1.INTRODUCTION

1.1 PROEJCT OVERVIEW

Chronic kidney disease, also called chronic kidney failure, involves a gradual loss of kidney function. Your kidneys filter wastes and excess fluids from your blood, which are then removed in your urine. Advanced chronic kidney disease can cause dangerous levels of fluid, electrolytes and wastes to build up in your body. In the early stages of chronic kidney disease, you might have few signs or symptoms. You might not realize that you have kidney disease until the condition is advanced. Treatment for chronic kidney disease focuses on slowing the progression of kidney damage, usually by controlling the cause. But even controlling the cause might not keep kidney damage from progressing. Chronic kidney disease can progress to end-stage kidney failure, which is fatal without artificial filtering (dialysis) or a kidney transplant. Without leading to vital stage of Kidney disease, Machine Learning algorithms can be used to predict the earlier detection. Early detection and cure of chronic kidney disease (CKD) is extremely desirable as it can lead to the prevention of unwanted consequences. Machine learning methods are being extensively advocated for early detection of symptoms and diagnosis of several diseases recently. With the same motivation, the aim of this study is to predict the various stages of CKD using machine learning classification algorithms on the dataset obtained from the medical records of affected people. Specifically, we have used the Random Forest algorithm to obtain a sustainable and practicable model to detect various stages of CKD with comprehensive medical accuracy. CKD, i.e., gradual decrease in the renal function spanning over a duration of several months to years without any major symptoms, is a life-threatening disease. It progresses in six stages according to the severity level. It is categorized into various stages based on the Glomerular Filtration Rate (GFR), which in turn utilizes several attributes, like age, sex, race, and Serum Creatinine. Among multiple available models for estimating GFR value, Chronic Kidney Disease Epidemiology Collaboration (CKD-EPI), which is a linear model, has been found to be quite efficient

because it allows detecting all CKD stages. CKD is often diagnosed in later stages when dialysis or kidney transplant are the only options left to save the patient's life. Whereas an early diagnosis can lead to the prevention of kidney failure. The best way to measure the kidney function or to predict the stages of kidney disease is to monitor the Glomerular Filtration Rate (GFR) on regular basis.

1.2 PURPOSE

A combination of estimated glomerular filtration rate (GFR), age, diet, existing medical conditions, and albuminuria can be used to assess the severity of kidney disease but requires more accurate information about the risk to the kidney is required to make clinical decisions about diagnosis, treatment, and referral. The purpose of this model is to develop and validate predictive models for chronic kidney disease. The main goal will be to evaluate kidney failure, which means the need for kidney dialysis or kidney transplant first. These models also teach the patient how to live a healthy life and help the doctor see the risk and severity of the disease, as well as how to proceed with the treatment in the future. It may be possible to identify patterns of data collection using ANN, mining methods, and the future occurrence of certain diseases that may cause harm can be predicted in advance. The purpose of the proposed model is to predict whether the patient will suffer or develop chronic kidney disease in the future if he continues their lifestyle. This information can be used to determine whether the kidney disease is using eGFR (glomerular filtration rate), which helps the doctor plan the appropriate treatment. Estimated glomerular filtration rate (eGFR) defines the degree of kidney disease and measures kidney function. The main function of the kidney is to filter the blood in the body. Kidney disease is a silent killer because it can cause kidney failure without causing any symptoms or concern. Chronic kidney disease is defined as a decline in kidney function over a period of months or years. Kidney disease is often caused by diabetes and high blood

pressure. Chronic kidney disease is a major health problem that affects people worldwide. Not getting the right treatment for chronic kidney disease can have serious consequences, affecting people who cannot afford it. Glomerular filtration rate (GFR) is the most accurate test to determine your kidney function and the degree of chronic kidney disease. Blood creatinine level, age, gender, and other characteristics can be used to calculate it. In most cases it is better to get sick early. Therefore, it ispossible to prevent serious illness. Chronic Kidney Disease is detected and analyzed with the help of Random Forest Algorithm. Random Forest is an algorithm that is used for supervised classification. It creates a forest of many trees to calculate the accuracy efficiently. The accuracy for this classifier is directly proportional to the number of trees. The results produced by Random Forest, even without hyper-parameter tuning, are more reliable because of its flexibility. It is simple and works very efficiently especially when the size of data set is large. It retains the accuracy rate by recognizing outliers and anomalies. However, it is not direct to implement and is computationally expensive.

2.LITERATURE SURVEY

2.1 EXISITING PROBLEM

The below table contains the list of articles that includes the existing problem.

S.No	YEAR	AUTHOR NAME	TITLE	ALGORITHM	DRAWBACKS
01	2022	Saurabh Pal	Chronic Kidney Disease Prediction Using Machine Learning Methods	Decision Tree, GFR, SVM, Machine Learning	It has been a challenging task to discover the correct set of attributes.
02	2021	Vineeta Gulatiand Neeraj Raheja	Comparative Analysis for prediction of kidney diseases using Intelligent Machine Learning Models	K-Nearest Neighbors, Logistic Regression, Decision Tree method Random Forest and, Naïve Bayes, Support Vector Machine and Multi-Layer Perceptron Algorithm	Sometimes setting a biasing values and activation function may consume more time for processing the inputs.
03	2021	Barot mitisha1, prof. Barkha bhavsa	Comparative Analysis for Prediction of Kidney Disease Using Intelligent Machine Learning Methods.	KNN, DT, NB, and SB classifiers	Individual F1-scores are 95% for non-CKD and 97% for CKD.In this, there lies a problem in the selection of best suitable attributes.
04	2019	Jing Xiao and Ruifeng Ding et.al	Comparison and development of machine learning tools in the prediction	Logistic regression, k-Nearest Neighbors(KNN) regression, SVC, Gaussian NB, decision tree	This requires the huge amount of dataset collection. Those data should be pre-processed by

			of chronic kidney disease progression.	classifier, Random Forest classifier.	an efficient method.
05	2018	Siddheshwar Tekale	Prediction of Chronic Kidney Disease Using Machine Learning Algorithm	Logistic regression, Elastic Net, lasso regression, ridge regression, support vector machine, random forest, XGBoost, neural network and k-nearest neighbor.	It has the problem of conducting the performance analysis criteria as there is need for specifying a optimum criteria value.

Table 2.1

2.2 REFERENCES

1. Saurabh Pal - Chronic Kidney Disease Prediction Using Machine Learning Methods.Issue:16,August 2022
2. Vineeta Gulatiand Neeraj Raheja - Comparative Analysis for prediction of kidney Diseases using Intelligent Machine Learning Models.Issue:2021
3. Barot mitishal - Prior Stage Kidney Disease Prediction Using AI & Supervised Machine Learning.Issue:12,Dec 2021
4. Jing Xiao and Ruifeng Ding et.al - Comparison and development of machine learning tools in the prediction of chronic kidney disease progression.Issue:2019
5. Siddheshwar Tekale,Pranjal Shingavi et.al - Prediction of Chronic Kidney Disease Using Machine Learning Algorithm.Issue:10.Oct 2018

2.3 PROBLEM STATEMENT DEFINITION

The kidney is one of the most important body organs that filtrates all the wastes and water from human body to make urine. Chronic Kidney Disease (CKD), also commonly known as chronic renal disease or chronic kidney failure, is a life-threatening disease that is attributed to the failure of the kidney in performing its routine functionality. It leads to the continuous decrease of Glomerular Filtration Rate (GFR) for a period of 3 months or more and is a universal health problem. Some common symptoms of the disease include hypertension, irregular foamy urine, vomiting, shortness of breath, itching and cramps, whereas high blood pressure and diabetes are the main cause of this disorder's is often diagnosed in later stages when dialysis or kidney transplant are the only options left to save the patient's life. Whereas an early diagnosis can lead to the prevention of kidney failure. The best way to measure the kidney function or to predict the stages of kidney disease is to monitor the Glomerular Filtration Rate (GFR) on regular basis. GFR is calculated using age, gender, race, and blood creatinine value of a person. Symptoms of CKD are not disease specific. The symptoms develop gradually, and some patients may not have any symptoms at all. Hence, it becomes complicated to detect the disease at early stages. Machine Learning (ML) has recently played a significant role for the diagnosis of diseases by just analyzing the records of existing patients and training a model to predict the behavior of new patients. ML is a branch of Artificial Intelligence in which the computing machine learns automatically and thus the prediction gets better from training experiences. A category of ML is supervised learning which may be used for regression or classification of dataset. ML is being used very effectively in different domains, especially, in the biomedical field for the detection and classification of several diseases. Different ML algorithms may be used to predict diseases with each one having its own strength and weaknesses. Among these, decision-tree provides classified reports for kidney related diseases with more accuracy. Thus, it seems quite suitable to be used to build a prediction system to diagnose kidney diseases at early stage.

