Tribhuvan University

Orchid International College

A Final Year Project Report

On

CHRONIC KIDNEY DISEASE PREDICTION USING NAIVE BAYES

Under the Supervision of

Er. Diwakar Upadhyaya

Lecturer

Orchid International College

Submitted To:

Department of computer science and information technology

Orchid international college

In partial fulfillment of the requirement for the Bachelor Degree in Computer Science and Information Technology

Submitted By:

Binaya Khadka (20799/075)

Krijan Chakradhar (20805/075)

Niraj Uprety (20814/075)

April, 2023

ORCHID INTERNATIONAL College
[TRIBHUVAN UNIVERSITY AFFILIATE]

SUPERVISOR’S RECOMMENDATION

I hereby recommend that the report prepared under my supervision by Binaya Khadka (TU Exam Roll No. 20799/075), Krijan Chakradhar (TU Exam Roll No. 20805/075), Niraj Uprety (TU Exam Roll No. 20814/075) entitled “**CHRONIC KIDNEY DISEASE PREDICTION USING NAIVE BAYES**” in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Information Technology be processed for evaluation.

…………………..…….

Er. Diwakar Upadhyaya

Lecturer, Department of CSIT
Orchid International College
Bijayachowk, Gaushala

ORCHID INTERNATIONAL College
[TRIBHUVAN UNIVERSITY AFFILIATE]

LETTER OF APPROVAL

This is to certify that this project prepared by Binaya Khadka, Krijan Chakradhar and Niraj Uprety entitled "Chronic Kidney Disease Prediction Using Naive Bayes" in partial fulfilment of the requirements for the degree of B. Sc. in Computer Science and Information Technology has been well studied. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

----------------------------- **Er. Diwakar Upadhyaya** Lecturer, Orchid International College Bijayachowk, Gaushala	----------------------------- **Er. Dhiraj Kumar Jha** Head Of Department, Orchid International College Bijayachowk, Gaushala
----------------------------- **Shikha Sharma** Internal Examiner, Full-Time Faculty, Orchid International College Bijayachowk, Gaushala	----------------------------- **External Examiner** Central Department of Computer Science and IT, Tribhuvan University, Kirtipur, Nepal

ACKNOWLEDGEMENT

"We would like to express our sincerest gratitude to our esteemed teacher, **Er. Dhiraj Kumar Jha**, for his unwavering support, guidance, and encouragement throughout this project. We would also like to extend our thanks to our supervisor, **Er. Diwakar Upadhyaya**, for his insightful feedback and valuable contributions to our work.

We are grateful for the opportunities and resources provided by **Orchid International College** for our academic growth and development. We would also like to extend our appreciation to the administration, staff, and fellow students for their support and collaboration.

This report would not have been possible without the contributions of everyone mentioned above. We are forever grateful and honored to have had the opportunity to work with such exceptional individuals. Thank you."

Binaya Khadka (20799/075)
Krijan Chakradhar (20805/075)
Niraj Uprety (20814/075)

ABSTRACT

The aim of this project is to improve the quality of life by early detection from chronic kidney diseases through web-based application. This application will use machine learning techniques, such as classification, to diagnose patients based on their data.

The system focuses on using Naive Bayes algorithms to predict the likelihood of a patient having kidney disease. This project utilizes data mining techniques to extract hidden information from massive datasets to make effective diagnoses and decisions. The results suggest that the combination of hemoglobin, albumin, specific gravity, hypertension, diabetes, and serum creatinine are the most important attributes in the early detection of chronic kidney disease.

Keywords:

Chronic kidney disease, Quality of life, Web-based application, Machine learning, Classification, Naive Bayes algorithms, Prediction, Data mining, Early detection

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
BSc CSIT	Bachelors in Science Computer Science and Information Technology
CKD	Chronic Kidney Disease
CSP	Common Spatial Pattern
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
IDE	Integrated Development Environment
JS	JavaScript
KNN	K-Nearest Neighbor
LDA	Linear Discriminant Analysis
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface

CHAPTER 1: INTRODUCTION

1.1 Introduction

The Chronic Kidney Disease Prediction system is a web-based system that helps patient keep track of their kidney health. It works by collecting information about the patient's health and using data mining algorithms to analyze the information and make predictions about the patient's condition. This system helps patient to make faster and more accurate diagnoses, and can be used from a distance. The goal is to make it easier for patient to keep track of their health and help them understand their medical status.

1.2 Problem Statement

- People with chronic diseases have a higher risk of health problems and unequal access to healthcare.
- Lack of technology to monitor and track health care for these diseases makes it difficult and time-consuming.
- Health measurements and records are often recorded on paper, which can lead to missing information.
- Due to lack of early diagnosis people are facing major consequences.

1.3 Objectives

- To focus on chronic kidney diseases by using Naive Bayes algorithms.
- To provide experimental results and analysis to measure accuracy of the system

1.4 Scope and Limitation

Scope:

- Developing a web-based application for chronic kidney disease diagnosis using Naive Bayes algorithm
- Using machine learning and data mining techniques for effective diagnosis and predictions
- Identifying the most important attributes in the early detection of chronic kidney disease

Limitations:

- The accuracy of the predictions may be limited by the quality and quantity of the data used
- The algorithm may not take into account all relevant factors that can impact the disease
- The application may not be accessible or usable for patients with limited internet access or technical skills.
- The application is based on statistical data and may not reflect the individual characteristics and conditions of each patient.

1.5 Development Methodology

Scrum is an agile development methodology that is widely used in software development projects. In this CKD application, we adopted the Scrum framework for the development process. This methodology involves iterative and incremental development, with frequent review and feedback sessions.

We used Scrum to manage development process by organizing work into sprints. Each sprint was typically two weeks long, and at the beginning of each sprint, we planned and committed to a set of deliverables. We held daily stand-up meetings to track progress and identify any issues that may arise.

The Scrum methodology helped us to stay focused on the development goals and ensured that we were continuously delivering value to the end-users. We were able to respond to changes in requirements and feedback from users quickly and efficiently, which helped to ensure that the application met the needs of its users. Overall, the Scrum methodology was an effective approach for managing the development of this CKD application.

1.6 Report Organization

The report organization for the chronic kidney disease prediction using data mining algorithm are listed below:

- Requirements Gathering: Understanding the needs and requirements of the system and stakeholders such as patients, doctors, and hospitals.

- Data Collection: Acquiring and preparing relevant medical data from various sources such as electronic medical records, lab reports, and hospital databases.
- Data Preprocessing: Cleaning and processing the collected data to remove any inconsistencies or errors.
- Model Selection: Selecting the appropriate machine learning algorithm such as Naive Bayes to diagnose patients based on their data.
- Model Training: Training the model using the preprocessed data and evaluating its performance.
- Model Deployment: Integrating the trained model into the web-based application.
- User Testing: Testing the system with real-world data to ensure it is accurate, user-friendly, and provides valuable insights.
- Deployment and Maintenance: Deploying the system in a live environment and providing ongoing maintenance and support.

CHAPTER 2: LITERATURE REVIEW

This study uses data mining classifiers to forecast the development of chronic kidney disease. Chronic diseases are on the rise today and have a major impact part in a person's life. Data mining technology is employed to extract the hidden information regarding chronic disease from a given dataset. Big data is another area of study that is utilized for the processing and storing of large amounts of structured, unstructured, and semi-structured data. Two data mining classifiers are employed in this study to forecast the development of chronic kidney disease. SVM with KNN (K-Nearest Neighbor) (Support Vector Machine). [6]

Making precise treatment decisions for patients using machine learning-based predictive analytics is a difficult task for doctors. Using clinical data, we offer machine learning algorithms in this study for predicting chronic renal disease. K-nearest neighbors (KNN), support vector machines (SVM), logistic regression (LR), and decision tree classifiers are four machine learning techniques that are investigated. The performance of these models is compared in order to choose the best classifier for predicting chronic kidney disease. These predictive models are built from a chronic kidney disease dataset. [5]

People with Chronic Kidney Disease (CKD) are unaware that occasionally, medical tests they undergo for other reasons can reveal important details about the disease. Sometimes, the identification of the disease is not handled with this information in an efficient manner.

Therefore, characteristics of various medical tests are examined to determine whether characteristics have information concerning CKD. Several features of healthy participants and subjects with CKD are examined in a database using various methodologies. The prominent characteristics that may aid in the detection of CKD are first determined using the Common Spatial Pattern (CSP) filter and Linear Discriminant Analysis (LDA). Here, CSP filter is used to separate CKD and non-CKD samples as efficiently as possible. The prominent attributes are then identified using classification techniques. According to these analyses, blood creatinine along with hemoglobin, albumin, specific gravity, hypertension, and diabetes mellitus are the most crucial factors in the early identification of CKD. Additionally, it implies that blood pressure and blood glucose properties could be employed in the lack of knowledge of hypertension and diabetes mellitus. [4]

Obtaining diagnostic outcomes using data mining is currently popular. The healthcare sector gathers a sizable volume of unmined data in an effort to uncover hidden information for accurate diagnosis and decision-making. Data mining is the process of identifying true and specific patterns in data and extracting hidden information from large datasets. Data mining methods include clustering, classification, association analysis, regression, and others. The study's goal is to predict chronic kidney disease (CKD) utilizing artificial neural networks and naive bayes classification techniques (ANN). The experimental findings used in the RapidMiner program demonstrate that Naive Bayes produces results that are more accurate than those of an Artificial Neural Network.[3]

The study analyzed the use of AI algorithms in the diagnosis of chronic kidney disease (CKD) in South Africa. The extreme gradient boosting (XGBoost) was chosen as the base model due to its high performance. The full model trained on all features achieved a testing accuracy, sensitivity, and specificity of 1.000. The reduced model using fewer features was developed to minimize the cost and time of diagnosis while maintaining high performance. The set-theory based rule combining different feature selection methods was used, which improved the performance of the reduced model and achieved accuracy, sensitivity, and specificity of 1.000.[2]

The paper focuses on the use of machine learning techniques in the prediction of Chronic Kidney Disease (CKD), also known as chronic nephritic sickness. CKD can cause various complications such as increased blood levels, anemia, weak bones, and nerve injury. Early detection and treatment can prevent the disease from getting worse. The paper analyzes the performance of Naive Bayes, K-Nearest Neighbor (KNN) and Random Forest classifier in terms of accuracy, precision, and execution time for CKD prediction. The results show that the Random Forest classifier performs better than Naive Bayes and KNN.[1]

CHAPTER 3: SYSTEM ANALYSIS

3.1 System Analysis

3.1.1 Requirement Analysis

The requirement analysis for a chronic kidney disease (CKD) prediction project would involve identifying and defining the specific goals, constraints, and requirements for the project. This process can involve the following steps:

- Understanding the problem: Identifying the specific goals of the project, such as predicting the risk of CKD in a patient, and defining the scope of the project.
- Stakeholder analysis: Identifying and engaging stakeholders, such as healthcare providers and patients, to gather their requirements and ensure that the final product meets their needs.
- Data collection: Identifying and collecting relevant data, such as patient medical records, demographic information, and lab results, to be used in the prediction model.
- Data analysis: Analyzing the collected data to identify patterns and relationships that can be used to predict the risk of CKD.
- Algorithm selection: Selecting the appropriate data mining algorithms for the prediction model based on the results of the data analysis.
- Model validation: Validating the accuracy of the prediction model using various methods, such as cross-validation and testing on a separate dataset.
- User interface design: Designing the user interface for the prediction model to ensure that it is intuitive and user-friendly.

3.1.2 Functional Requirements

The functional requirements for a chronic kidney disease (CKD) prediction project are the specific features and capabilities that the final product must have in order to meet the goals and needs of the stakeholders.

- Data input: The ability for users to input patient data, such as medical history, demographic information, and lab results, into the system.

- Prediction model: The implementation of a data mining algorithm to predict the risk of CKD based on the input data.
- Data storage: The ability to securely store patient data and prediction results for future use and analysis.
- User interface: An intuitive and user-friendly interface for inputting and displaying data and prediction results.
- Accuracy: A high degree of accuracy in the prediction model, validated through testing on a separate dataset.
- Privacy and security: The implementation of appropriate security measures to protect patient data and ensure privacy.

3.1.3 Non-functional Requirements

Non-functional requirements for a chronic kidney disease (CKD) prediction project are the underlying constraints and characteristics that must be met in order for the product to be successful.

Performance: The ability of the system to respond quickly and efficiently to user requests, even when dealing with large amounts of data.

Scalability: The ability of the system to handle increased demand as the number of users and patients grows over time.

Reliability: The ability of the system to operate effectively and consistently, with minimal downtime or errors.

