model for structured medical information with the XGBoost, with an accuracy of 98.24% and a sensitivity of 98.35%, thus reducing the number of false negatives and ensuring that the information is clinically reliable. The system also uses the Google Gemini LLM to address the “black box” issue of AI and offers localized dietary recommendations that are explained and understood. The objectives were achieved, but there are some limitations such as using only one dataset, lack of coverage of comorbidities, constant dependence on the internet, and the lack of clinical trials. Despite this, the study lays a precise technology basis for local management of hypertension.

5.3 Recommendations

The HyperSense project was successful, however, there are suggestions for improvement in the future.

1. **IoT Integration:** Embed APIs into wearables such as smartwatches and Bluetooth sphygmomanometers, and save real-time metrics such as heart rate, sleep, BP, etc. directly to PostgreSQL to eliminate manual data entry.

2. **Model Improvement:** Make the predictive dataset comprehensive and include comorbidities like Type 2 Diabetes, hypercholesterolemia and chronic kidney disease to make it a true cardiovascular support tool.

3. **Localization & Edge AI:** Enable dietary recommendation via quantized LLM locally & add rural-accessible multilingual components (Hausa, Yoruba, Igbo, Swahili).

4. **Clinical Testing & EHR Integration:** Pilot testing and modify the back-end to HL7/HIPAA standards for predictions to synced with hospital records.

REFERENCES

Adeloye, D., Owolabi, E. O., Ojji, D. B., Auta, A., Dewan, M. T., Olanrewaju, T. O., Ogah, O. S., Omoyele, C., Ezeigwe, N., Mpazanje, R. G., Gadanya, M. A., Agogo, E., Alemu, W., Adebiyi, A. O., & Harhay, M. O. (2021). Prevalence, awareness, treatment, and control of hypertension in Nigeria in 1995 and 2020: A systematic analysis of current evidence. *The Journal of Clinical Hypertension*, *23*(5), 963–977

Eshiet, U. I., Orebiyi, E. P., Umoh, I. O., Etukudo, B., Anwana, E. E., Gabriel, U., & Nzedibe, O. E. (2024). Assessment of cardiovascular risks, predisposing factors, and awareness of hypertension risk factors among civil servants in Nigeria. *Medicine India*, *3*(2), 70–76.

Islam, S. M. S., Talukder, A., Awal, M. A., Siddiqui, M. M. U., Ahamad, M. M., Ahammed, B., Rawal, L. B., Alizadehsani, R., Abawajy, J., Laranjo, L., Chow, C. K., & Maddison, R. (2022). Machine Learning Approaches for Predicting Hypertension and Its Associated Factors Using Population-Level Data From Three South Asian Countries. *Frontiers in Cardiovascular Medicine*, *9*, 839379.

Nwevo, C. O., Nkang, J., Nwilegbara, S., Okonkwo, U., Otu, A., & Ameh, S. (2024). Prevalence of Hypertension and Associated Factors among Schoolteachers in Calabar, Nigeria: A Cross-Sectional Study. *Nigerian Journal of Clinical Practice*, *27*(7), 850–858.

Visco, V., Izzo, C., Mancusi, C., Rispoli, A., Tedeschi, M., Virtuoso, N., Giano, A., Gioia, R., Melfi, A., Serio, B., Rusciano, M. R., Di Pietro, P., Bramanti, A., Galasso, G., D'Angelo, G., Carrizzo, A., Vecchione, C., & Ciccarelli, M. (2023). Artificial Intelligence in Hypertension Management: An Ace up Your Sleeve. *Journal of Cardiovascular Development and Disease*, *10*(2), 74.

Ajayi, S. O., et al. (2021). Prevalence of Chronic Kidney Disease as a Marker of Hypertension Target Organ Damage in Africa: A Systematic Review. *International Journal of Hypertension*.

Aljohani, N., & Chandran, D. (2021). The Adoption of Mobile Health Applications by Patients in Developing Countries: A Systematic Review. *International Journal of Advanced Computer Science and Applications*, 12(4).

Bashir, M. A., et al. (2021). Prevalence of Prehypertension in Nigeria: A Systematic Review and Meta-analysis. *International Journal of Epidemiological Health Sciences*, 2, e20.

Batubo, N. P., Moore, J. B., & Zulyniak, M. A. (2024). Dietary factors and hypertension risk in West Africa: a systematic review and meta-analysis. *Nutrition Journal*.