3.IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

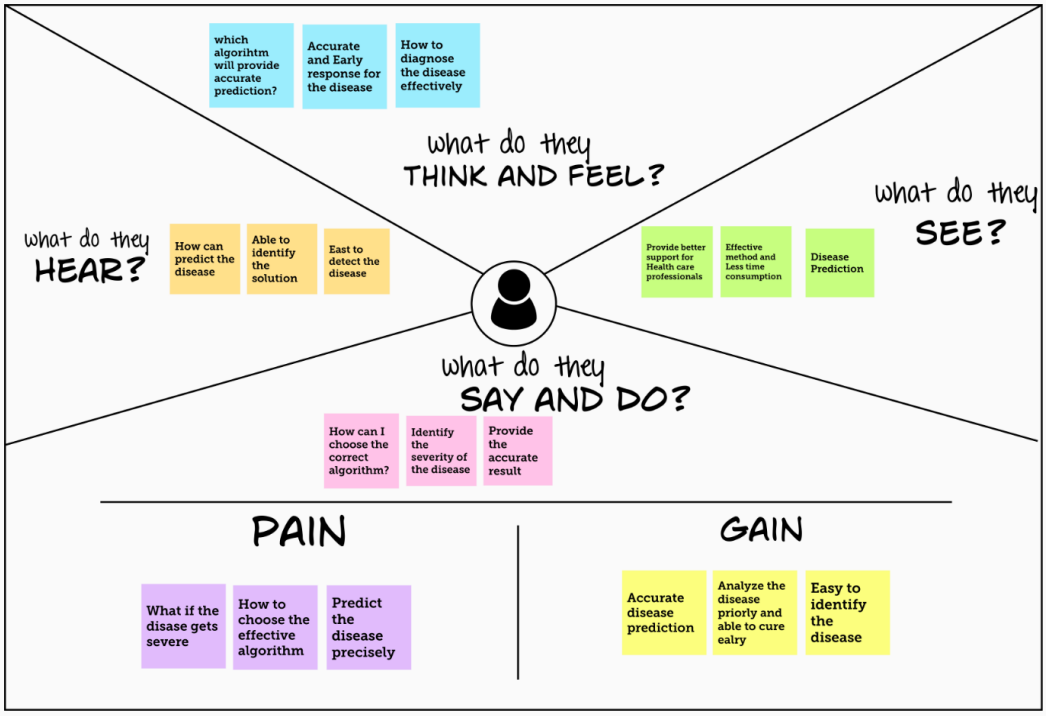

Fig 3.1

3.2 IDEATON AND BRAINSTORMING

Brainstorm and Idea Prioritization Template:

Step-1: Team Gathering, Collaboration and Select the Problem Statement

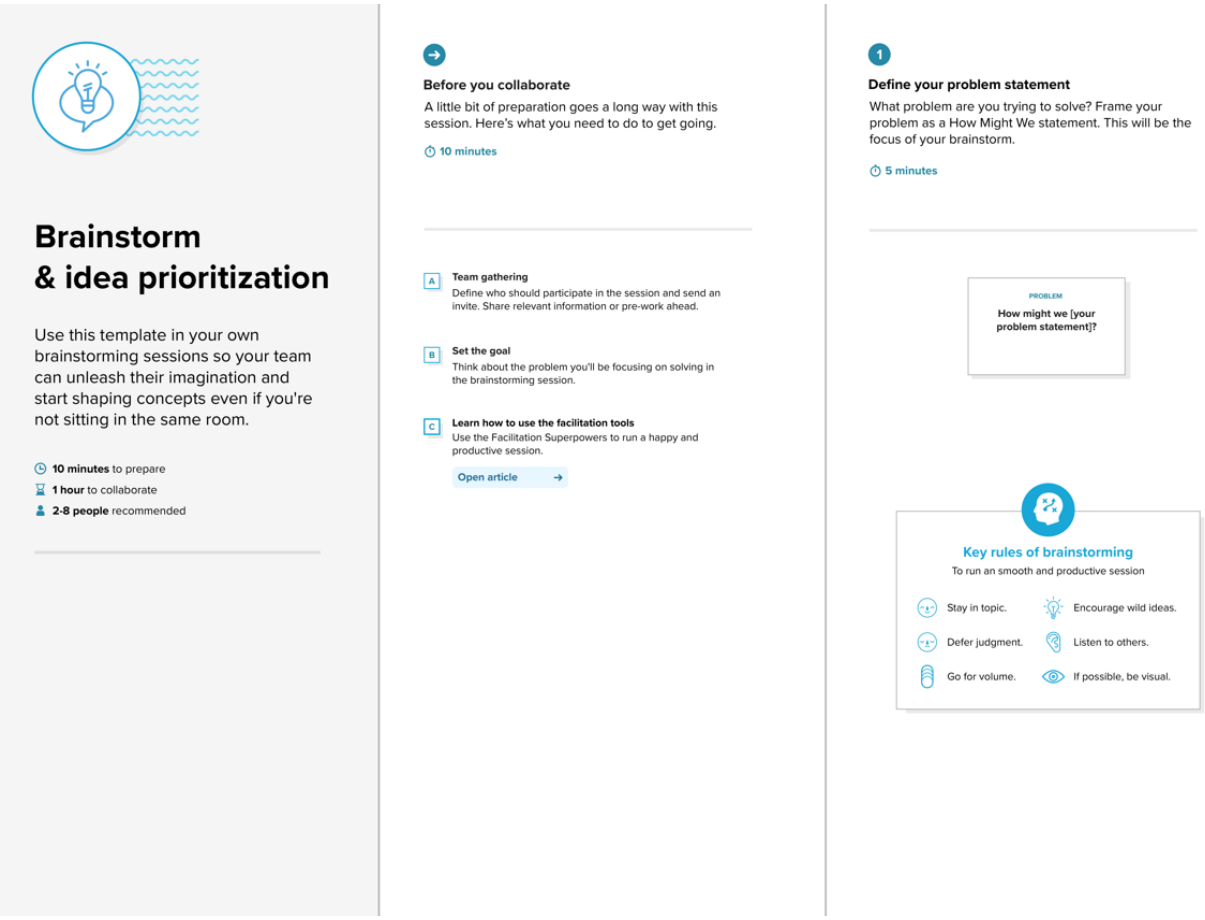

Fig 3.2.1

Step-2: Brainstorm, Idea Listing and Grouping

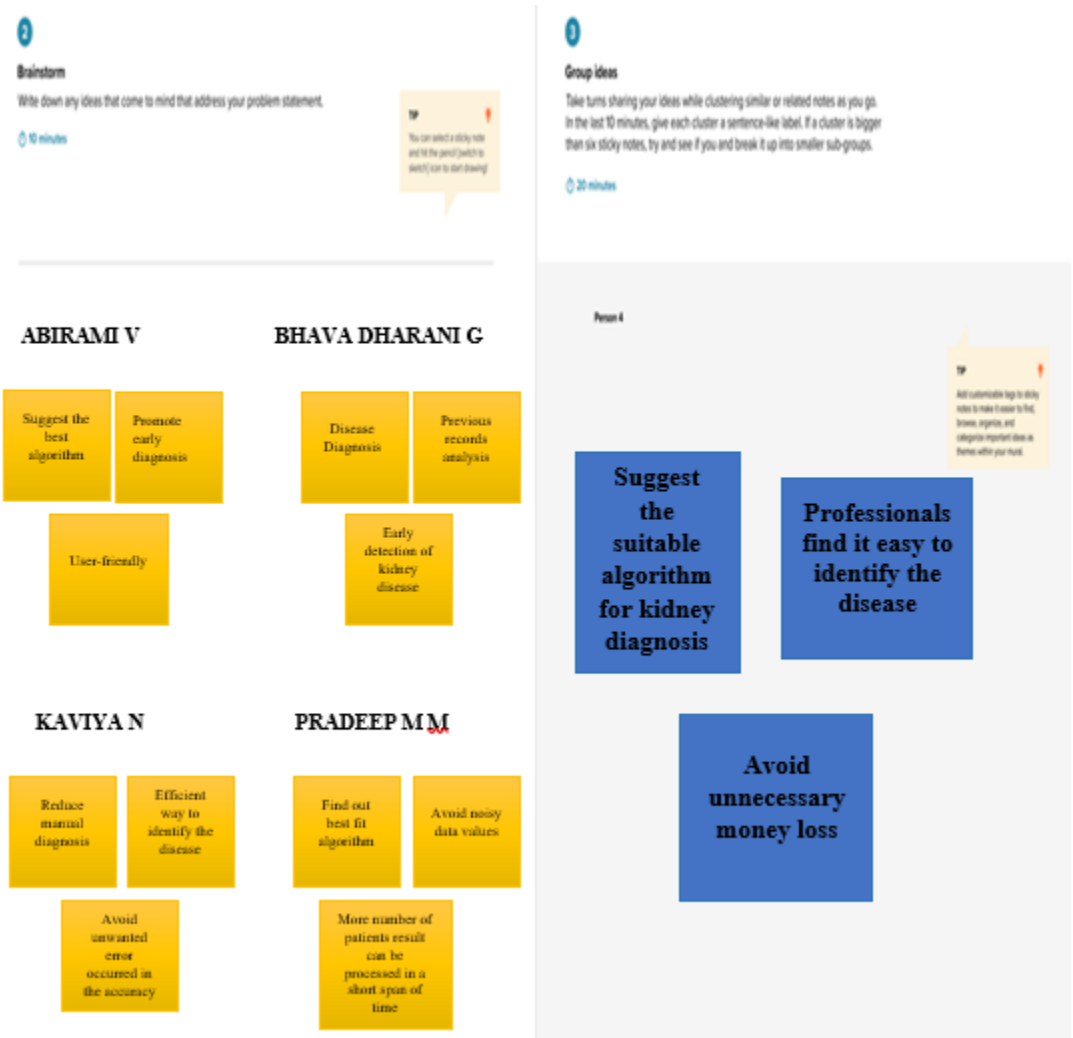

Fig 3.2.2

Step-3: Idea Prioritization

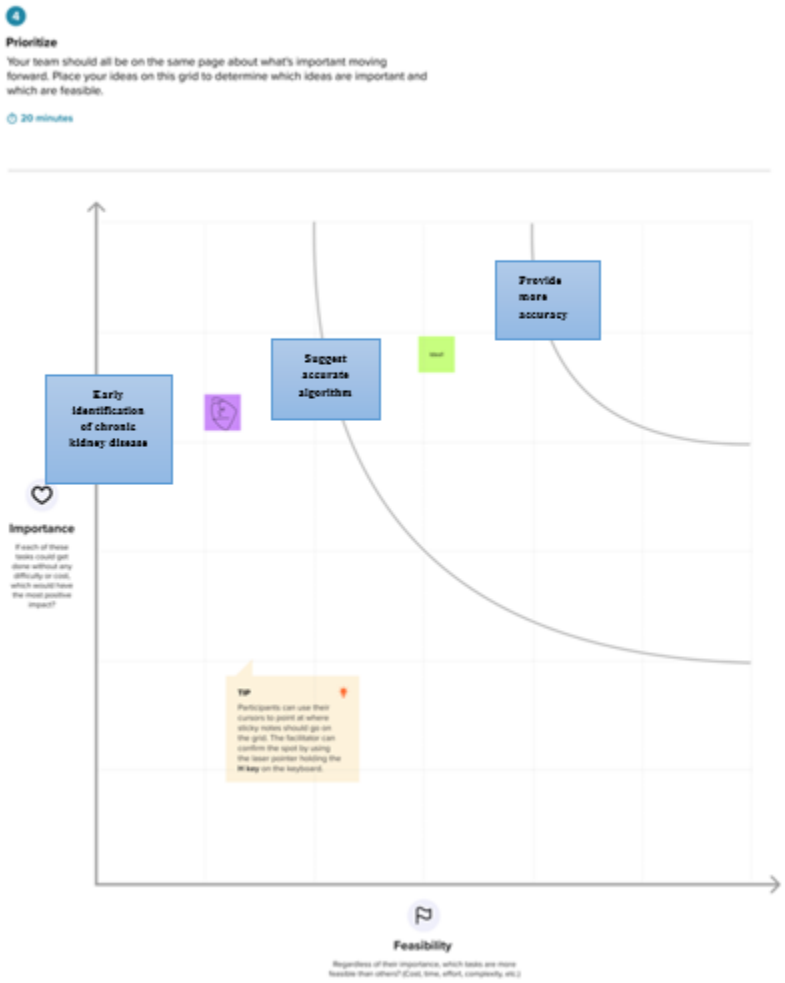

Fig 3.2.3

3.3 PROPOSED SOLUTION

Project team shall fill the following information in proposed solution template.

S. No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Early Detection of Chronic Kidney Disease using Machine Learning.
2.	Idea / Solution description	Early detection and cure of Chronic Kidney Disease (CKD) is extremely desirable as it can lead to the prevention of unwanted consequences. Machine learning methods are used to predict the various stages of CKD using the dataset obtained from the medical records of affected people.
3.	Novelty / Uniqueness	Specifically, we have used the Random Forest and J48 algorithms to obtain a sustainable and practicable model to detect various stages of CKD with comprehensive medical accuracy.
4.	Social Impact / Customer Satisfaction	As the Chronic Kidney Diseases (CKD) is a silent disease, as most sufferers have no symptoms until kidney function drops. Our project can be a huge game changer in a medical field by helping doctors to predict the CKD in a early stage and to the lives of thousands of people.
5.	Business Model (Revenue Model)	This application is recommended to patients in low cost with subscription basis.
6.	Scalability of the Solution	Machine-learning methods can be employed to analyze nanomaterials better and nanoscale biological materials, and aid in the effort to find new materials and the best routes to design nanomaterials optimally.

Table 3.3

3.4 PROPOSED SOLUTION FIT

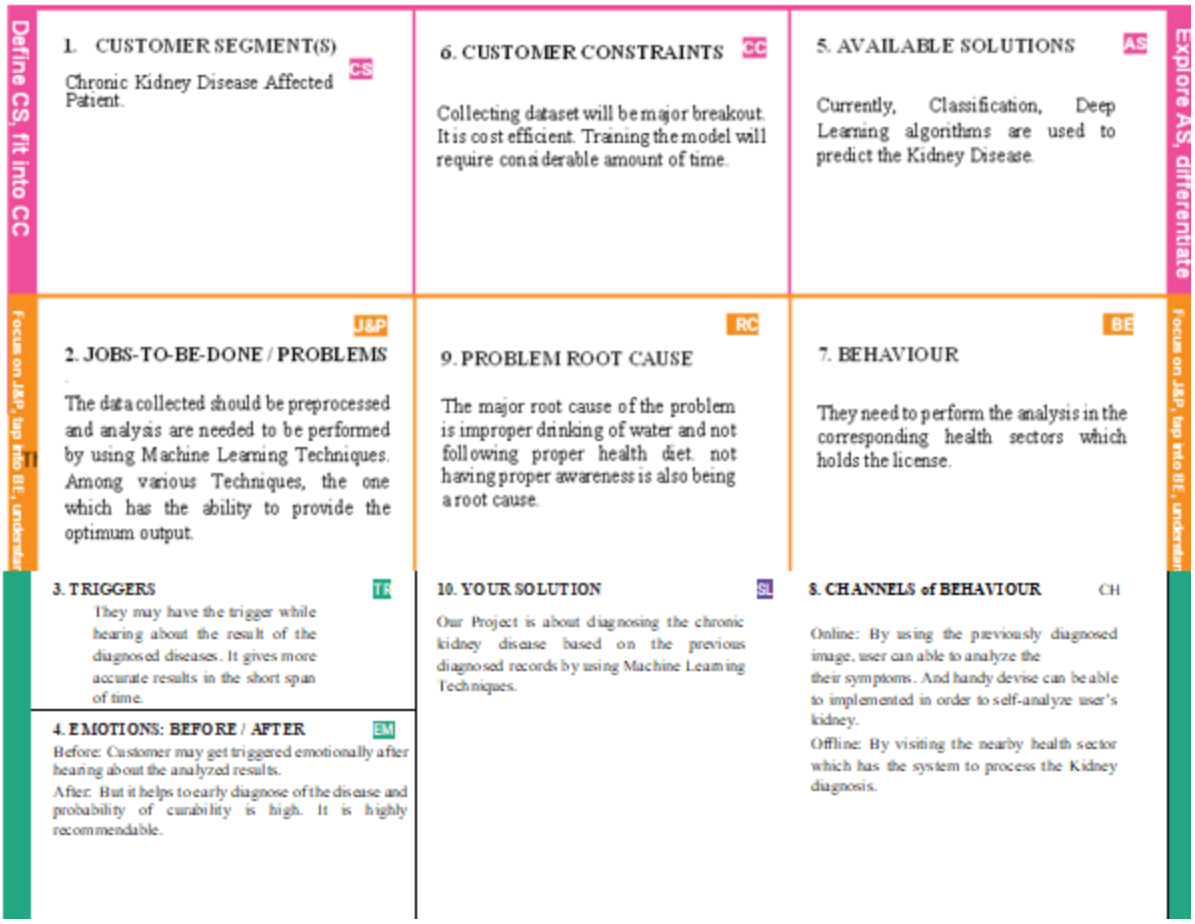

Fig 3.4

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP Confirmation via Phone number.
FR-3	Capturing image	Capture the image of the kidney by using Radioactive material and check the parameter of the scanned image.
FR-4	Capturing image	Upload the image for the prediction of the disease in the kidney.
FR-5	Kidney Identification	Identify the kidney and predict the disease in the kidney.
FR-6	Image Description	Suggestion the best method for diagnosing the disease.

Fig 4.1

4.2 NON-FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Datasets of all the kidney is used to find the disease that present in the kidney.
NFR-2	Security	The information belongs to the user and kidney are secured highly without vulnerable to the malicious users.
NFR-3	Reliability	The dataset collected on the kidney should be important for predicting the disease in the kidney.
NFR-4	Performance	Performance is based on the collected dataset which is used for disease prediction.
NFR-5	Availability	It is available for all the user who tend to predict the disease in the kidney.
NFR-6	Scalability	Increasing the analysis range for the prediction of disease in the kidney.