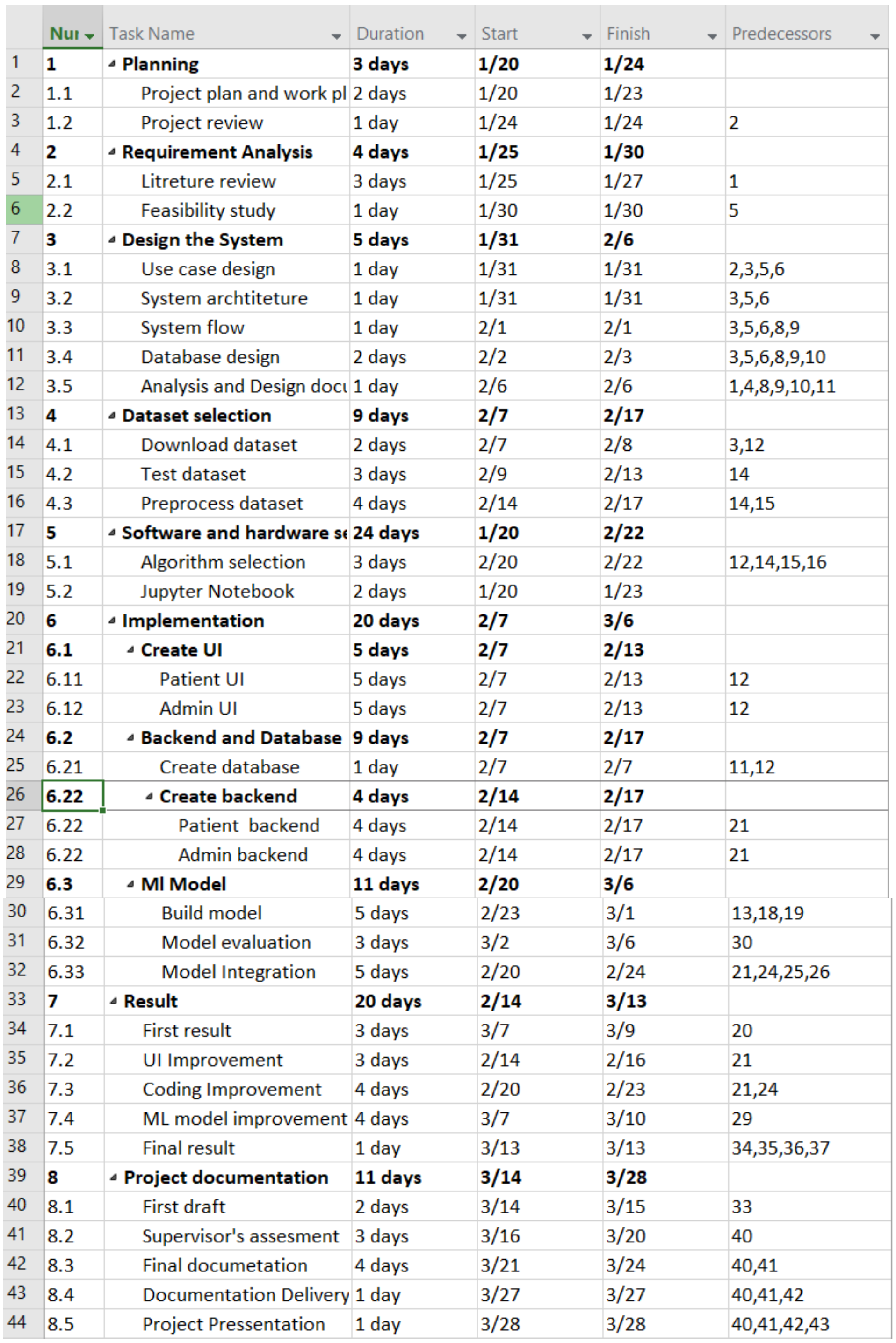
Usability: The ease with which users can understand and use the system, with minimal training required.

Security: The implementation of appropriate security measures to protect patient data and ensure privacy.

3.1.4 Feasibility Analysis

A feasibility study for a chronic kidney disease (CKD) prediction project is an assessment of the potential viability and success of the project, based on various factors such as technical, economic, and operational considerations. A feasibility study for a CKD prediction project may include the following steps:

- Technical feasibility: Assessing the availability and suitability of technology, such as data mining algorithms, to implement the prediction model.
- Economic feasibility: Assessing the costs and benefits of the project, including the cost of developing and implementing the prediction model and the potential return on investment.
- Operational feasibility: Assessing the resources, including personnel and infrastructure, needed to successfully complete the project and implement the prediction model.
- Data feasibility: Assessing the availability and quality of the data needed to build the prediction model, including patient medical records, demographic information, and lab results.
- Stakeholder feasibility: Assessing the willingness and ability of stakeholders, such as healthcare providers and patients, to adopt and use the prediction model.
- Schedule Feasibility: Schedule feasibility is an important component of the feasibility study for a chronic kidney disease (CKD) prediction project. Schedule feasibility involves assessing the amount of time required to complete the project and determine whether it can be completed within the desired timeframe.



	Nu	Task Name	Duration	Start	Finish	Predecessors
1	**1**	**Planning**	**3 days**	**1/20**	**1/24**	
2	1.1	Project plan and work pl	2 days	1/20	1/23	
3	1.2	Project review	1 day	1/24	1/24	2
4	**2**	**Requirement Analysis**	**4 days**	**1/25**	**1/30**	
5	2.1	Litreture review	3 days	1/25	1/27	1
6	2.2	Feasibility study	1 day	1/30	1/30	5
7	**3**	**Design the System**	**5 days**	**1/31**	**2/6**	
8	3.1	Use case design	1 day	1/31	1/31	2,3,5,6
9	3.2	System archtiteture	1 day	1/31	1/31	3,5,6
10	3.3	System flow	1 day	2/1	2/1	3,5,6,8,9
11	3.4	Database design	2 days	2/2	2/3	3,5,6,8,9,10
12	3.5	Analysis and Design doc	1 day	2/6	2/6	1,4,8,9,10,11
13	**4**	**Dataset selection**	**9 days**	**2/7**	**2/17**	
14	4.1	Download dataset	2 days	2/7	2/8	3,12
15	4.2	Test dataset	3 days	2/9	2/13	14
16	4.3	Preprocess dataset	4 days	2/14	2/17	14,15
17	**5**	**Software and hardware s**	**24 days**	**1/20**	**2/22**	
18	5.1	Algorithm selection	3 days	2/20	2/22	12,14,15,16
19	5.2	Jupyter Notebook	2 days	1/20	1/23	
20	**6**	**Implementation**	**20 days**	**2/7**	**3/6**	
21	**6.1**	**Create UI**	**5 days**	**2/7**	**2/13**	
22	6.11	Patient UI	5 days	2/7	2/13	12
23	6.12	Admin UI	5 days	2/7	2/13	12
24	**6.2**	**Backend and Database**	**9 days**	**2/7**	**2/17**	
25	6.21	Create database	1 day	2/7	2/7	11,12
26	**6.22**	**Create backend**	**4 days**	**2/14**	**2/17**	
27	6.22	Patient backend	4 days	2/14	2/17	21
28	6.22	Admin backend	4 days	2/14	2/17	21
29	**6.3**	**Ml Model**	**11 days**	**2/20**	**3/6**	
30	6.31	Build model	5 days	2/23	3/1	13,18,19
31	6.32	Model evaluation	3 days	3/2	3/6	30
32	6.33	Model Integration	5 days	2/20	2/24	21,24,25,26
33	**7**	**Result**	**20 days**	**2/14**	**3/13**	
34	7.1	First result	3 days	3/7	3/9	20
35	7.2	UI Improvement	3 days	2/14	2/16	21
36	7.3	Coding Improvement	4 days	2/20	2/23	21,24
37	7.4	ML model improvement	4 days	3/7	3/10	29
38	7.5	Final result	1 day	3/13	3/13	34,35,36,37
39	**8**	**Project documentation**	**11 days**	**3/14**	**3/28**	
40	8.1	First draft	2 days	3/14	3/15	33
41	8.2	Supervisor's assesment	3 days	3/16	3/20	40
42	8.3	Final documetation	4 days	3/21	3/24	40,41
43	8.4	Documentation Delivery	1 day	3/27	3/27	40,41,42
44	8.5	Project Presentation	1 day	3/28	3/28	40,41,42,43

Figure 3.1: Gantt Chart

3.1.5 Analysis

3.1.5.1 Use Case Diagram

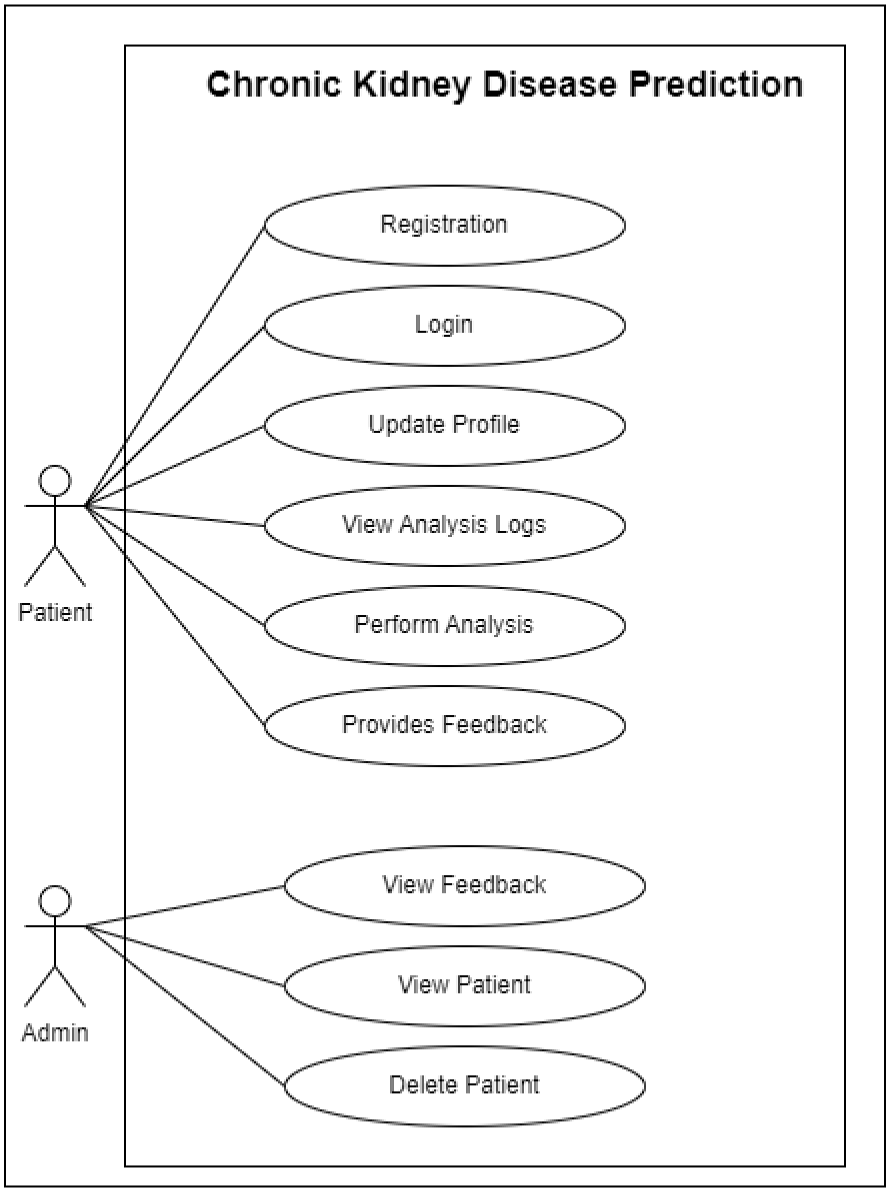


Figure 3.2: Use Case Diagram

Table 3.1: UC1 Login

Use case Identifier	**UC1 - Login**
Primary Actor	User
Secondary Actor	None
Description	User attempts to access the system by logging in
Pre-condition	Correct username and password required for login.
Success Scenario	Login Successfully
Failure Scenario	Login Failed

Table 3.2: UC2 Registration

Use case Identifier	**UC2 - Registration**
Primary Actor	User
Secondary Actor	None
Description	User attempts to get registration
Pre-condition	User provides name, email address and password. Password and Confirm password should match.
Success Scenario	User gets registered in to the system
Failure Scenario	Registration Failed

Table 3.3: UC3 Update Profile

Use case Identifier	**UC3 – Update Profile**
Primary Actor	User
Secondary Actor	None
Description	User modify the existed data.
Pre-condition	Name, email address and password changed
Success Scenario	User details get updated in the database
Failure Scenario	Update Failed

Table 3.4: UC4 View Analysis Log

Use case Identifier	**UC4 – View Analysis Log**
Primary Actor	User
Secondary Actor	None
Description	User try to access the analysis logs that are performed.
Pre-condition	At least one analysis should be performed
Success Scenario	User gets analysis logs
Failure Scenario	User does not gets analysis logs

Table 3.5: UC5 Perform Analysis

Use case Identifier	**UC5 – Perform Analysis**
Primary Actor	User
Secondary Actor	None
Description	User inputs value in the form and perform the analysis
Pre-condition	All the filed should be filled
Success Scenario	User gets precise result
Failure Scenario	User gets result based on average scenario