Cai, A., et al. (2021). The Use of Machine Learning for the Care of Hypertension and Heart Failure. *JACC: Asia*, 1(2).

Chiarito, M., et al. (2023). Artificial Intelligence and Cardiovascular Risk Prediction: All That Glitters is not Gold. *European Cardiology Review*, 17.

Chowdhury, M. Z. I., et al. (2022). Prediction of hypertension using traditional regression and machine learning models: A systematic review. *PLoS ONE*, 17(4).

Cucciniello, M., et al. (2021). Development features and study characteristics of mobile health apps in the management of chronic conditions. *npj Digital Medicine*, 4.

Engda, A. A., Salau, A. O., & Ajala, O. (2025). Classical Machine Learning Approaches for Early Hypertension Risk Prediction: A Systematic Review. *Applied AI Letters*.

Hengst, T. M., et al. (2023). The facilitators and barriers of mHealth adoption among people with a low socio-economic position. *Digital Health*, 9.

Jayatilake, S., & Ganegoda, G. (2021). Involvement of Machine Learning Tools in Healthcare Decision Making. *Journal of Healthcare Engineering*.

Liu, T., et al. (2025). Machine learning based prediction models for cardiovascular disease risk using electronic health records. *European Heart Journal - Digital Health*, 6.

Malik, K. S., et al. (2022). Prevalence and Risks Factors of Prehypertension in Africa: A Systematic Review. *Annals of Global Health*, 88(1).

Nuryunarsih, D., et al. (2023). Predicting Changes in Systolic and Diastolic Blood Pressure of Hypertensive Patients in Indonesia Using Machine Learning. *Current Hypertension Reports*, 25.

Nwamekwe, C. O., et al. (2024). Machine Learning-Based Prediction Algorithms for the Mitigation of Maternal and Fetal Mortality in Nigerian Tertiary Hospitals. *International Journal of Engineering Inventions*.

Olowoyo, P., et al. (2025). Prevalence of hypertension in Africa in the last two decades: systematic review and meta-analysis. *Cardiovascular Research*, 121.

Pudjihartono, N., et al. (2022). A Review of Feature Selection Methods for Machine Learning-Based Disease Risk Prediction. *Frontiers in Bioinformatics*, 2.

Schutte, A. E., et al. (2023). Addressing global disparities in blood pressure control: perspectives of the International Society of Hypertension. *Cardiovascular Research*, 119.

Secinaro, S., et al. (2021). The role of artificial intelligence in healthcare: a structured literature review. *BMC Medical Informatics and Decision Making*, 21.

Shahgholian, G., et al. (2025). A Brief Review on Classification of Patients with High Blood Pressure Using Machine Learning Algorithms. *International Journal of Smart Electrical Engineering*, 14.

Uyigue, N., & Mohammadnezhad, M. (2024). Investigating the Determinants and Preventive Approaches of Hypertension Among Adolescents in Nigeria. *Journal of Integrated Health*, 3(2).

APPENDICES

APPENDIX A: SNIPPETS OF CORE SOURCE CODE

A.1: XGBoost Model Training and Serialize (Google Colab)

The following script shows the data preprocessing, hyperparameter tuning and training of a XGBoost ensemble model prior to being exported to the web server.

```
import pandas as pd

import xgboost as xgb

import pickle

from sklearn.model_selection import train_test_split

from sklearn.metrics import accuracy_score


# 1. Load the Dataset

df = pd.read_csv('hypertension_dataset.csv')

df['Has_Hypertension'] = df['Has_Hypertension'].map({'Yes': 1, 'No': 0})


# 2. Preprocessing (One-Hot Encoding for Categorical Variables)

X = pd.get_dummies(df.drop('Has_Hypertension', axis=1), drop_first=True)

y = df['Has_Hypertension']


# 3. Train-Test Split (80/20)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


# 4. Initialize and Train XGBoost Classifier

model = xgb.XGBClassifier(

    n_estimators=100,
```

```
    learning_rate=0.1,

    max_depth=5,

    eval_metric='logloss'

)

model.fit(X_train, y_train)


# 5. Evaluate Model

predictions = model.predict(X_test)

print(f"Validation Accuracy: {accuracy_score(y_test, predictions) *
100:.2f}%")


# 6. Serialize (Save) the Model for Flask Backend

with open('model.pkl', 'wb') as file:

    pickle.dump(model, file)
```