Fig 4.2

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

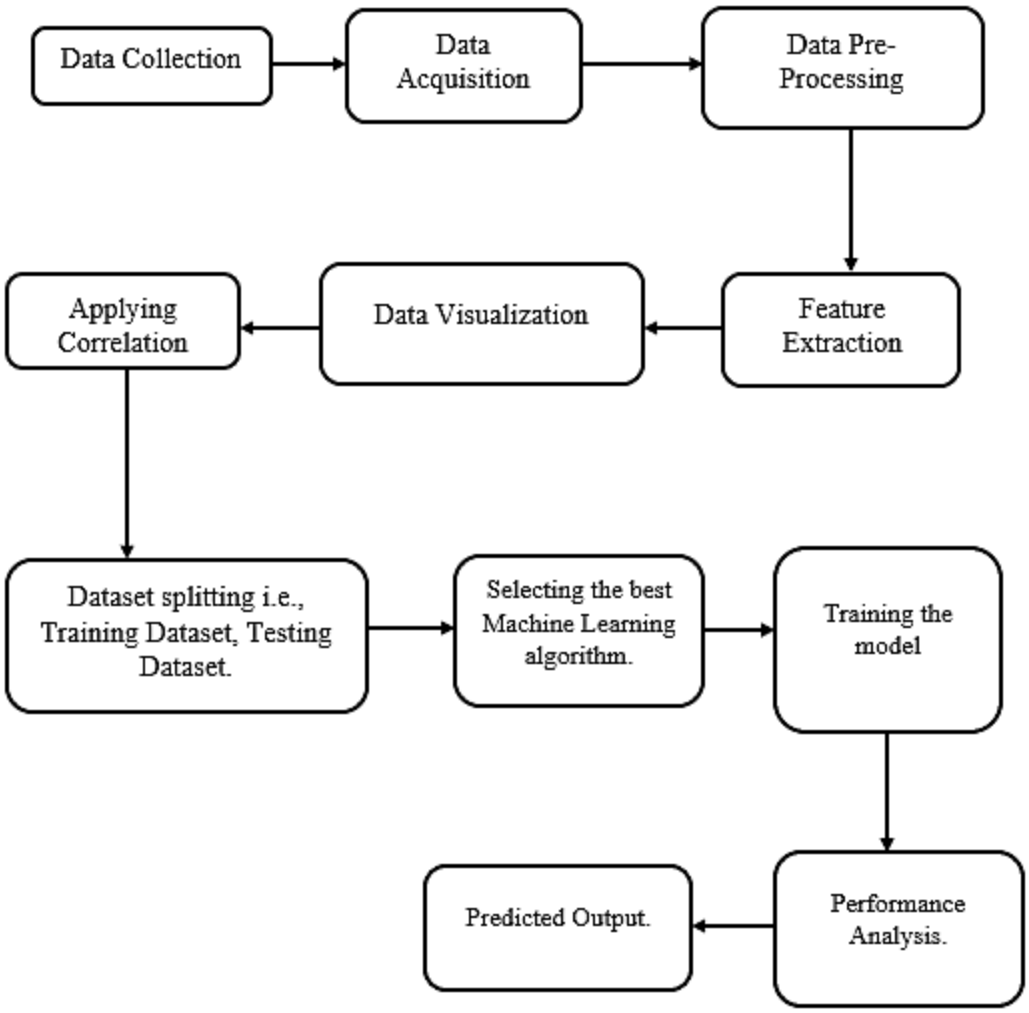

Fig 5.1

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

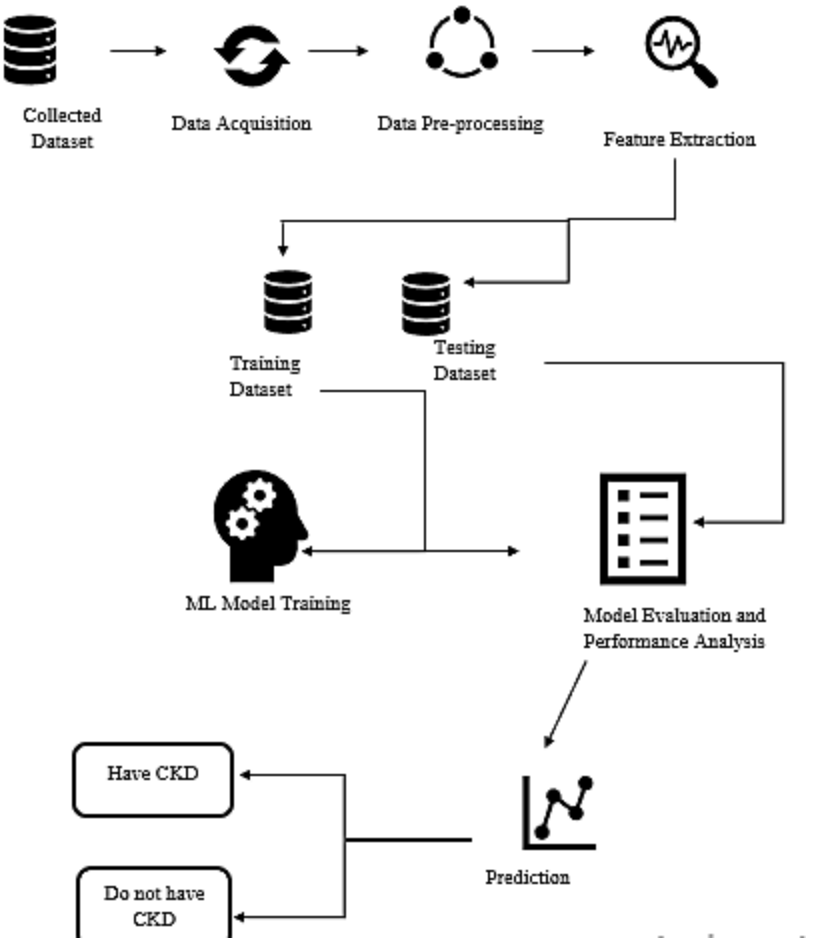

Fig 5.2

Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	User interacts by using web user interface.	HTML, CSS and Python Flask
2.	Application Logic-1 (Login)	User can able to login if that person is already registered to the site.	HTML, CSS and Python Flask
3.	Application Logic-2 (Register)	User needs to be registered if that person is new to the site.	HTML, CSS and Python Flask.
4.	Application Logic-3(Reporting Form)	User needs to click on the reporting form in order to get the prediction result	Front end- HTML, CSS and Python Flask. Back end – Query Languages, Python.
5.	Database	Data Type-String, Numeral values.	Query Languages such as MySQL, NoSQL etc.
6.	Cloud Database	Database Service on Cloud.	IBM DB2, IBM Cloud ant etc.
7.	File Storage	File storage requirements.	Local File-system.
8.	External API-1	Anyone can access the details with some restrictions to the personal details of other users.	Web API.
9.	External API-2	Accessibility.	Aadhar API.
10.	Machine Learning Model	Predict the result based on the training and testing dataset.	Data Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System.	Local System.

Table 5.2.1

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks are used for predictive data analysis, providing clear and actionable error messages.	Tensor flow, Sci-kit learn, Keras.
2.	Security Implementations	OTP will be sent to the registered email id. Unauthorized users could not access the user's details.	Email Verification.
3.	Scalable Architecture	Scalability is improved for implementing the three-tier architecture.	Three tier architecture.
4.	Availability	For enhancing the high availability, load balancer is needed.	Load Balancer.
5.	Performance	The model could be able to process large number of datasets.	Load Balancer.

Table 5.2.2

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the site by entering my email, password, and confirming my email id.	I can access my account and view the details.	High	Sprint-1
	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the site.	I can receive confirmation email and click confirm.	High	Sprint-1
	Login	USN-3	As a user, I can login into the site by clicking onto the login link.	I can successfully login to the page and my details will be shown.	Low	Sprint-2
	Dashboard	USN-4	As a user, I can access my dashboard.	I can modify the details in the dashboard.	Medium	Sprint-2
	Homepage	USN-5	As a user, I can view the contact details and required information.	Based on user requirements,Contents are categorized.	High	Sprint-3
Customer Care Executive	Help	USN-6	As a user, I could contact the site owner if I faced any issues.	Report issues option will be provided.	High	Sprint-3

Table 5.3

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story/Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect the suitable dataset for predicting the chronic kidney disease.	10	High	Kaviya.N
Sprint-1	Data Pre-Processing	USN-2	Datasets are transformed into useful format.	7	Medium	Kaviya.N
Sprint-2	Model Building	USN-3	Calculate the Index values	10	High	Abirami.V
Sprint-2		USN-4	Splitting the Model into Training and Testing from the overall dataset.	7	Medium	Abirami.V
Sprint-3	Training and Testing	USN-5	Train the Model using Regression algorithm and testing the performance of the model.	10	High	Bhava Dharani.G
Sprint-3	Application Building	USN-6	Build the HTML and python code	7	Medium	Bhava Dharani.G
Sprint-4		USN-7	Run Flask App	10	High	Pradeep.M .M
Sprint-4	Implementation of the Application	USN-8	Deploy the model on IBM cloud.	7	Medium	Pradeep.M .M