Table 3.6: UC6 Provide Feedback

Use case Identifier	**UC6 – Provides Feedback**
Primary Actor	User
Secondary Actor	None
Description	User provides feedback with message
Pre-condition	All the filed should be filled
Success Scenario	Message sent successfully
Failure Scenario	Some Filed are missing

Table 3.7: UC7 View Feedback

Use case Identifier	**UC7 –View Feedback**
Primary Actor	Admin
Secondary Actor	None
Description	Admin will view the message send by the user
Pre-condition	Contain at least one message
Success Scenario	Admin gets the message
Failure Scenario	Admin does not get message

Table 3.8: UC8 View Patient

Use case Identifier	**UC8 –View Patient**
Primary Actor	Admin
Secondary Actor	None
Description	Admin view the patient details like reports and patient information
Pre-condition	Patient must have performed analysis
Success Scenario	Views patient data and report
Failure Scenario	Views only patient data but not report

Table 3.9: UC9 Delete Patient

Use case Identifier	**UC1 –Delete Patient**
Primary Actor	Admin
Secondary Actor	None
Description	Admin delete the patient account
Pre-condition	Patient have never performed any analysis
Success Scenario	Patient deleted successfully
Failure Scenario	Patient cannot be deleted

3.1.5.2 ER Diagram

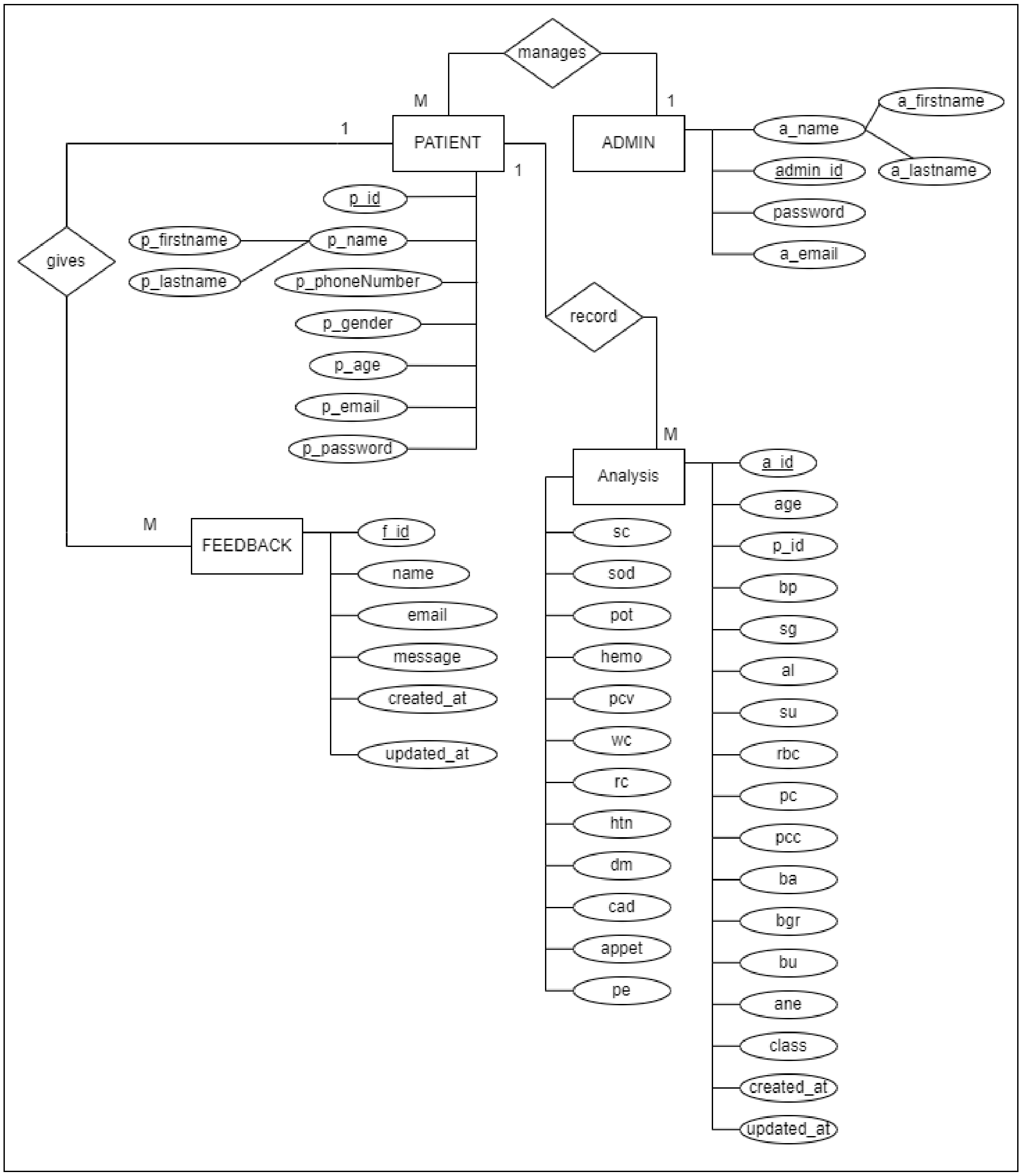


Figure 3.3: ER Diagram

An Entity-Relationship (ER) diagram is a graphical representation of entities and their relationships to each other in a database system. In the context of a Chronic Kidney Disease Prediction system, the ER diagram would contain entities for "Patient," "Admin," and "Analysis."

The "Patient" entity would contain information about each patient, such as name, age, gender. The "Admin" entity would contain information about the person responsible for managing the system, such as the ability to edit or delete patient information. The "Analysis" entity would contain information about each analysis performed on a patient, such as the date and time, result, and model inputs.

The ER diagram would show the relationships between these entities, such as a Patient having one or many Analyses, and an Admin being responsible for many Patients. The diagram would also show the attributes of each entity, such as the Patient's name, the admin’s name, and the Analysis's result.

3.1.6 Dataset Analysis

Given 24 health related attributes taken in 2-month period of 400 patients, using the information of the 158 patients with complete records to predict the outcome (i.e. whether one has chronic kidney disease) of the remaining 242 patients (with missing values in their records).

We use the following representation to collect the dataset

age - age

bp - blood pressure

sg - specific gravity

al - albumin

su - sugar

rbc - red blood cells

pc - pus cell

pcc - pus cell clumps

ba - bacteria

bgr - blood glucose random

bu - blood urea

sc - serum creatinine

sod - sodium

pot - potassium

hemo - hemoglobin

pcv - packed cell volume

wc - white blood cell count

rc - red blood cell count

htn - hypertension

dm - diabetes mellitus

cad - coronary artery disease

appet - appetite

pe - pedal edema

ane - anemia

class - class

CHAPTER 4: SYSTEM DESIGN

4.1 Design

The Chronic Kidney Disease Prediction using Data Mining Algorithm includes the system architecture, system flowchart, ML model flowchart, class diagram, and UI design. The system architecture provides a high-level overview of the components and their interactions in the system. The system flowchart outlines the steps involved in the flow of information and tasks within the system. The ML model flowchart illustrates the steps involved in creating the prediction model. The class diagram defines the classes, attributes, and methods involved in the implementation of the system. The UI design specifies the visual components and the user experience of the system.