A.2: Integration of XGBoost and Gemini LLM requires the Flask API Controller

This routing function on the backend takes patient information from the frontend and calculates the risk by using the serialized XGBoost model, and retrieves a localized dietary advice from the Google Gemini API.

```
from flask import Flask, request, jsonify

import pickle

import numpy as np

import google.generativeai as genai

import os


app = Flask(__name__)
```

```
# Load Serialized Machine Learning Model
with open('model.pkl', 'rb') as file:
    xgb_model = pickle.load(file)

# Configure Google Gemini API
genai.configure(api_key=os.getenv("GEMINI_API_KEY"))
llm = genai.GenerativeModel('gemini-pro')

@app.route('/api/predict', methods=['POST'])
def predict_risk():
    try:
        # 1. Receive JSON payload from Frontend
        data = request.get_json()

        # 2. Extract and format numerical features for XGBoost
        features = np.array([[
            data['age'], data['salt_intake'], data['stress_score'],
            data['sleep_duration'], data['bmi']
        ]])

        # 3. Compute Hypertension Risk
        prediction = xgb_model.predict(features)[0]
        risk_level = "High Risk / Hypertensive" if prediction == 1 else
"Normal / Low Risk"
```

```python
        # 4. Generate Localized Health Advisory via Gemini LLM
        prompt = f"""
        A patient in West Africa has a {risk_level} of hypertension.
        Their stress score is {data['stress_score']}/10 and they consume
heavy amounts of bouillon cubes.
        Provide a short, culturally relevant dietary and lifestyle advisory
in 3 sentences.
        """
        advisory_response = llm.generate_content(prompt)

        # 5. Return aggregated JSON back to Frontend Dashboard
        return jsonify({
            'status': 'success',
            'computed_risk': risk_level,
            'ai_advisory': advisory_response.text
        }), 200

    except Exception as e:
        return jsonify({'error': str(e)}), 500
```

APPENDIX B: SAMPLE DATASET EXTRACTION

This table format shows a sample of the multifactorial, structured medical data (hypertension_dataset.csv) used to train the predictive engine.

Age	Salt Intake	Stress Score	BP History	Sleep Duration	BMI	Medication	Family History	Exercise Level	Has Hypertension
69	8.0	9	Normal	6.4	25.8	None	Yes	Low	Yes
32	11.7	10	Normal	5.4	23.4	None	No	Low	No
78	9.5	3	Normal	7.1	18.7	None	No	Moderate	No
38	10.0	10	Hypertension	4.2	22.1	ACE Inhibitor	No	Low	Yes
41	9.8	1	Prehypertension	5.8	16.2	Other	No	Moderate	No
20	10.8	3	Hypertension	5.2	21.9	Beta Blocker	Yes	High	Yes
39	8.9	0	Normal	7.8	27.6	Beta Blocker	Yes	High	No
70	5.9	1	Hypertension	7.2	25.8	None	No	Moderate	Yes
19	9.3	7	Normal	4.7	36.5	Beta Blocker	Yes	Low	Yes
47	7.2	5	Normal	6.2	24.3	None	No	High	No

APPENDIX C: SYSTEM INSTALLATION AND USER GUIDE

A guide for administrators and future researchers to install the HyperSense application locally.

C.1 Prerequisites

- o Installed Python version >= 3.9.
- o This is a locally installed and running PostgreSQL.
- o A Google Gemini API Key.

C.2 Setup Instructions

1. **Clone the Repository:** Copy the Project files to a local folder and open the folder with the terminal.

```
cd hypersense-project
```

2. **Use Virtual Environment:** Separate the project dependencies so as not to have conflicts with other software programs.

```
python -m venv venv
```

`source venv/bin/activate` (On Windows use: `venv\Scripts\activate`)

3. **Install Dependencies:** Install the following packages: Flask, XGBoost, SQLAlchemy and Generative AI SDK from the requirements file.

```
pip install –r requirements.txt
```

4. **Set up Environment Variables:** Make sure that there is a `.env` file in the root folder and safely enter the database URI and AI keys:

```
DATABASE_URL=postgresql://postgres:password@localhost:5432/hypers
ense_db

GEMINI_API_KEY= AIzaSyDq4Al9nWv5zlabh64LdPLJr-tFN37VCus
```

5. **Start the Database:** Run the Flask migration script to establish the relational tables (Users, Health Records).

```
flask db upgrade
```

6. **Start the local backend server:** Launch the Application.

```
flask run
```

7. **Access the Dashboard:** You can open a modern browser and go to http://127.0.0.1:5000 to communicate with the front end dashboard.