Fig 6.1

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	8	29 Oct 2022
Sprint-2	10	6 Days	31 Oct 2022	05 Nov 2022	7	05 Nov 2022
Sprint-3	10	6 Days	06 Oct 2022	12 Nov 2022	8	12 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	7	19 Nov 2022

Fig 6.2

VELOCITY:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day.

$$AV = \frac{\text{sprint duration}}{\text{velocity}} = 6/10=0.6$$

6.2 PROJECT DELIVERY SCHEDULE

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gatheringinformation by referring the technical papers, research publications, journals etc.	1 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains and Gains, prepare list of problem Statements that are to be solved by this project.	7 SEPTEMBER 2022 & 9 SEPTEMBER 2022
Ideation	List the ideas by organizing abrainstorming session and prioritize the top three ideas based on the feasibility and importance.	15 SEPTEMBER 2022
ProposedSolution	Prepare the proposed solutiondocument, which includes novelty, feasibility of idea, revenue model, social impact, scalability of solution, etc.	22 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 SEPTEMBER 2022
SolutionArchitecture	Prepare solution architecturedocument.	30 SEPTEMBER 2022
CustomerJourney	Prepare the customer journeymaps to understand the user interactions and experiences with the application (entry to exit).	6 OCTOBER2022
Functional Requirement	Prepare the functional requirement document.	11 OCTOBER 2022

Data FlowDiagrams	Draw the data flow diagrams and submit for review.	11 OCTOBER 2022
TechnologyArchitecture	Prepare the technology architecture diagram.	14 OCTOBER 2022
Prepare Milestone &Activity List	Prepare the milestones and activity list of the project.	21 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 &4	Develop and submit the developed code by testing it.	IN PROGRESS...

Fig 6.2

6.3 REPORTS FROM JIRA

BURNDOWN CHART

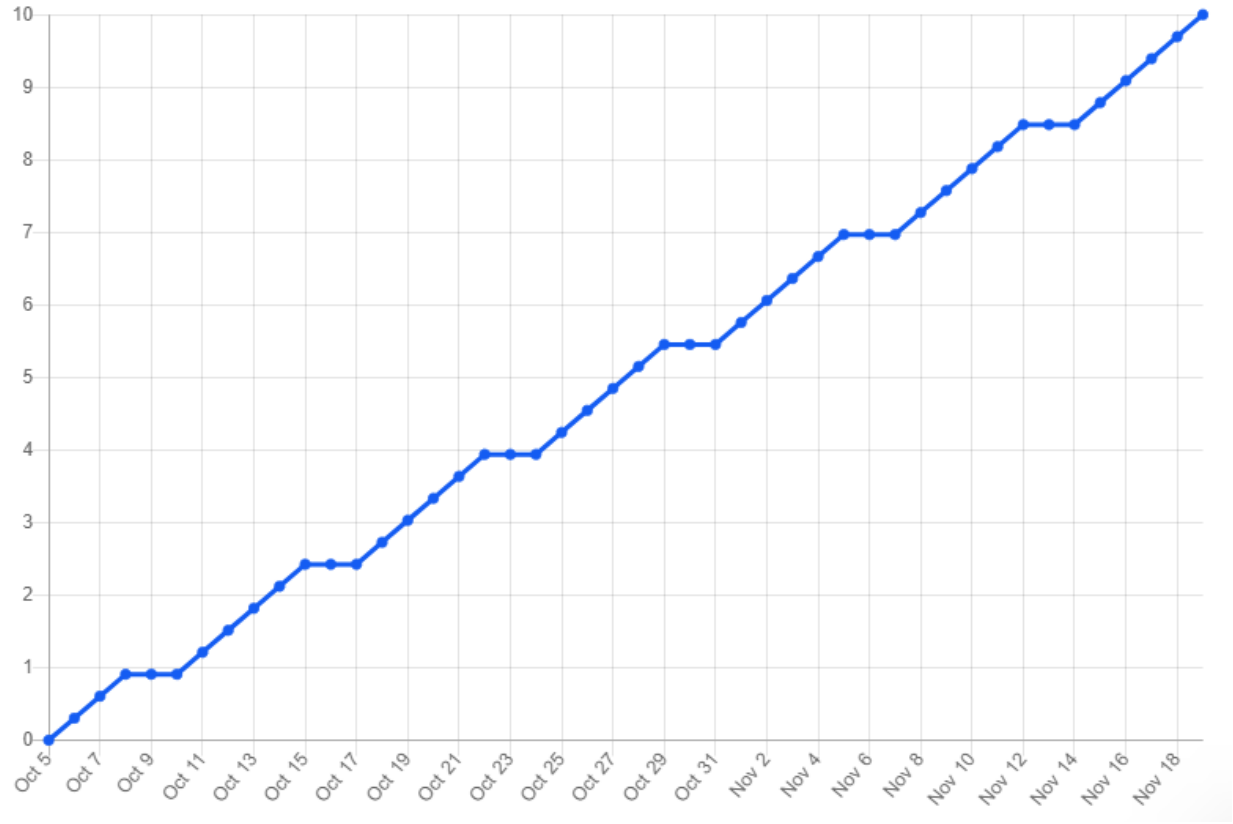

Fig 6.3.1

BURNUP CHART

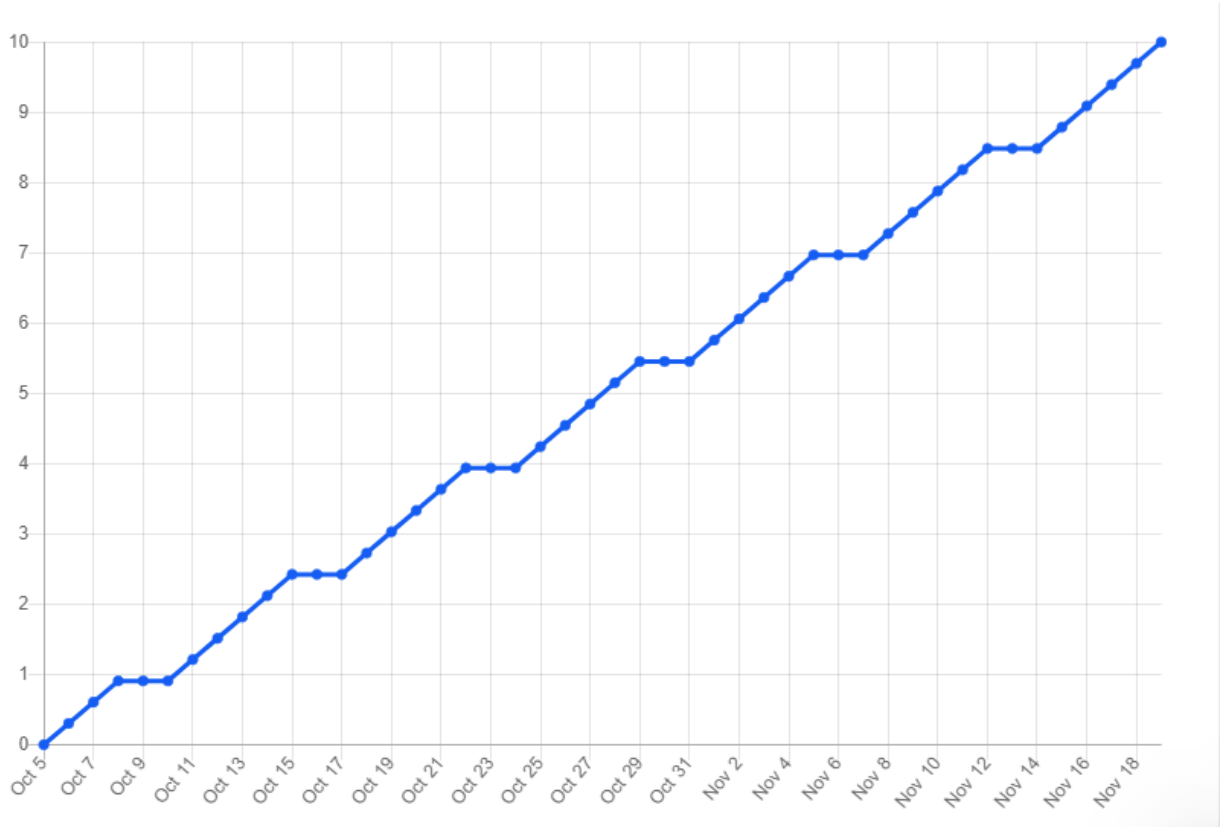

Fig 6.3.2

7.CODING AND SOLUTIONING

7.1 FEATURE 1(RANDOM FOREST ALGORITHM MODEL)

Random Forest Classifier is used to train and test the model for detecting the Chronic Kidney Disease (CKD) with the help of collected and pre-processed dataset collections. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Moreover, NumPy forms the foundation of the Machine Learning stack. Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. Sea born is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.EDA is applied to investigate the data and summarize the key insights. It will give you the basic understanding of your data, it is distribution, null values and much more. You can either explore data using graphs or through some python functions. There will be two types of analysis. Descriptive statistics are brief informational coefficients that summarize a given data set, which can be either a representation of the entire population or a sample of a population. Descriptive statistics are broken down into measures of central tendency and measures of variability. Measures of central tendency include the mean, median, and mode, while measures of variability include standard deviation, variance, minimum and maximum variables, kurtosis, and Skewness. Label Encoding refers to converting the labels into a numeric form to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning. “Pickling” is the process whereby a Python object

hierarchy is converted into a byte stream, and “unpickling” is the inverse operation, whereby a byte stream is converted back into an object hierarchy. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements machine learning algorithms under the Gradient Boosting framework.