4.1.1 System Architecture

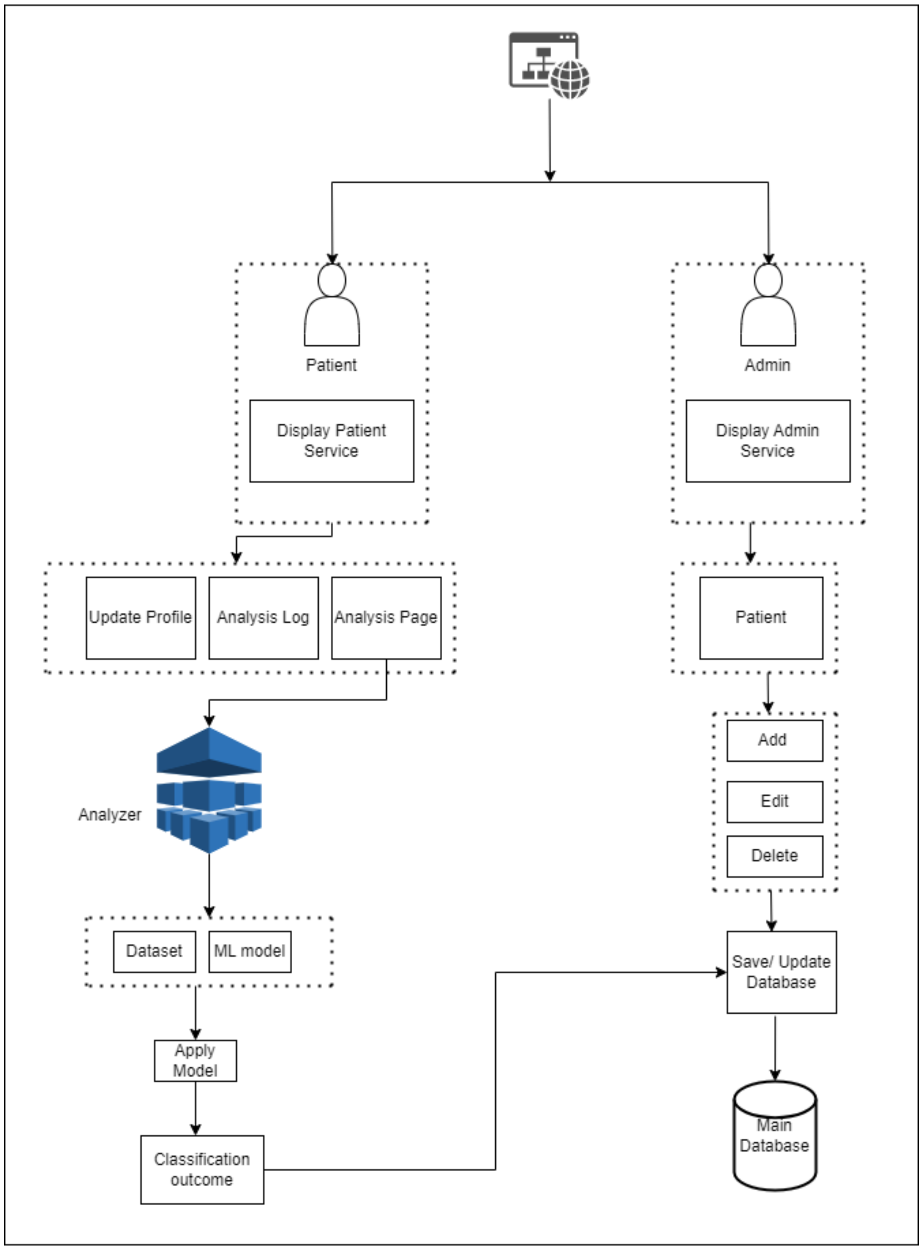


Figure 4.1: System Architecture

4.1.2 System Flowchart

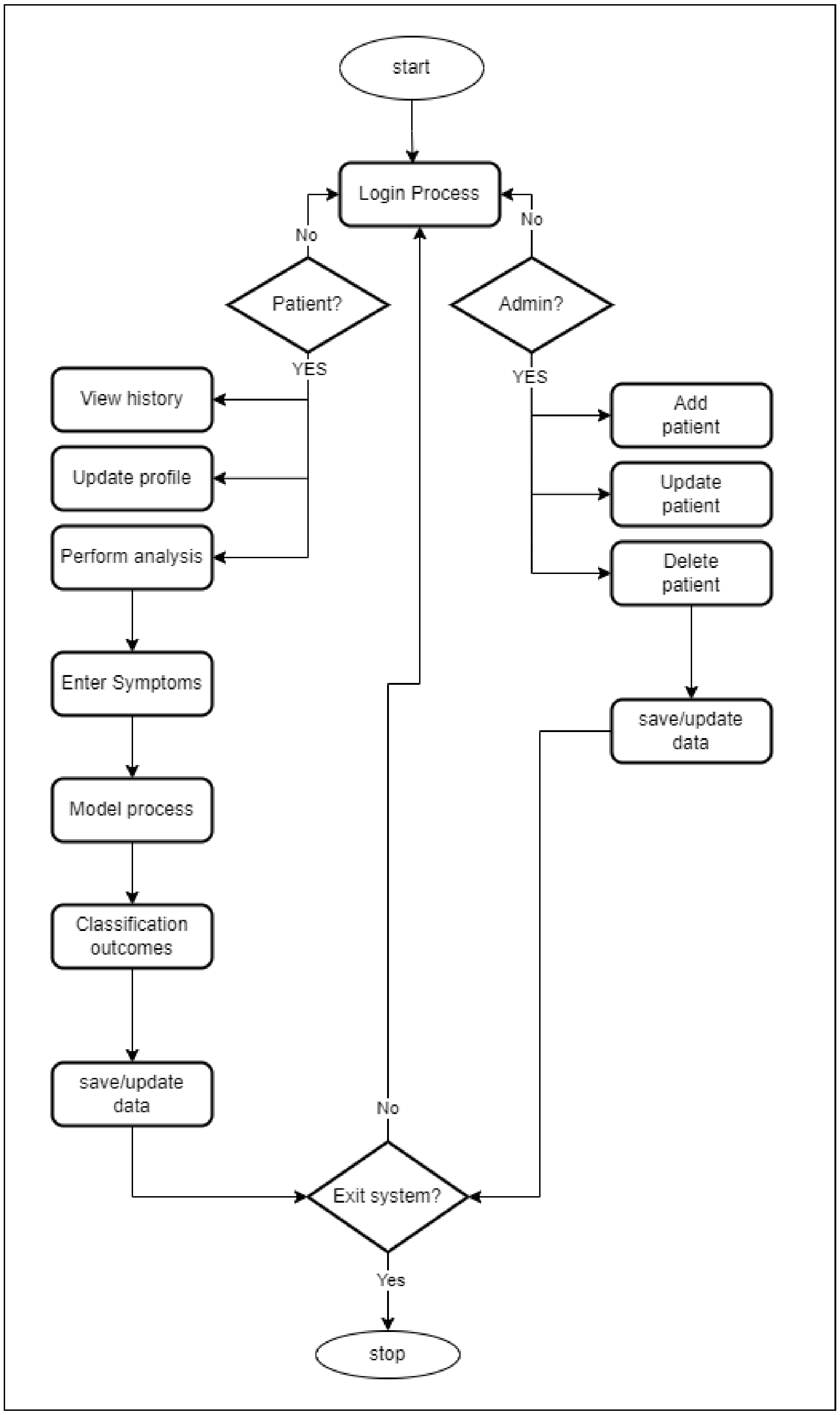


Figure 4.2: System Flow Diagram

4.1.3 ML Model Flowchart

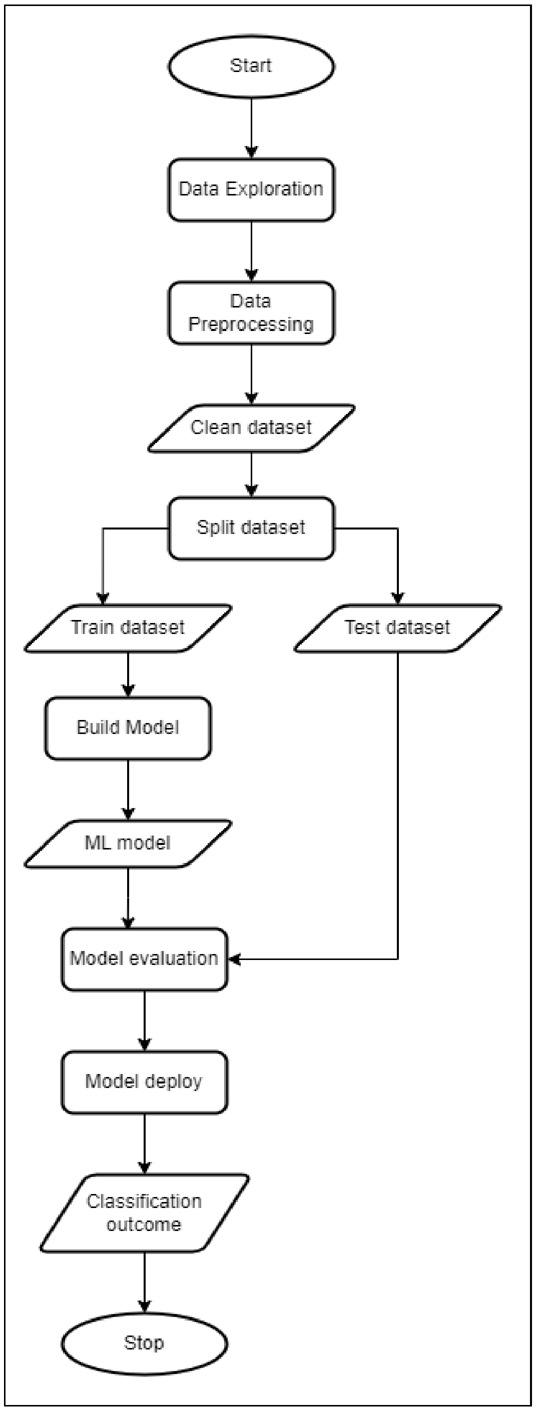


Figure 4.3: ML Model

4.1.4 Class Diagram

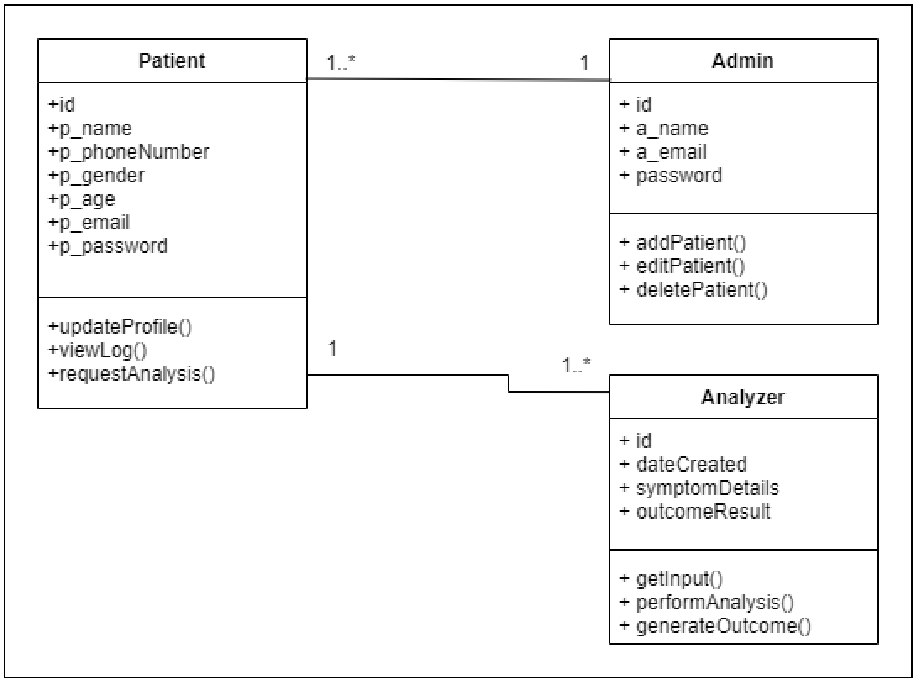


Figure 4.4: Class Diagram for CKD

The class diagram for the CKD application includes three main classes: Patient, Admin, and Analyzer. The Admin class has a one-to-many relationship with the Patient class, which means that one admin can manage many patients. The Patient class also has a one-to-many relationship with the Analyzer class, which indicates that one patient can have multiple analysis results.

4.1.5 Activity Diagram

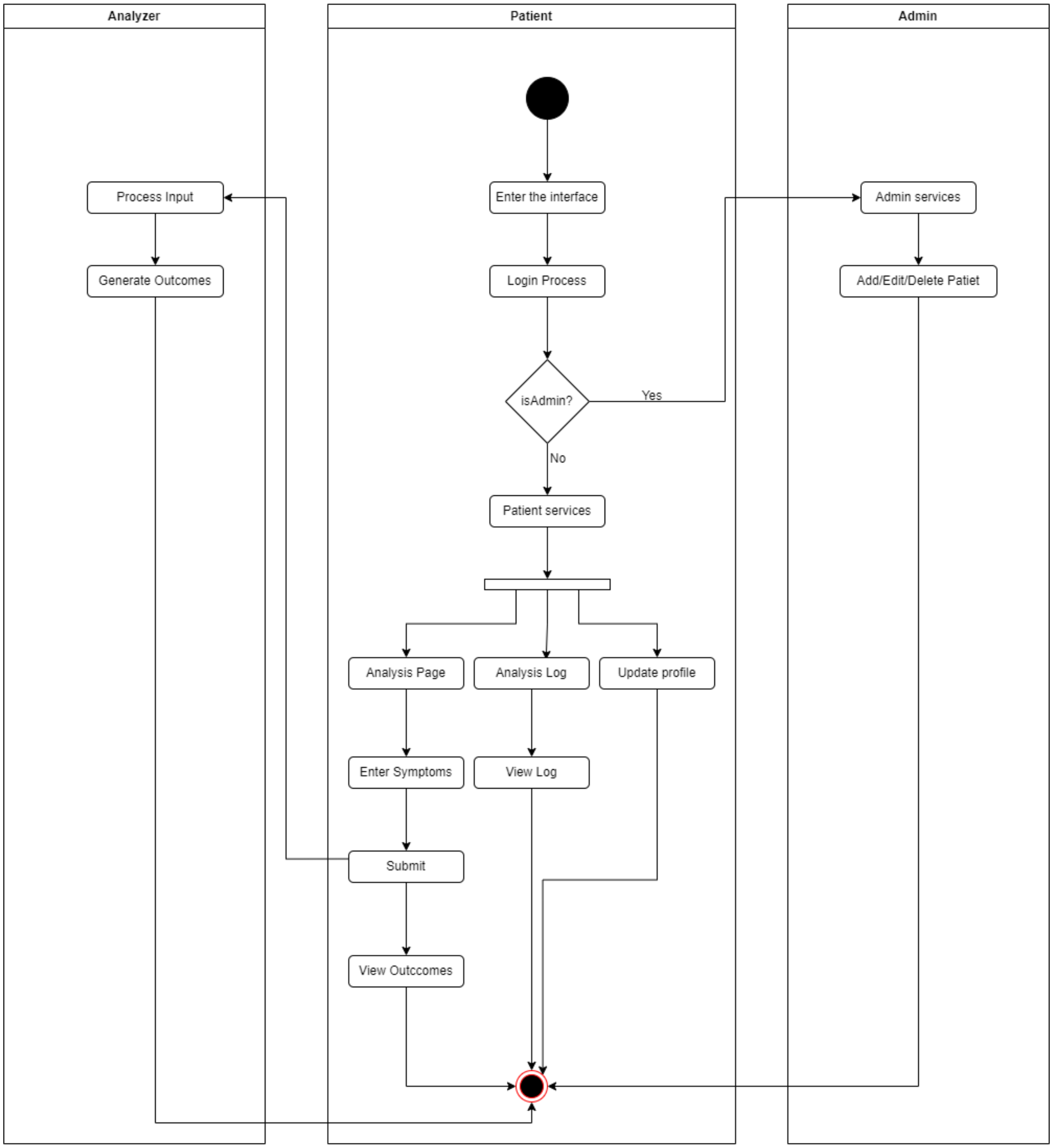


Figure 4.5: Activity Diagram

4.1.6 Sequence Diagram

The sequence diagram for the Chronic Kidney Disease Prediction using Data Mining Algorithm involves three entities: Patient, System, and Admin. The Patient requests login to the System and receives a response, while the admin also logs in and can add, delete, or edit Patient information. The System works on a CKD severity dataset and performs preprocessing, training, and model building. The Patient inputs symptoms, and the System generates a result and sends it back to the Patient. Additionally, the Patient can request an analysis log, and the System sends the response.

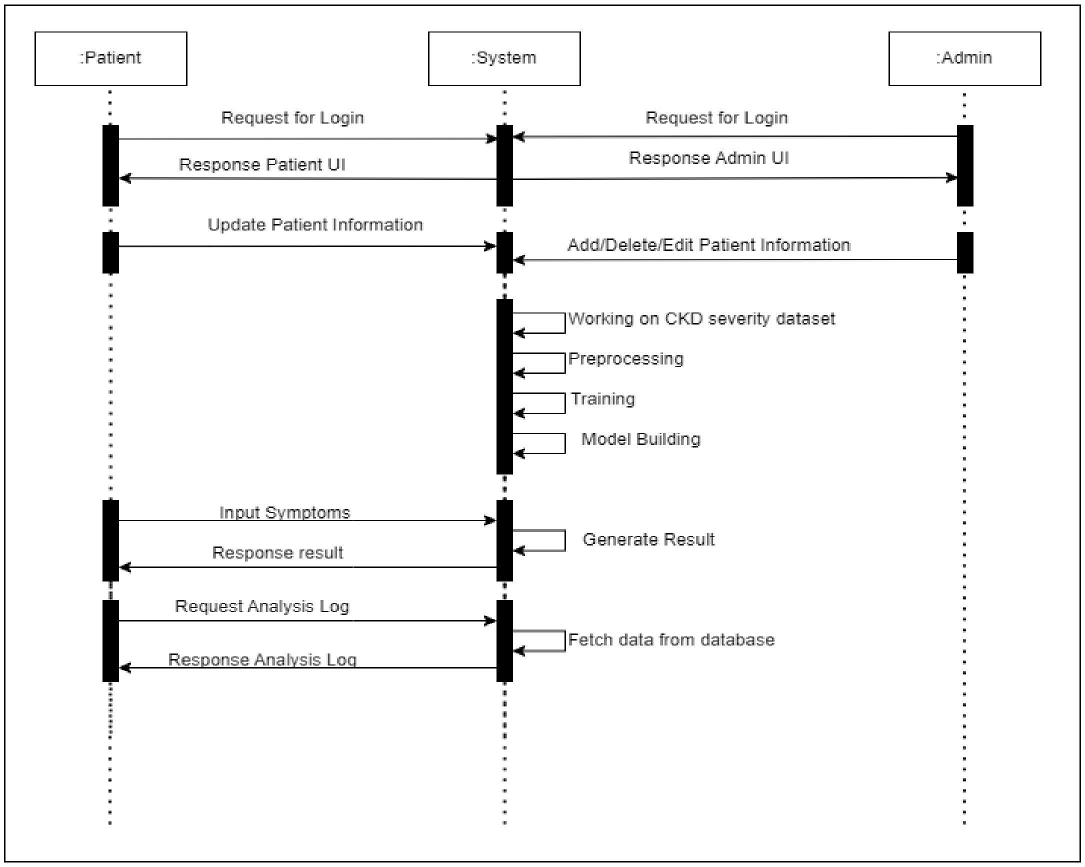


Figure 4.6: Sequence Diagram

4.1.5 Algorithm

For this CKD application, we decided to use the Naive Bayes classifier for predicting whether a patient has CKD or not. The Naive Bayes classifier is a probabilistic machine learning algorithm based on Bayes' theorem. It works by assuming that all features are independent of each other, given the class variable. This assumption simplifies the computation and allows for efficient training and prediction.

To implement the Naive Bayes algorithm in this CKD application, we first collected a dataset with 24 parameters, including age, blood pressure, and other factors that are known to be associated with CKD. We then split the dataset into training and testing sets and used the training set to fit the Naive Bayes model. Once the model was trained, we used the testing set to evaluate its performance.

The Naive Bayes algorithm proved to be effective in predicting CKD. By analyzing the 24 parameters in the input data, the algorithm was able to accurately classify whether a patient has CKD or not. The algorithm also allowed us to identify which parameters were most important in predicting CKD. This information can be used to develop more effective prevention and treatment strategies for patients with CKD.

It is based Bayes' theorem
Bayes' theorem (alternatively Bayes' law or Bayes' rule), named after Thomas Bayes, describes the probability of an event, based on prior knowledge of conditions that might be related to the event.

Bayes' theorem may be derived from the definition of conditional probability:

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Where A and B are events and $P(B) \neq 0$

- $P(A|B)$ is a conditional probability: the probability of event A occurring given that B is true. It is also called the posterior probability of A given B.

- P(B|A) is also a conditional probability: the probability of event B occurring given that A is true. It can also interpret as the likelihood of A given a fixed B because P(B|A)=L(A|B).
- P(A) and P(B) are the probabilities of observing A and B respectively without any given conditions; they are known as the prior probability and marginal probability.

CHAPTER 5: Implementation and Testing

5.1 Implementation

In the implementation phase of this Chronic Kidney Disease Prediction system using data mining algorithm, we successfully carried out all the designs created in the design section. Initially, we studied the use of the system from the use case diagram, which helped us identify the key features of the system such as the ability for users to perform analysis and view results, and for the admin to view patient details. We created a database based on the ER diagram that had entities such as analysis, which stores user-entered values, and result, which stores the prediction results. We also created the admin and user tables to store their respective details. The sequence diagram was used to sequence the various steps in the system, and this helped us to ensure that the system operated seamlessly.

We also understood the activity of the patient and admin from the activity diagram. Based on the class diagram, we created different modules for the system. Using the system and algorithm flowchart, we created the model based on the naive Bayes algorithm. We then trained the model with appropriate datasets and tested it to ensure that it predicted chronic kidney disease accurately. Finally, we integrated the modules, ran various tests, and made sure that the system functioned correctly.

In conclusion, the implementation phase was successful, and we were able to build a fully functional Chronic Kidney Disease Prediction system using data mining algorithms. The system provides a valuable tool for early detection and treatment of kidney disease, and it has the potential to improve healthcare outcomes.

5.1.1 Tool used

Chronic kidney disease (CKD) is a serious medical condition that requires proper management and monitoring. Various tools are used to help patients manage their condition, including blood pressure monitors, urine test strips, and laboratory tests to monitor kidney function. Additionally, mobile apps and online platforms are available to help patients track their medications, symptoms, and appointments. These tools can help patients better manage their CKD and improve their overall quality of life.

5.1.1.1 Frontend Tools

- HTML:

  HTML is a markup language used to create the structure and forms in a CKD application. It provides a way to define the layout and content of web pages, including headings, paragraphs, and input fields. HTML is used to create forms that allow users to enter their health information, such as blood pressure readings and medication schedules. These forms can be customized to include dropdown menus, checkboxes, and radio buttons to make data entry easier for patients.

- CSS:

  CSS is used to style the CKD application and make it more visually appealing. It provides a way to define the colors, fonts, and layout of web pages. In a CKD application, CSS can be used to make the forms and data tables more readable and easier to navigate. For example, CSS can be used to create a color scheme that distinguishes between different types of information or to highlight important data points.

- Bootstrap:

  Bootstrap is a popular front-end framework used to make responsive applications. It provides a set of pre-designed HTML and CSS components that can be easily customized to fit the needs of a CKD application. With Bootstrap, developers can create responsive layouts that adapt to different screen sizes, making the application accessible on both desktop and mobile devices.

- JavaScript:

  JavaScript (JS) is used in some sections of a CKD application to add interactivity and improve the user experience. For example, JS can be used to create pop-up windows that display additional information when a user clicks on a button. It can also be used to validate user input and provide real-time feedback to help users enter data correctly. JS is a powerful tool that can add a lot of functionality to a CKD application.

5.1.1.2 Backend Tools

- PHP and Laravel:

  On the backend side, PHP and Laravel are popular tools used to build CKD applications. PHP is a server-side scripting language that is used to generate dynamic content and interact with databases. Laravel is a PHP framework that provides a set of tools and conventions to make it easier to build web applications. It includes features like routing, middleware, and database ORM that simplify common tasks and speed up development.

- MySQL:

  MySQL is a popular database management system used to store the data in a CKD application. It provides a way to organize and retrieve data in a fast and efficient way. With MySQL, developers can create tables to store patient data, such as lab results, medical history, and appointment schedules. MySQL also provides tools for managing data integrity and security, ensuring that patient data is stored safely and accurately.

- Python:

  Python, with libraries like NumPy and Pandas, can be used to create models based on Naive Bayes algorithms. The models can be trained using patient data to predict outcomes and identify potential risk factors for CKD. The models can then be stored as pickle files and executed in a CKD application to provide real-time insights and recommendations to patients and healthcare providers. Python is a powerful tool for data analysis and machine learning, and can help improve the accuracy and effectiveness of CKD management.

5.1.2 Implementation Details of Modules

This CKD application consists of several modules that are responsible for various functions. In the implementation details section, we will describe the classes, procedures, functions, methods, and algorithms used in each module.

The first module of this application is the user authentication module, which includes classes and functions for user registration and login. The registration form accepts the user's name, email, password. Upon submission, the details are stored in the database using the MySQL database management system. The login form verifies the user's credentials and allows them to access the application.

The second module of this application is the data entry and analysis module. This module consists of classes and procedures that allow the user to input their medical data, including blood pressure, glucose level, and other parameters. The module uses a Naive Bayes algorithm to analyze the data and predict the likelihood of the user having CKD. The algorithm takes 24 parameters into account and predicts whether the user has CKD or not CKD.

The third module of this application is the feedback module. This module includes classes and procedures for the user to provide feedback on the application. The feedback form allows users to submit their comments, suggestions, and criticisms. The data from this module are also stored in the MySQL database.

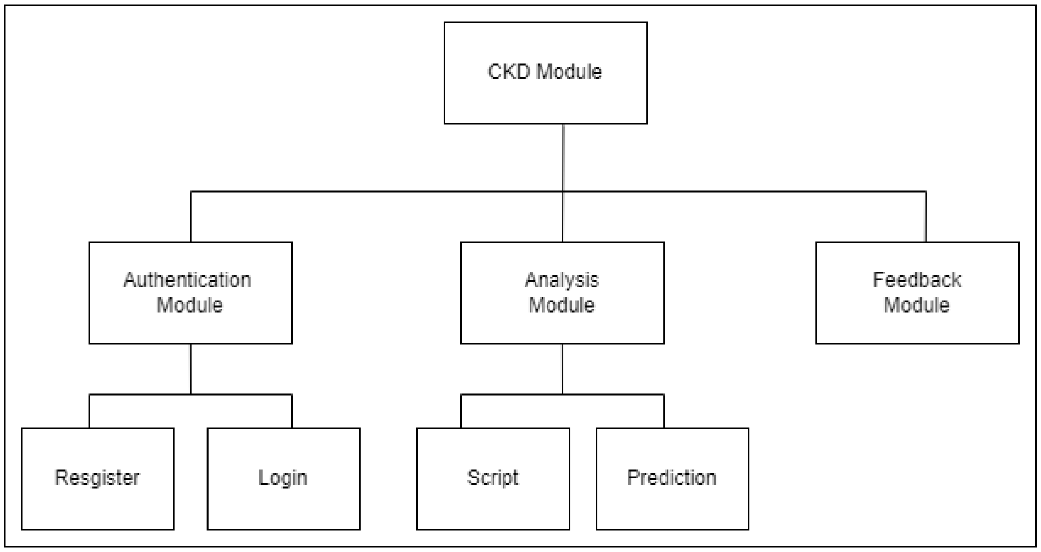


Figure 5.1: Model Hierarchy

5.2 Testing

In this CKD application, testing played a critical role in ensuring that the software met the highest standards of quality and accuracy. Manual testing was chosen as the primary approach to test the application because of its effectiveness in identifying errors and usability issues.

During the manual testing phase, the application was subjected to a battery of tests that varied in complexity and scope. The testing covered everything from basic input validation to complex algorithmic logic that was used in the CKD model. Each test was designed to simulate real-world scenarios that the application would encounter during regular usage.

We also evaluated the user experience of the application, ensuring that the interface was intuitive and user-friendly. We tested for usability issues, such as how easy it was to navigate through the various forms and how easy it was to interpret the results. This helped to ensure that users could easily use the application to access the CKD analysis and feedback features.

The team followed a comprehensive testing process that involved a range of activities such as creating test cases, executing test scenarios, and logging defects. This allowed us to track the progress of the testing phase, prioritize defects, and ensure that all issues were addressed before the application was released.

Overall, the manual testing approach was crucial in identifying and correcting errors and ensuring that the application met the highest standards of quality and accuracy. It helped to ensure that the CKD application was reliable, functional, and met the needs of its intended users.

5.2.1 Test Case for Unit Testing

In addition to manual testing, we also conducted unit testing to ensure that each unit of the application was functioning correctly. We tested each unit manually, including login, registration, analysis form, feedback form routing, and many others. We also tested the model using Jupyter notebook by passing default values and different case values in the script.py file.

Through unit testing, we were able to catch any bugs or issues at an early stage and prevent them from affecting the overall functionality of the application.

Registration

Table 5.1: Registration Test case

Test case ID	Test Scenario	Testing steps	Input Test data	Expected Result	Actual Result	Pass /Fail
TC-01	Registration with correct and valid data	1. Open the site 2. Click on sign up 3. Fill the data in the form	a) Firstname: abc b) Lastname: xyz c) Email: test1@gmail.com d) Password: Test1234!	User should be registered successfully	As expected	Pass
TC-02	Registration with invalid password	1. Open the site 2. Click on sign up 3. Fill the data in the form	a) Firstname: abc b) Lastname: xyz c) Email: test1@gmail.com d) Password: Test	Registration must fail with message “Password must be greater than 8 characters”	As expected	Pass
TC-03	Registration with some empty fields	1. Open the site 2. Click on sign up 3. Fill the data in the form	a) Firstname: abc b) Lastname: xyz c) Email: test1@gmail.com d) Password:	Registration must fail with error message “Empty field”	As expected	Pass

Login

Table 5.2: Login Test Case

Test case ID	Test Scenario	Testing steps	Input Test data	Expected Result	Actual Result	Pass /Fail
TC-01	User Login with valid data	1. Open the site 2. Click on log in 3. Fill the data in the form	a) Username: test1@gmail.com b) Password: Test1234!	User should be logged in successfully	As expected	Pass
TC-02	User login with invalid password	1. Open the site 2. Click on log in 3. Fill the data in the form	a) Username: test1@gmail.com b) Password: test	Login failed with message “Invalid usernme or password”	As expected	Pass

5.2.1 Test Case for Integration Testing

Integration testing was also conducted to verify that the CKD model was working as expected. We integrated the model with this CKD web system and tested whether it was functioning properly. We sent data from the form to the model and checked whether the output was correct. After several rounds of testing, the integration passed and the model was deemed to be functioning correctly.