7.2 FEATURE 2(FLASK CONNECTIVITY)

The framework is the basis upon which software programs are built. It serves as a foundation for software developers, allowing them to create a variety of applications for certain platforms. It is a set of functions and predefined classes used to connect with the system software and handle inputs and outputs. It simplifies the life of a developer while giving them the ability to use certain extensions and makes the online applications scalable and maintainable. Flask is a web application framework written in Python. A Web Application Framework or a simply a Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, among other examples. Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. Both are Pocco projects. The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications. Flask is often referred to as a micro-framework. It is designed to keep the core of the application simple and scalable. Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application. Unlike the Django framework, Flask is very Pythonic. It’s easy to get started with Flask, because it doesn’t have a huge learning curve. HTML stands for Hyper Text Markup Language. HTML is the standard markup language for creating Web pages. HTML describes the structure of a Web page. HTML consists of a series of elements. HTML elements tell the browser how to display the content. Flask is used for

developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are: There is a built-in development server and a fast debugger provided. The model deployed using Flask is used to predict the Chronic Kidney Disease. Hypertext markup language (HTML) is the basic language used to create documents for the Web and, along with HTTP (hypertext transfer protocol) and URLs (universal resource locators), is one of the three main protocols of the Web. Hypertext is text that contains hyperlinks. A hyperlink is an automated cross-reference to another location on the same document or to another document which, when selected by a user, causes the computer to display the linked location or document within a concise period. A markup language is a set of tags that can be embedded in digital text to provide additional information about it, including its content, structure and appearance. This information facilitates automated operations on the text, including formatting it for display, searching it and even modifying it. Some type of markup language is employed by every word processing program and by nearly every other program that displays text, although such languages and their tags are typically hidden from the user.HTML consists of a set of predefined tags that can be embedded in text by web site designers in order to indicate the details of how web pages are rendered (i.e., converted into a final, easily usable, form) by web browsers. These details include paragraphing, margins, fonts (including style and size), columns, colors (background and text), links, the location of images, text flow around images, tables, and user input form elements (such as spaces for adding text and submit buttons).

7.3 DATABASE SCHEMA

In the recent decades, the evolution of omics technologies has led to advances in all biological fields, creating a demand for effective storage, management and exchange of rapidly generated data and research discoveries. To address this need, the development of databases of experimental outputs has become a common part of scientific practice in order to serve as knowledge sources and data-sharing platforms, providing information about genes, transcripts, proteins or metabolites. In this review, we present omics databases available currently, with a special focus on their application in kidney research and possibly in clinical practice. Databases are divided into two categories: general databases with a broad information scope and kidney-specific databases distinctively concentrated on kidney pathologies. In research, databases can be used as a rich source of information about pathophysiological mechanisms and molecular targets. In the future, databases will support clinicians with their decisions, providing better and faster diagnoses and setting the direction towards more preventive, personalized medicine. We also provide a test case demonstrating the potential of biological databases in comparing multi-omics datasets and generating new hypotheses to answer a critical and common diagnostic problem in nephrology practice. In the future, employment of databases combined with data integration and data mining should provide powerful insights into unlocking the mysteries of kidney disease, leading to a potential impact on pharmacological intervention and therapeutic disease management.

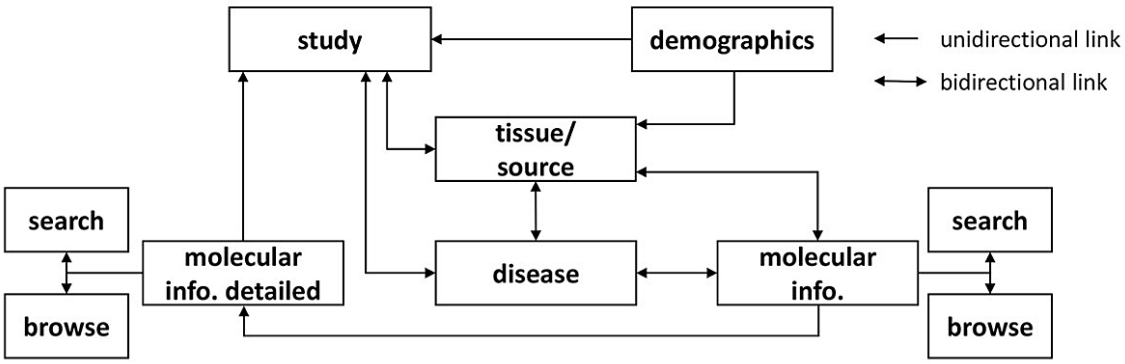
study
demographics
unidirectional link
bidirectional link
tissue/
source
search
molecular
info. detailed
disease
molecular
info.
search
browse
browse

Fig 7.3

8.TESTING

8.1 TEST CASES

Test Case ID	15455	**Test Case Description**	Test the Chronic Kidney Disease Prediction Functionality		
Created By	PRADEEP M M	**Reviewed By**	BHAVA DHARANI G		
Tester's Name	KAVIYA N	**Date Tested**	November 11, 2022	**Test Case (Pass/Fail/ Not Executed)**	Pass
	ABIRAMI V				

S #	Prerequisites:
1	Access to Chrome Browser
2	Entering the details required
3	check for correct values
4	Application to train the model

S #	Test Data
1	By Clicking the website link
2	Details should be in a integer format
3	Data should be filled
4	Provide the datasets for model training

Test Scenario Verify whether the deployed project predicts as per expected

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Navigate to corresponding website link	Site should open	As Expected	Pass
2	Enter the details	Details should be entered	As Expected	Pass
3	Click Submit	Check the result	As Expected	Pass
4	Output results	Results are generated	As Expected	Pass

Fig 8.1

8.2 USER ACCEPTANCE TESTING

9.RESULTS

9.1 PERFORMANCE METRICS

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gatheringinformation by referring thetechnical papers, research publications, journals etc.	1 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, prepare list of problem Statements that are to besolvedby this project.	7 SEPTEMBER 2022 & 9 SEPTEMBER 2022
Ideation	List the ideas by organizing a brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	15 SEPTEMBER 2022
ProposedSolution	Preparethe proposed solutiondocument, which includes novelty, feasibility of idea, revenue model, social impact, scalability of solution, etc.	22 SEPTEMBER 2022
Problem Solution Fit	Prepareproblem - solution fitdocument.	30 SEPTEMBER 2022
SolutionArchitecture	Prepare solution architecturedocument.	30 SEPTEMBER 2022

CustomerJourney	Preparethe customer journey maps to understand the userinteractions & experiences with the application (entry to exit).	6 OCTOBER 2022
FunctionalRequirement	Prepare the functionalrequirement document.	11 OCTOBER 2022
Data FlowDiagrams	Draw the data flow diagrams and submitfor review.	11 OCTOBER 2022
TechnologyArchitecture	Prepare the technologyarchitectu re diagram.	14 OCTOBER 2022
Prepare Milestone &ActivityList	Preparethe milestones &activity list of the project.	21 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 &4	Develop & submit the developed code by testing it.	IN PROGRESS...

Table 9.1

10.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

The early detection of CKD allows patients to receive timely treatment, slowing the disease's progression. Due to its rapid recognition performance and accuracy, machine learning models can effectively assist physicians in achieving this goal.Chronic kidney disease (CKD) is a type of kidney disease in which there is gradual loss of kidney function over a period of months to years, Initially, there are generally no symptoms; later, symptoms may include leg swelling, feeling tired, vomiting, loss of appetite, and confusion. Complications include an increased risk of heart disease, high blood pressure, bone disease, and anemia. CKD is associated with a decrease in kidney function related to age and is accelerated in hypertension, diabetes, obesity, and primary kidney disorders. CKD is a global health problem with a high morbidity and mortality rate, and it induces other diseases. As there are no obvious symptoms during the early stages of CKD, patients often do not notice the disease, this being the main feature, eventually leading to a complete loss of kidney function. Early detection of CKD allows patients to receive timely treatment to improve the progression of this disease. As it has been proposed in the objectives of the work, the aim is to develop an automatic learning model for the prediction in the diagnosis of CKD and to contribute to the reduction of significant complications in the disease such as dialysis processes, kidney transplantation, or reaching death. The main criterion of success for this project, with the help of machine learning, is to identify the behaviors or behavior patterns in the initial stages of CKD to improve the quality of life of patients.

DISADVANTAGES:

The idea for the approach of this project arises from the current situation regarding the increase in the confirmatory diagnosis of CKD, and lack of treatment or the user's ignorance of its pathologies leads to irreversible kidney failure in the final stages of CKD, such as dialysis for life, financially affecting the health system, as it is a costly treatment that generates the most significant amount of absorption of the resources available for health in Iraq. This could be reduced by using tools such as machine learning to classify ERC from the initial stages. Although the application of machine learning in healthcare and other areas is favorable, the field of kidney disease has not yet exploited its full potential.

11.CONCLUSION

Chronic Kidney Disease (CKD) or chronic renal disease has become a major issue with a steady growth rate. A person can only survive without kidneys for an average time of 18 days, which makes a huge demand for a kidney transplant and Dialysis. It is important to have effective methods for early prediction of CKD.Machine learning methods are effective in CKD prediction. This work proposes a workflow to predict CKD status based on clinical data, incorporating data prepossessing, a missing value handling method with collaborative filtering and attributes selection.Out of the 11 machine learning methods considered, the extra tree classifier and random forest classifier are shown to result in the highest accuracy and minimal bias to the attributes. The research also considers the practical aspects of data collection and highlights the importance of incorporating domain knowledge when using machine learning for CKD status prediction.

12.FUTURE SCOPE

The increasing rate of CKD has placed a large negative impact on individuals' lives. Advanced ML technology has made the early detection of CKD easier and more accurate. Doctors and medical care professionals have used ML algorithms in the effective diagnosis of CKD. However, there is very little research on the detection of secondary infections of prolonged CKD such as albuminuria and toxin production through the ML algorithm. These secondary infections also place a negative impact, especially on diabetics and patients with high blood pressure. Therefore,Determination of the role of ML algorithms in detecting CKD-associated diseases can be an effective research. Further research on effective treatment prediction and nutritional chart prediction of CKD patients through ML algorithm needs to be done in the future. Advanced technologies such as CNN, ML, random forest, and different classifiers can be used for these aspects to increase the recovery rate in CKD. By following this way, researchers and medical care professionals can enhance their service quality in accurate CKD diagnosis and treatment. Effective detection of CKD through ML algorithm is rapid and cost-effective, and due to this reason, the method can gain large popularity in the future.

13.APPENDIX

Source Code:

Random Forest Alogirhtm:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#from python.nominal import associations
import warnings
warnings.filterwarnings('ignore')
from google.colab import drive
drive.mount('/content/drive')
df=pd.read_csv(r"/content/chronickidneydisease.csv")
df
df.duplicated().sum()
df.info()
def convert_dtype(df,feature):
    df[feature]=pd.to_numeric(df[feature],errors='coerce')
#whereever we have Nan values , this errors parameter will hanfle that
features=['pcv','wc','rc']
for i in features:
    convert_dtype(df,i)
plt.subplot(1,2,1)
sns.boxplot(x=df['classification'],y=df['age'])
def extract_cat_num(kidney):
    cat_col=[col for col in kidney.columns if kidney[col].dtype=='O']
    num_col=[col for col in kidney.columns if kidney[col].dtype!='O']
    return cat_col,num_col
cat_col,num_col=extract_cat_num(df)
len(num_col)
plt.figure(figsize=(30,30))
for i,feature in enumerate(num_col):
    plt.subplot(5,3,i+1)
    df[feature].hist()
    plt.title(feature)
```

```
len(cat_col)
plt.figure(figsize=(20,20))
for i,feature in enumerate(cat_col):
    plt.subplot(4,3,i+1)
    sns.countplot(df[feature])

df.groupby(['rbc','classification'])['rc'].agg(['count','mean','median','min','max'])

plt.figure(figsize=(10,10))
plt.scatter(x=df.hemo,y=df['pcv'],color="red")
plt.xlabel('Hemo')
plt.ylabel('pcv')
plt.title('Relationship between haemoglobin and packed cell volume')

Text(0.5, 1.0, 'Relationship between haemoglobin and packed cell volume')

grid=sns.FacetGrid(df,hue='classification',aspect=2)

grid.map(sns.kdeplot,'rc')

grid.add_legend()

plt.figure(figsize=(12,10))

sns.scatterplot(x=df['rc'],y=df['hemo'],hue=df['classification'])

plt.xlabel('rc')

plt.ylabel('hemo')

plt.title('Relationship between haemoglobin and red blood cell count')

Text(0.5, 1.0, 'Relationship between haemoglobin and red blood cell count')

from dython.nominal import identify_nominal_columns

categorical_features=identify_nominal_columns(df)

categorical_features

complete_correlation= associations(df, filename= 'complete_correlation')

df_complete_corr=complete_correlation['corr']

df_complete_corr.dropna(axis=1, how='all').dropna(axis=0, how='all')

df.corr().style.background_gradient(cmap="nipy_spectral_r")

df=df.drop(["id"],axis=1)
```

```
df=df.drop(["age"],axis=1)
df=df.drop(["wc"],axis=1)
df.columns
df.describe()
missing_values=df.columns[df.isnull().any()]
df[missing_values].isnull().sum()
labels = []
valuecount = []
percentcount = []
for col in missing_values:
   labels.append(col)
   valuecount.append(df[col].isnull().sum())
   percentcount.append(df[col].isnull().sum()/df.shape[0])
ind = np.arange(len(labels))
fig, (ax1, ax2) = plt.subplots(1,2,figsize=(10,5))
rects = ax1.barh(ind, np.array(valuecount), color='yellow')
ax1.set_yticks(ind)
ax1.set_yticklabels(labels, rotation='horizontal')
ax1.set_xlabel("Count of missing values")
ax1.set_title("Variables with missing values");
rects = ax2.barh(ind, np.array(percentcount), color='green')
ax2.set_yticks(ind)
ax2.set_yticklabels(labels, rotation='horizontal')
ax2.set_xlabel("Percentage of missing values")
ax2.set_title("Variables with missing values");
print("Total count of missing value in a dataset:",df.isnull().sum().
```

```
df['bp'] = df['bp'].fillna(df['bp'].mean())
df["bp"].isnull().sum()
df['sg'] = df['sg'].fillna(df['sg'].mean())
df["sg"].isnull().sum()
df['al'] = df['al'].fillna(df['al'].mean())
df["al"].isnull().sum()
df['su'] = df['su'].fillna(df['su'].mean())
df["su"].isnull().sum()
df['rbc'] = df['rbc'].fillna("not mentioned")
df["rbc"].isnull().sum()
df['pc'] = df['pc'].fillna(df['pc'].mode()[0])
df["pc"].isnull().sum()
df = df.dropna(axis=0, subset=['pcc',"htn","appet"])
df['bgr'] = df['al'].fillna(df['al'].mean())
df["bgr"].isnull().sum()
df['bu'] = df['bu'].fillna(df['bu'].mean())
df["bu"].isnull().sum()
df['sc'] = df['sc'].fillna(df['sc'].mean())
df['sod'] = df['sod'].fillna(df['sod'].mean())
df['pot'] = df['pot'].fillna(df['pot'].mean())
df['pcv'] = df['pcv'].fillna(df['pcv'].mode()[0])
df['hemo'] = df['hemo'].fillna(df['hemo'].mean())
df['rc'] = df['rc'].fillna("not mentioned")
df["rc"].isnull().sum()
df.info()
df.isnull().sum()
```

```
for column in df:
    if (df[column].dtype=="object"):
       print("\n",column)
       print(df[column].value_counts(),"\n")
or column in df:
    if (df[column].dtype=="object"):
       print("\n",column)
       print(df[column].value_counts(),"\n")
objList = df.select_dtypes(include = "object").columns
print (objList)
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()

for feat in objList:
   df[feat] = le.fit_transform(df[feat].astype(str))
print (df.info())
df.dtypes
X = df.drop(['classification', 'sg',  'rc', 'pcv'], axis = 1)
y = df['classification']
X.columns
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators = 20)
model.fit(X_train, y_train)
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
confusion_matrix(y_test, model.predict(X_test))
print(f"Accuracy is {round(accuracy_score(y_test, model.predict(X_test))*100, 2)}%")
mport pickle
pickle.dump(model, open('kidney.pkl', 'wb'))
from xgboost import XGBClassifier
params={'learning-rate':[0,0.5,0.20,0.25],
      'max_depth':[5,8,10],
     'min_child_weight':[1,3,5,7],
     'gamma':[0.0,0.1,0.2,0.4],
     'colsample_bytree':[0.3,0.4,0.7]}
from sklearn.model_selection import RandomizedSearchCV
classifier=XGBClassifier()
random_search.best_params_
classifier=XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
          colsample_bynode=1, colsample_bytree=0.3, gamma=0.2, gpu_id=-1,
          importance_type='gain', interaction_constraints='', learning_rate=0.300000012,
max_delta_step=0,
          max_depth=5, min_child_weight=1,
          monotone_constraints='()', n_estimators=100, n_jobs=8,
          num_parallel_tree=1, random_state=0, reg_alpha=0, reg_lambda=1,
          scale_pos_weight=1, subsample=1, tree_method='exact',
          validate_parameters=1, verbosity=None)
y_pred=classifier.predict(X_test)
random_search=RandomizedSearchCV(classifier,param_distributions=params,n_iter=5,
scoring='r')
random_search.fit(X_train,y_train)
random_search.best_estimator_    #Checking for best model
```