Analysis Form

Table 5.3: Analysis Test Case

Test case ID	Test Scenario	Testing steps	Input Test data	Expected Result	Actual Result	Pass/ Fail
TC-01	User fills the form without missing any attribute	1. Open the site 2. Log in 3. Click Analysis Page 4. Fill the form	a) Age: 23 b) Blood Pressure: 80 c) Specific Gravity: 1.02 d) Albunin: 1 e) Sugar:0 f) Red blood cells: normal g) Pus cell: normal h) Bacteria : not present i) Blood glucose random: 121 j) Blood urea:36 k) Serum creatinine:1.2 l) Sodium:111 m) Potassium: 2.5	User form submission successful.	As expected	Pass

			n) Haemoglobin: 15.4 o) Packed cell volume:55 p) White blood cell count: 7800 q) Red blood cell count: 5.2 r) Hypertension: yes s) Diabetes mellitus: yes t) Coronary artery disease: no u) Appetite: good v) Pedal edema:no w) Anemia: no			
TC-02	User fills the form by missing some attribute	1. Open the site 2. Log in 3. Click Analysis Page 4. Fill the form	5. Age: 6. Blood Pressure: 7. Specific Gravity: 1. 8. Albunin: 9. Sugar: 10. Red blood cells: 11. Pus cell: 12. Bacteria : 13. Blood glucose random: 121 14. Blood urea:36 15. Serum creatinine:1.2 16. Sodium:111 17. Potassium: 2.5 18. Haemoglobin:	User form submission successful. Prediction based on average data	As expected	Pass

			19. Packed cell volume:55 20. White blood cell count: 21. Red blood cell count: 5.2 22. Hypertension: 23. Diabetes mellitus: yes 24. Coronary artery disease: no 25. Appetite: good 26. Pedal edema:no 27. Anemia:			

5.2.2 Test Case for System Testing

System testing was conducted to evaluate the overall functionality of the CKD application. This type of testing is done to check whether the system is working as intended and meets the user's requirements. We tested the application's functionality by integrating all the units of the application and checking how well they worked together. We checked that the input was correctly processed, and the output was accurate. We also tested the application's usability, security, and performance. After several rounds of testing, the application passed the system test, and we were confident that it was ready for deployment.

5.3 Result Analysis

This CKD application is responsible for analyzing the 24 input parameters that the user provides to the model. These parameters include various medical indicators such as age, blood pressure, specific gravity, albumin, and more. These parameters are essential in predicting the user's CKD status. This project utilized the Naive Bayes model to classify the user into two categories, CKD or notCKD, based on the provided parameters.

To provide accurate results, all 24 parameters are considered during the analysis. If a user fails to provide any of the parameters, this project use the average value of that parameter for the user. This ensures that even if the user does not have all the information, they can still receive accurate predictions.

The feedback system in this CKD application allows users to send messages to the system. This feedback helps us to improve the application further and to identify and address any issues that users may encounter. The team carefully reviews all feedback received and works to improve the application based on the user's suggestions.

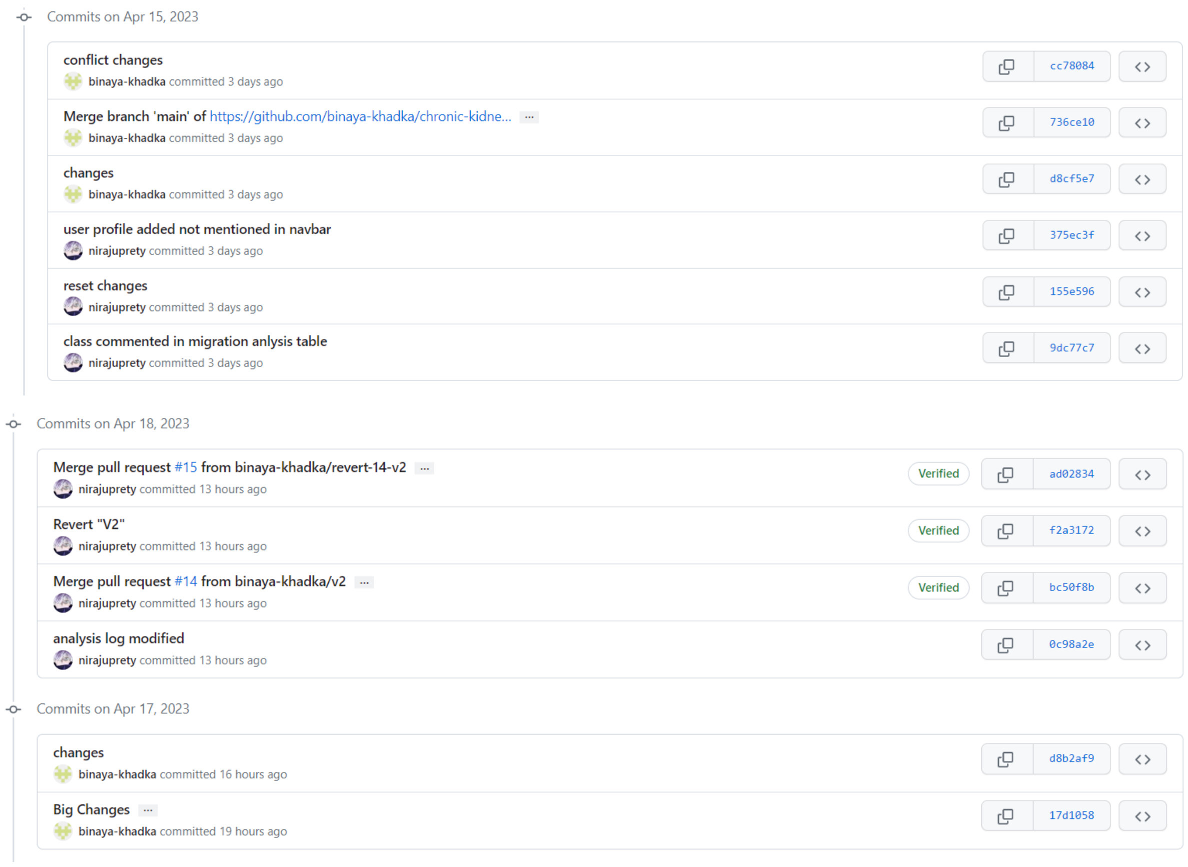
This project utilized a range of tools such as Python with libraries NumPy and Pandas to develop the Naive Bayes model used in the Result Analysis section. The model was trained using a large dataset of medical indicators and CKD status to ensure its accuracy. The model has a high degree of accuracy about 91%, and the team is continually working to refine it further.

The Result Analysis section of this CKD application is a critical component that enables users to obtain accurate predictions of their CKD status. It is designed to be intuitive and user-friendly, ensuring that users can quickly and easily understand the results of their analysis. Overall, the Result Analysis section is an essential feature of this CKD application that helps users to monitor their health status and take appropriate measures to manage CKD.

5.4 Configuration Management

Configuration management is an essential part of the software development process, and it involves tracking and controlling changes to software systems, applications, and infrastructure. The main goal of configuration management is to ensure that all components of a system are consistent and working correctly. This includes tracking changes to source code, documentation, libraries, dependencies, and other related files.

In this CKD system, we implemented configuration management by using version control systems like Git. We maintained a central repository that contains all the source code, documentation, and other related files. Each time we made changes to the system, we committed those changes to the repository with appropriate comments and descriptions. This allowed us to keep track of all changes made to the system and provided a way to revert to previous versions if necessary.



Commits on Apr 15, 2023

conflict changes
binaya-khadka committed 3 days ago
cc78084

Merge branch 'main' of https://github.com/binaya-khadka/chronic-kidne... ···
binaya-khadka committed 3 days ago
736ce10

changes
binaya-khadka committed 3 days ago
d8cf5e7

user profile added not mentioned in navbar
nirajuprety committed 3 days ago
375ec3f

reset changes
nirajuprety committed 3 days ago
155e596

class commented in migration anlysis table
nirajuprety committed 3 days ago
9dc77c7

Commits on Apr 18, 2023

Merge pull request #15 from binaya-khadka/revert-14-v2 ···
nirajuprety committed 13 hours ago
Verified
ad02834

Revert "V2"
nirajuprety committed 13 hours ago
Verified
f2a3172

Merge pull request #14 from binaya-khadka/v2 ···
nirajuprety committed 13 hours ago
Verified
bc50f8b

analysis log modified
nirajuprety committed 13 hours ago
0c98a2e

Commits on Apr 17, 2023

changes
binaya-khadka committed 16 hours ago
d8b2af9

Big Changes ···
binaya-khadka committed 19 hours ago
17d1058

Figure 5.2: Version Control

CHAPTER 6: Conclusion and Future Recommendation

6.1 Conclusion

In conclusion, the Chronic Kidney Disease Prediction system that we developed using data mining algorithms has been successfully implemented. The system was designed with the use of several diagrams, including the use case diagram, ER diagram, sequence diagram, activity diagram, and class diagram, which helped us to visualize the system's features and sequence of operations. The system was created using the naive Bayes algorithm, which was trained with appropriate datasets to ensure accurate prediction of chronic kidney disease.

This system is a valuable tool for early detection and treatment of kidney disease. It allows users to perform analysis and view results, while the admin can view patient details. With its accuracy in predicting chronic kidney disease, the system has the potential to improve healthcare outcomes and save lives. Overall, we are satisfied with the successful implementation of the system, and we hope that it will be widely used in the medical community to benefit patients suffering from chronic kidney disease.

6.2 Future Recommendation

One possible future recommendation for the Chronic Kidney Disease Prediction system is to assign a doctor to each patient based on their symptoms and results. This will ensure that patients receive personalized medical attention and treatment, which can improve their chances of recovery. Additionally, providing notes to patients based on their symptoms and results can help them understand their condition better, and can also aid in self-care management.

Another potential recommendation is to introduce an online video conference feature that connects patients with their assigned doctors. This will allow patients to receive medical advice and treatment remotely, without the need for physical appointments. This feature will be especially useful for patients who have difficulty traveling to medical facilities or live in remote areas. Online video conferences can also be used to monitor the progress of patients and make necessary adjustments to their treatment plans.

Overall, these recommendations will improve the quality of care provided to patients, while also reducing the burden on healthcare systems. With the rapid advancement of technology, it is essential to incorporate these innovative features into healthcare systems to improve access to care and outcomes for patients.

References

[1] Wang, A. Ogunleye and Qing-Guo, "XGBoost Model for Chronic Kidney Disease Diagnosis," [Online].

[2] A. Charleonnan, T. Fufaung, T. Niyomwong, W. Chokchueypattanakit, S. Suwannawach and N. Ninchawee, "Predictive Analytics for Chronic Kidney Disease Using Machine Learning Techniques," [Online].

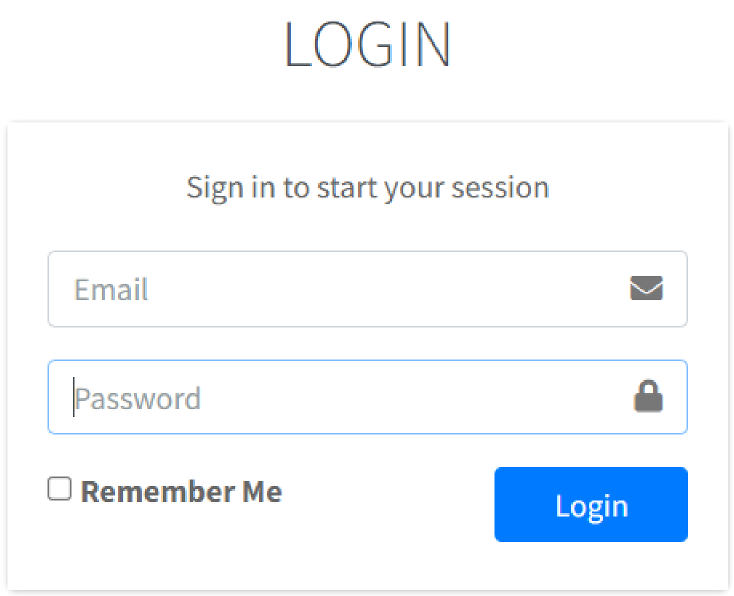
[3] G. Kaur and A. Sharma, "Predict Chronic Kidney Disease Using Data Mining Algorithms In Hadoop," [Online].

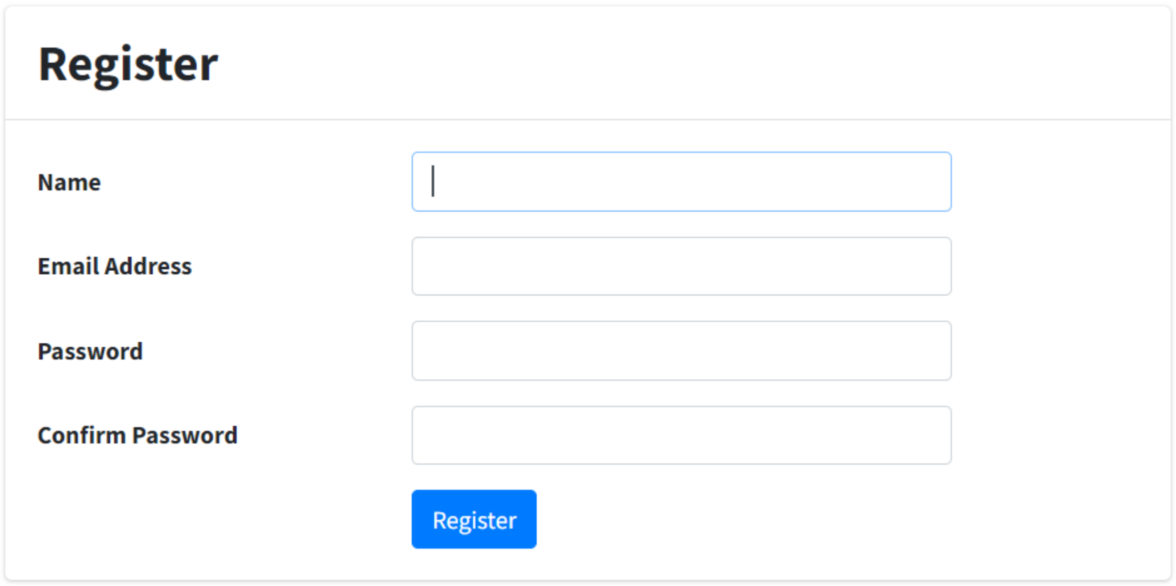
[4] Kunwar, Veenita, Chandel, Khushboo, A. S. Sabitha, Bansal and Abhay, "CHRONIC KIDNEY DISEASE ANALYSIS USING DATA MINING CLASSIFICATION TECHNIQUES," [Online].

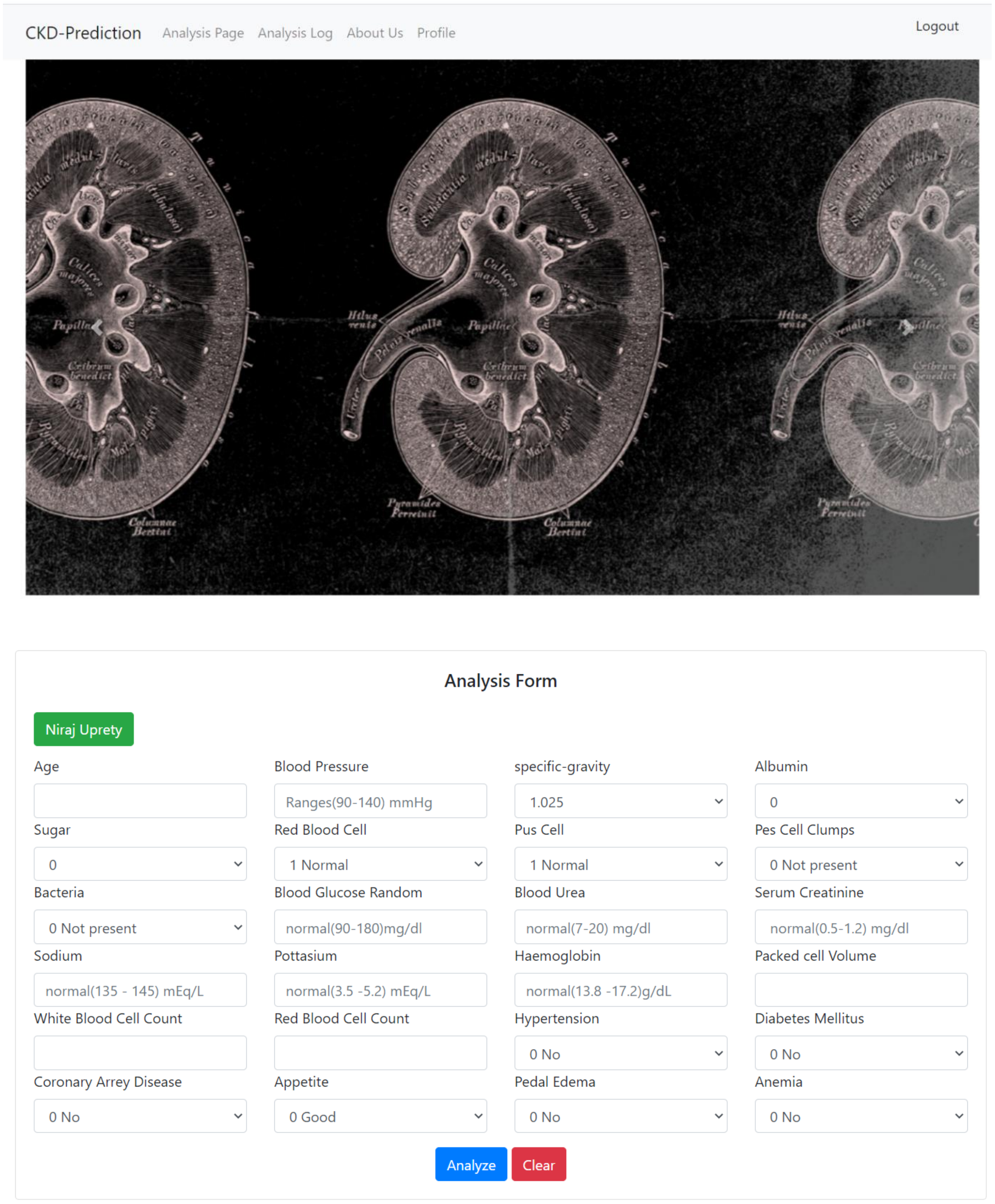
[5] Nishanth, Anandanadarajah, Thiruvaran and Tharmarajah, "Identifying important attributes for early detection of Chronic Kidney Disease," [Online].

[6] R, Devika, Avilala, S. Vaishnavi, Subramaniyaswamy and V, "Comparative Study of Classifier for Chronic Kidney Disease prediction using Naive Bayes, KNN and Random Fores," [Online].

APPENDIX


LOGIN
Sign in to start your session
Email
Password
Remember Me
Login

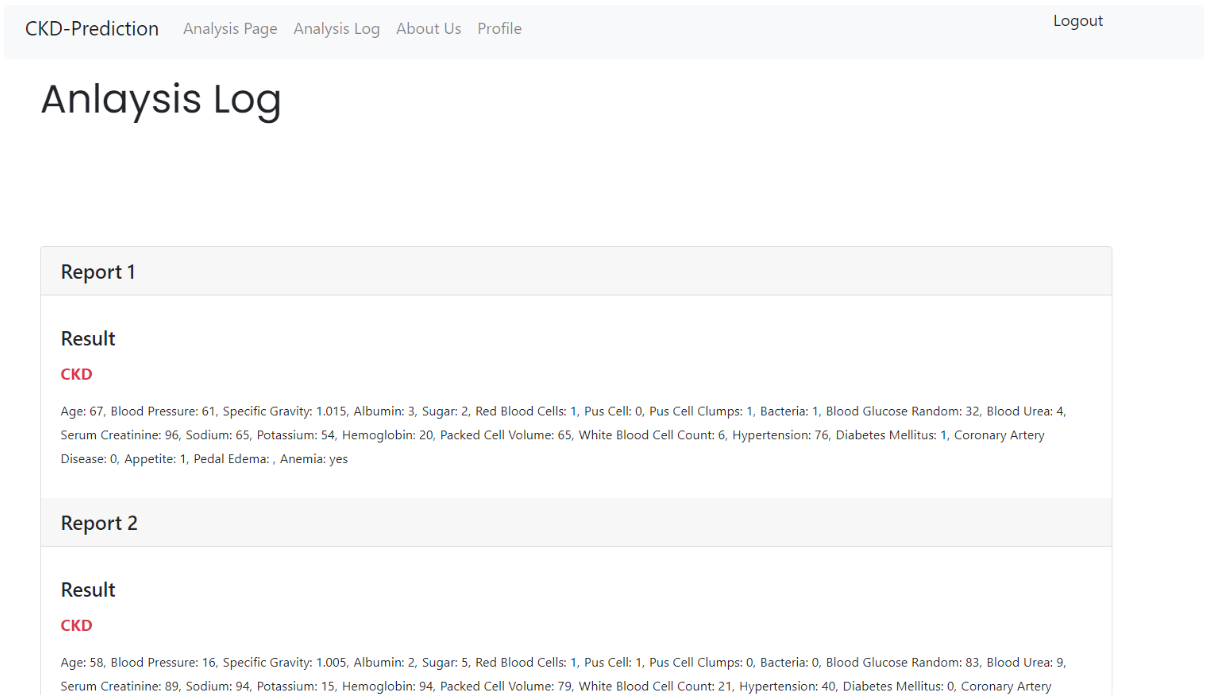

Register
Name
Email Address
Password
Confirm Password
Register



Analysis Form

Niraj Uprety

Age	Blood Pressure	specific-gravity	Albumin
	Ranges(90-140) mmHg	1.025	0
Sugar	Red Blood Cell	Pus Cell	Pes Cell Clumps
0	1 Normal	1 Normal	0 Not present
Bacteria	Blood Glucose Random	Blood Urea	Serum Creatinine
0 Not present	normal(90-180)mg/dl	normal(7-20) mg/dl	normal(0.5-1.2) mg/dl
Sodium	Pottasium	Haemoglobin	Packed cell Volume
normal(135 - 145) mEq/L	normal(3.5 -5.2) mEq/L	normal(13.8 -17.2)g/dL	
White Blood Cell Count	Red Blood Cell Count	Hypertension	Diabetes Mellitus
		0 No	0 No
Coronary Arrey Disease	Appetite	Pedal Edema	Anemia
0 No	0 Good	0 No	0 No



Anlaysis Log

Report 1

Result

CKD

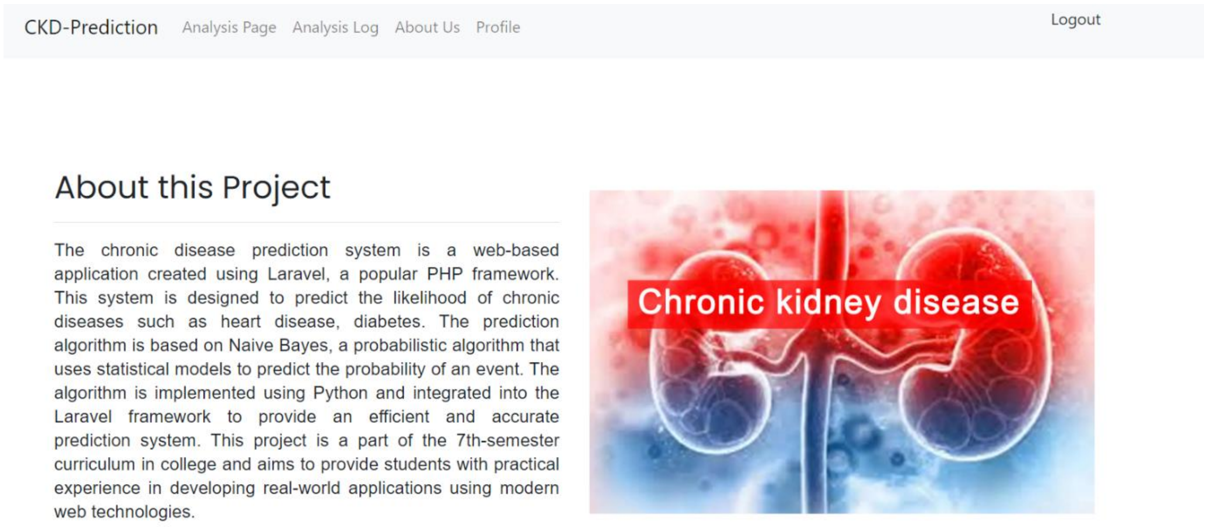
Age: 67, Blood Pressure: 61, Specific Gravity: 1.015, Albumin: 3, Sugar: 2, Red Blood Cells: 1, Pus Cell: 0, Pus Cell Clumps: 1, Bacteria: 1, Blood Glucose Random: 32, Blood Urea: 4, Serum Creatinine: 96, Sodium: 65, Potassium: 54, Hemoglobin: 20, Packed Cell Volume: 65, White Blood Cell Count: 6, Hypertension: 76, Diabetes Mellitus: 1, Coronary Artery Disease: 0, Appetite: 1, Pedal Edema: , Anemia: yes