```
from sklearn.metrics import confusion_matrix,accuracy_score
confusion_matrix(y_test,y_pred)
array([[54,  0],
       [ 0, 23]], dtype=int64)
accuracy_score(y_test,y_pred)
from sklearn.ensemble import RandomForestRegressor
reg=RandomForestRegressor()
reg.fit(X_train,y_train)
X_test.columns
y_pred=reg.predict(X_test)
pickle. dump(reg, open('randomreg_chronic', 'wb'))
y_pred
l_pred=list(y_pred)
l_test=list(y_test)
d={'prob':l_pred,'out':y_test}
df_i=pd.DataFrame(d)
df_i.head(20)
df_i.to_csv(r"/content/chronickidneydisease.csv")
```

HTML:

index.html

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
<title>CKD</title>
<style>
body {
```

```
background: linear-gradient(
                rgba(10,10,10, .35),
                rgba(10,10,10, .105)),
                url("https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRaHsWxQQ9G7o2Lf-uSvMiAMMmdd2gWwZxzJQ&usqp=CAU.JPG");
    background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
.container {
  border: 2px solid #ccc;
  padding: 10px;
  width: 20em;
height:21em;
background-color:white;
}
.hello{
opacity: 0.5;
}
</style>
</head>
<body>
<marquee><p style="font-size:30px;color:white;"><strong>IBM NALAIYA THIRAN PROJECT</strong></marquee>
<center><p style="font-size:50px;color:WHITE;">CHRONIC KIDNEY DISEASES PREDICTION</p></center>
<form action="/val" method="post"><center>
```

```
  <label for="age"><p style="font-
size:20px;color:black;"><strong>AGE</strong></label><br>
  <input type="number" id="age" name="age"><br><br><br>
 <label for="bp"><strong>BLOOD PRESSURE</strong></label><br>
  <input type="number" id="bp" name="bp">
<br>
<br>
<br>
  <label for="sg"><strong>URINARY SPECIFIC GRAVITY(SG)</strong></label><br>
  <input type="number" id="sg" name="sg">
<br>
<br>
<br>
  <label for="al"><strong>AL</strong></label><br>
  <input type="number" id="al" name="al">
<br>
<br>
<br>
  <label for="su"><strong>SU</strong></label><br>
  <input type="number" id="su" name="su">
<br>
<br>
<br>
  <label for="rbc"><strong>RBC</strong></label><br>
  <input type="text" id="rbc" name="rbc">
<br>
<br>
```

```
<br>
  <label for="pc"><strong>PC</strong></label><br>
  <input type="text" id="pc" name="pc">
<br>
<br>
<br>
  <label for="pcc"><strong>PCC</strong></label><br>
  <input type="text" id="pcc" name="pcc">
<br>
<br>
<br>
  <label for="ba"><strong>BA</strong></label><br>
  <input type="text" id="ba" name="ba">
<br>
<br>
<br>
  <label for="bgr"><strong>BGR</strong></label><br>
  <input type="number" id="bgr" name="bgr">
<br>
<br>
<br>
  <label for="bu"><strong>BU</strong></label><br>
  <input type="number" id="bu" name="bu">
<br>
<br>
<br>
```

```
  <label for="sc"><strong>SC</strong></label><br>
  <input type="number" id="sc" name="sc">
<br>
<br><br>
  <label for="sod">SOD</label><br>
  <input type="number" id="sod" name="sod">
<br>
<br>
<br>
  <label for="pot"><strong>POT</strong></label><br>
  <input type="number" id="pot" name="pot">
<br>
<br>
<br>
  <label for="hemo"><strong>HEMO</strong></label><br>
  <input type="number" id="hemo" name="hemo">
<br>
<br>
<br>
  <label for="pcv"><strong>PCV</strong></label><br>
  <input type="text" id="pcv" name="pcv">
<br>
<br>
<br>
  <label for="wc"><strong>WC</strong></label><br>
  <input type="text" id="wc" name="wc">
```

```
<br>
<br>
<br>
  <label for="rc"><strong>RC</strong></label><br>
  <input type="number" id="rc" name="rc">
<br>
<br>
<br>
  <label for="htn"><strong>HTN</strong></label><br>
  <input type="text" id="htn" name="htn">
<br>
<br>
<br>
  <label for="dm"><strong>DM</strong></label><br>
  <input type="text" id="dm" name="dm">
<br>
<br>
<br>
  <label for="cad"><strong>CAD</strong></label><br>
  <input type="text" id="cad" name="cad">
<br>
<br>
<br>
  <label for="appet"><strong>APPET</strong></label><br>
  <input type="text" id="appet" name="appet">
<br>
```

```
<br>
<br>
  <label for="pe"><strong>PE</strong></label><br>
  <input type="text" id="pe" name="pe">
<br>
<br>
<br>
  <label for="ane"><strong>ANE</strong></label><br>
  <input type="text" id="ane" name="ane">
<br>
<br></center>
  <center><strong><button type="submit">CHECK</button></center></strong>
</form>
</body>
</html>
```

rename.html

```
<html>
<head>
<style>
body {
  background-color:"red";
}
</style>
</head>
<body>
<br>
<br>
<br>
<center><h1>{{answer1}}</h1></center>
<br>
<center><h1>{{answer2}}</h1></center>
<center><img alt="Qries" src="https://c8.alamy.com/comp/2AYXWRH/sad-unhealthy-ill-kidney-character-hold-in-hand-fast-food-soda-beverage-bottle-and-burger-human-anatomy-genitourinary-system-internal-sick-organ-vector-bad-nutrition-addiction-cartoon-illustration-2AYXWRH.jpg"
        width="500" height="500"></center>


</body>
</html>
```

rename2.html

```
<html>
<head>
<style>
body {
  background-color: #E6E6FA;
}
</style>
</head>
<body >
<br>
<br>
<br>
<center><h1>{{answer1}}</h1></center>
<br>
<center><h1>{{answer2}}</h1></center>
<center><img alt="Qries" src="https://thumbs.dreamstime.com/z/happy-cute-smiling-healthy-kidney-vector-happy-cute-smiling-healthy-kidney-vector-modern-style-cartoon-character-illustration-icon-100355932.jpg"
        width="500" height="500"></center>
</body>
</html>
```

FLASK CONNECTIVITY:

```
import pickle
loaded_class = pickle.load(open('randomclass_chronic', 'rb'))
loaded_reg = pickle.load(open('randomreg_chronic', 'rb'))
import numpy as np
import pandas as pd
import flask
from flask import Flask, request, redirect, render_template
app = Flask(__name__)
@app.route("/",methods=['GET', 'POST'])
def index():
    return render_template('index.html')
@app.route("/val",methods=['POST'])

def val():
    test=[]
    if request.method == 'POST':
        test.append(request.form.get("age"))
        test.append(request.form.get("bp"))
        test.append(request.form.get("sg"))
        test.append(request.form.get("al"))
        test.append(request.form.get("su"))
        rb=request.form.get("rbc")
        if rb=='abnormal':
            test.append(1)
        else:
```

```
    test.append(0)
pc=request.form.get("pc")
if pc=='abnormal':
    test.append(1)
else:
    test.append(0)
pcc=request.form.get("pcc")
if pcc=='present':
    test.append(1)
else:
    test.append(0)
ba=request.form.get("ba")
if ba=='present':
    test.append(1)
else:
    test.append(0)
test.append(request.form.get("bgr"))
test.append(request.form.get("bu"))
test.append(request.form.get("sc"))
test.append(request.form.get("sod"))
test.append(request.form.get("pot"))
test.append(request.form.get("hemo"))
test.append(request.form.get("pcv"))
test.append(request.form.get("wc"))
test.append(request.form.get("rc"))
ht=request.form.get("htn")
```

```
if ht=='yes':
    test.append(1)
else:
    test.append(0)
d=request.form.get("dm")
if d=='yes':
    test.append(1)
else:
    test.append(0)
ca=request.form.get("cad")
if ca=='yes':
    test.append(1)
else:
    test.append(0)
ap=request.form.get("appet")
if ap=='good':
    test.append(1)
elif ap=='poor':
    test.append(0)
else:
    test.append(np.nan)
p=request.form.get("pe")
if p=='yes':
    test.append(1)
else:
    test.append(0)
```

```
        an=request.form.get("ane")
        if an=='yes':
            test.append(1)
        else:
            test.append(0)
    print(test)
    test_df=pd.DataFrame(test)
    test_df=np.array(test_df).reshape(1, -1)

    ans1=loaded_class.predict(test_df)
    ans2=loaded_reg.predict(test_df)
    if int(ans1)>=0.5:
        answer1="Sorry to say!! You have CHRONIC DISEASE!!!"
        return render_template('rename.html',answer1=answer1,answer2=ans2)
    else:
        answer1="Happy to say that you don't have CHRONIC DISEASE"
        return render_template('rename2.html',answer1=answer1,answer2=ans2)
if __name__ == "__main__":
    app.debug=True
    app.run(debug=False)
```