Report 2

Result

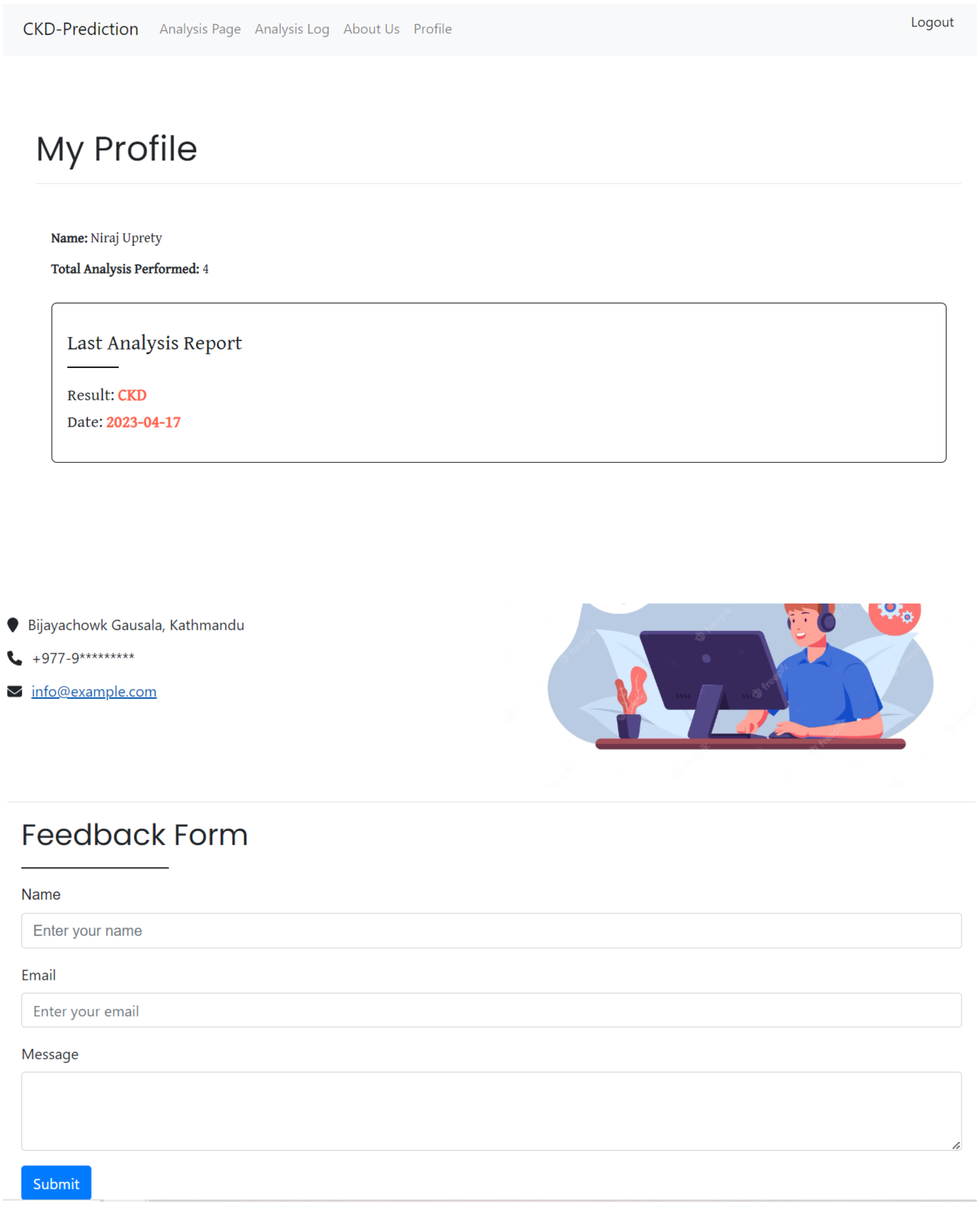
CKD

Age: 58, Blood Pressure: 16, Specific Gravity: 1.005, Albumin: 2, Sugar: 5, Red Blood Cells: 1, Pus Cell: 1, Pus Cell Clumps: 0, Bacteria: 0, Blood Glucose Random: 83, Blood Urea: 9, Serum Creatinine: 89, Sodium: 94, Potassium: 15, Hemoglobin: 94, Packed Cell Volume: 79, White Blood Cell Count: 21, Hypertension: 40, Diabetes Mellitus: 0, Coronary Artery



About this Project

The chronic disease prediction system is a web-based application created using Laravel, a popular PHP framework. This system is designed to predict the likelihood of chronic diseases such as heart disease, diabetes. The prediction algorithm is based on Naive Bayes, a probabilistic algorithm that uses statistical models to predict the probability of an event. The algorithm is implemented using Python and integrated into the Laravel framework to provide an efficient and accurate prediction system. This project is a part of the 7th-semester curriculum in college and aims to provide students with practical experience in developing real-world applications using modern web technologies.



CKD-Prediction Analysis Page Analysis Log About Us Profile Logout

My Profile

Name: Niraj Uprety

Total Analysis Performed: 4

Last Analysis Report

Result: **CKD**

Date: **2023-04-17**

Bijayachowk Gausala, Kathmandu

+977-9*********

info@example.com

Feedback Form

Name

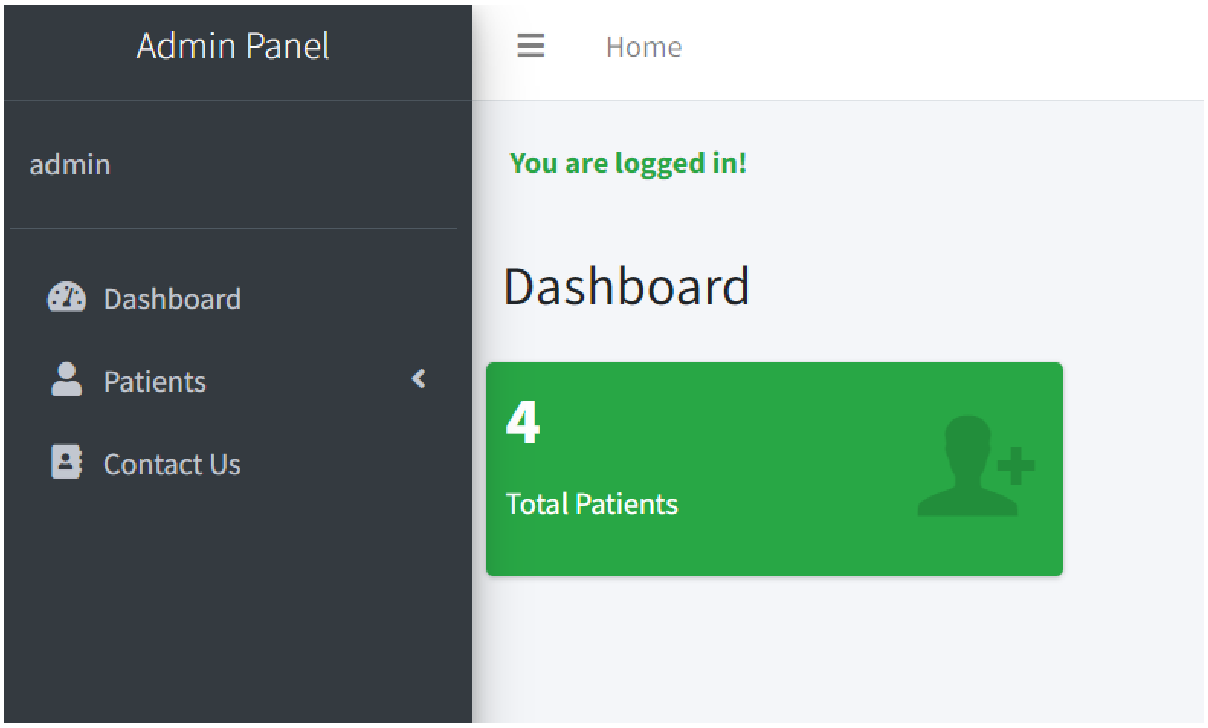
Enter your name

Email

Enter your email

Message

Submit

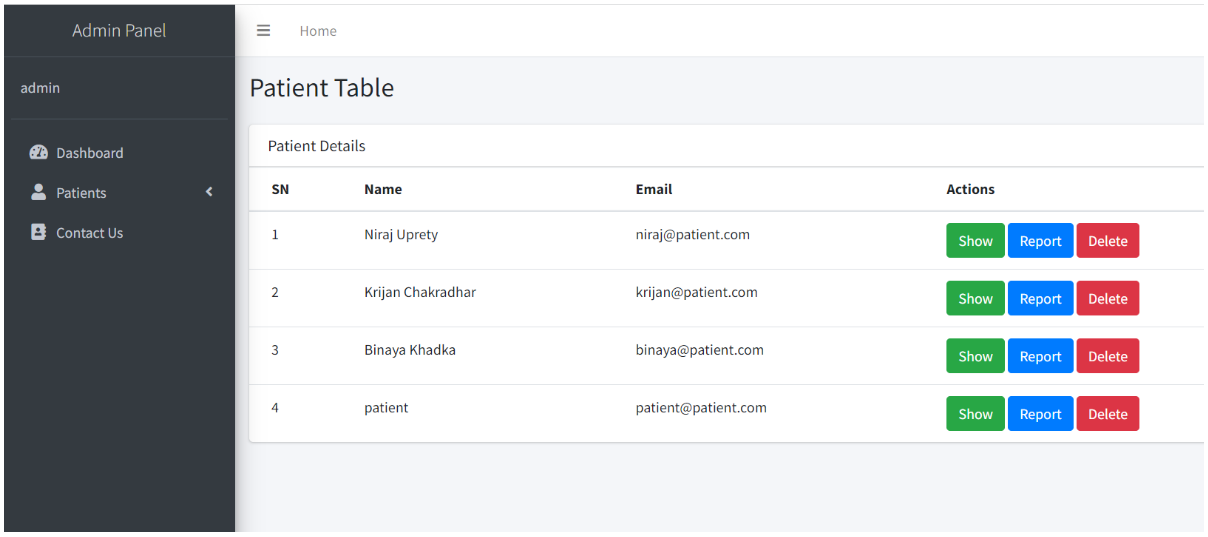


You are logged in!

Dashboard

4

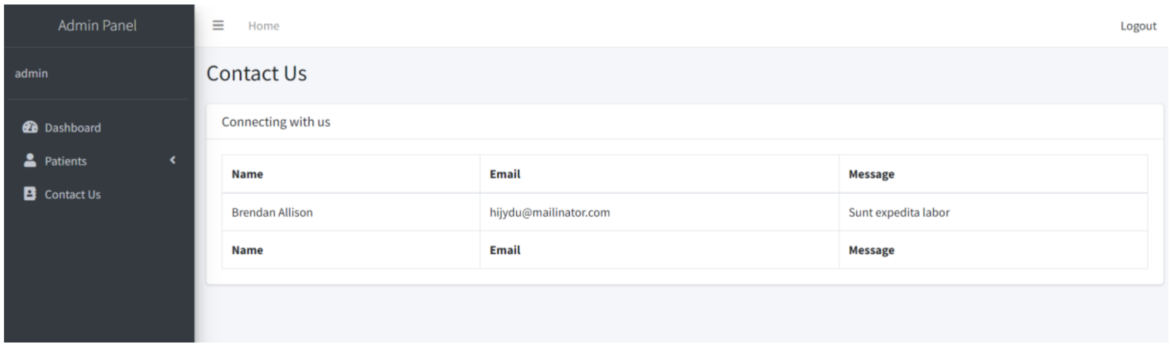
Total Patients



Patient Table

Patient Details

SN	Name	Email	Actions
1	Niraj Uprety	niraj@patient.com	Show Report Delete
2	Krijan Chakradhar	krijan@patient.com	Show Report Delete
3	Binaya Khadka	binaya@patient.com	Show Report Delete
4	patient	patient@patient.com	Show Report Delete



Contact Us

Connecting with us

Name	Email	Message
Brendan Allison	hijydu@mailinator.com	Sunt expedita labor
Name	**Email**	**Message**

Major snippet of ML Model

```
import math

def gaussian_pdf(x, mean, std):

    """Compute the probability density function of a Gaussian distribution"""

    exponent = np.exp(-((x - mean) ** 2) / (2 * std ** 2))

    return (1 / (np.sqrt(2 * np.pi) * std)) * exponent


class GaussianNaiveBayes:

    def fit(self, X, y):

        self.classes = np.unique(y)

        self.mean = np.zeros((len(self.classes), X.shape[1]))

        self.std = np.zeros((len(self.classes), X.shape[1]))

        self.prior = np.zeros(len(self.classes))


        for i, c in enumerate(self.classes):

            X_c = X[c == y]

            self.mean[i, :] = X_c.mean(axis=0)

            self.std[i, :] = X_c.std(axis=0)

            self.prior[i] = X_c.shape[0] / X.shape[0]


    def predict(self, X):

        posteriors = np.zeros((X.shape[0], len(self.classes)))
```

```
for i, c in enumerate(self.classes):
    likelihood = gaussian_pdf(X, self.mean[i, :], self.std[i, :])
    posteriors[:, i] = np.log(likelihood).sum(axis=1) + np.log(self.prior[i])

log_probabilities = posteriors - np.max(posteriors, axis=1)[:, np.newaxis]
probabilities = np.exp(log_probabilities)
confidences = np.max(probabilities, axis=1) - np.partition(probabilities, -2, axis=1)[:,-2]
predicted_labels = self.classes[np.argmax(posteriors, axis=1)]

return predicted_labels, confidences
```