A PROJECT REPORT ON

EARLY DETECTION OF CHRONIC KIDNEY DISEASE

TEAM ID: PNT2022TMID22283

Bachelor of Engineering
Computer Science and Engineering
Vel Tech Multi Tech Dr.Rangarajan Dr.Sakunthala Engineering College
Avadi, Chennai-600062.

Faculty Evaluator : Mr.Insozhan.P

Faculty Mentor : Mr. Sathish Kumar.P

Team Members :

Avinash.V	(113119UG03012)
EdwinKingsten.A	(113119UG03023)
Manikandan.P	(113119UG03055)
Gokul.D.G	(113119UG03028)

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Chronic Kidney Disease (CKD) is a major medical problem and can be cured if treated in the early stages. Usually, people are not aware that medical tests we take for different purposes could contain valuable information concerning kidney diseases. Consequently, attributes of various medical tests are investigated to distinguish which attributes may contain helpful information about the disease. The information says that it helps us to measure the severity of the problem and we make use of such information to build a machine learning model that predicts Chronic Kidney Disease.

1.2 Purpose

Reduce the burden of chronic kidney disease and related complications. Medicine – to control associated problems, such as high blood pressure and high cholesterol. Dialysis – treatment to replicate some of the kidney's functions, which may be necessary in advanced CKD. Continuous increase in the number of patients with end-stage renal disease demands early detection of chronic kidney disease (CKD). The aim of the present study was to diagnose CKD in its earliest stages in a randomly selected population using a diagnostic algorithm developed by the working group.

2. LITERATURE SURVEY

2.1 Existing Problem

The Existing Chronic kidney disease Detector use two main tests to measure kidney function and stage of kidney disease. The first test is known as eGFR(estimated glomerular filtration rate). First a blood test is done to determine how much creatinine a chemical waste molecule generated by muscle metabolism - is present in the bloodstream. Creatinine levels, as well as factors such as age, gender and body size, are then used to estimate the rate of glomerular filtration. Then Albumin-to-creatinine ratio (ACR) is measured to detect elevated protein. The evaluation of albuminuria is to measure urinary ACR in a spot urine sample . Although the 24-hour collection has been the "gold standard," for detecting protein excretion such as urinary albumin-to-creatinine ratio (ACR) of correct variations in urinary concentration in urine collections.

2.2 References

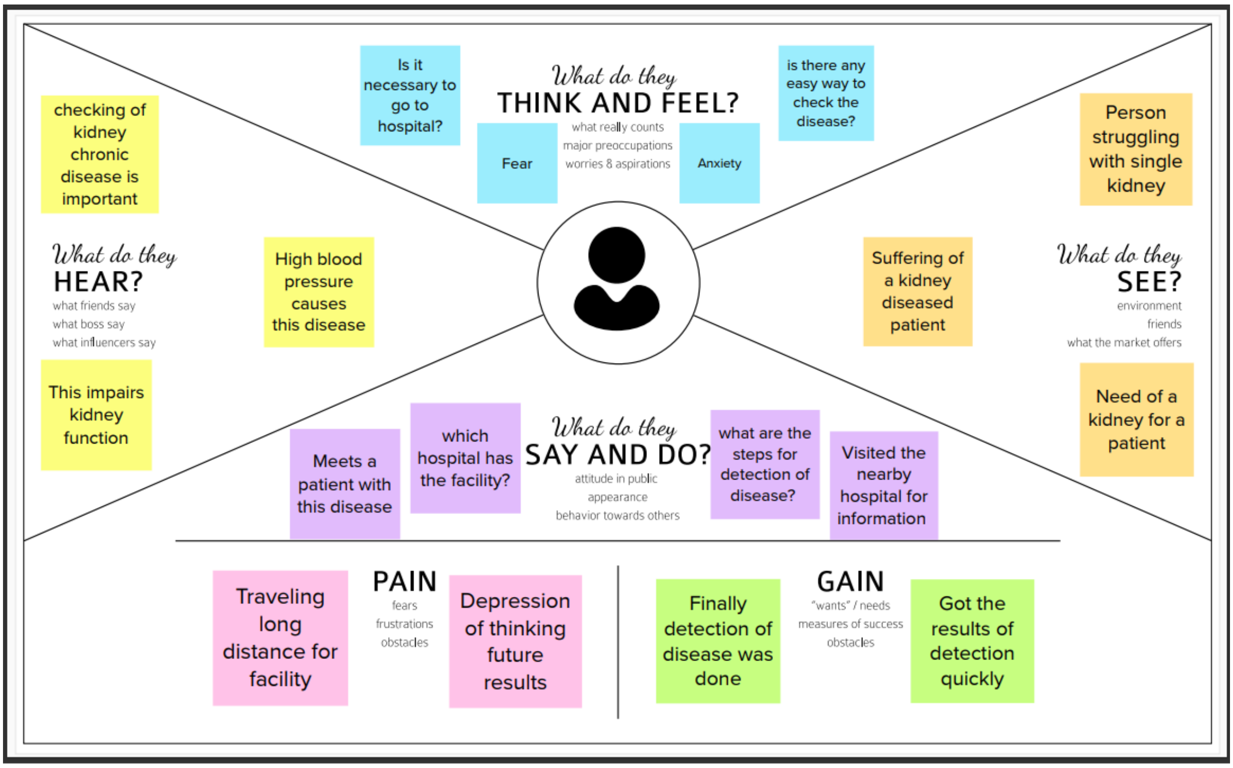
- Devika R, Sai Vaishnavi A, Subramaniyaswamy V *" Comparative Study of Classifier for Chronic Kidney Disease Prediction"* Using Naive Bayes, KNN and Random Forest. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC).
- Kunwar V, Chandel K, Sai Sabitha A, Bansal A "*Chronic Kidney Disease Analysis Using Data Mining Classification Techniques*". 2016 6th International Conference-Cloud System and Big Data Engineering.
- Amirgaliyev Y, Shamiluulu S, Serek A "*Analysis of Chronic Kidney Disease Dataset by Applying Machine Learning Methods*". 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT).

2.3 Problem Statement Definition

Chronic Kidney Disease (CKD) is a serious medical condition that, if caught early enough, is curable. Most individuals are unaware that the various medical tests we undergo for various reasons may provide important information about kidney disorders. As a result, characteristics of numerous medical tests are examined to see which characteristics might contain useful information about the disease. According to the information, doing so enables us to assess the problem's severity, and we utilize this data to create a machine learning model that forecasts chronic kidney disease. If chronic kidney disease is addressed early on, it may be cured. This project's primary goal is to more accurately and quickly identify whether a patient has chronic kidney disease using diagnostic data including Blood Pressure (Bp), Albumin, and other parameters (Al).

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Is it necessary to go to hospital?

Fear

Anxiety

is there any easy way to check the disease?

What do they HEAR?
what friends say
what boss say
what influencers say

checking of kidney chronic disease is important

High blood pressure causes this disease

This impairs kidney function

What do they SEE?
environment
friends
what the market offers

Person struggling with single kidney

Suffering of a kidney diseased patient

Need of a kidney for a patient

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

Meets a patient with this disease

which hospital has the facility?

what are the steps for detection of disease?

Visited the nearby hospital for information

PAIN
fears
frustrations
obstacles

Traveling long distance for facility

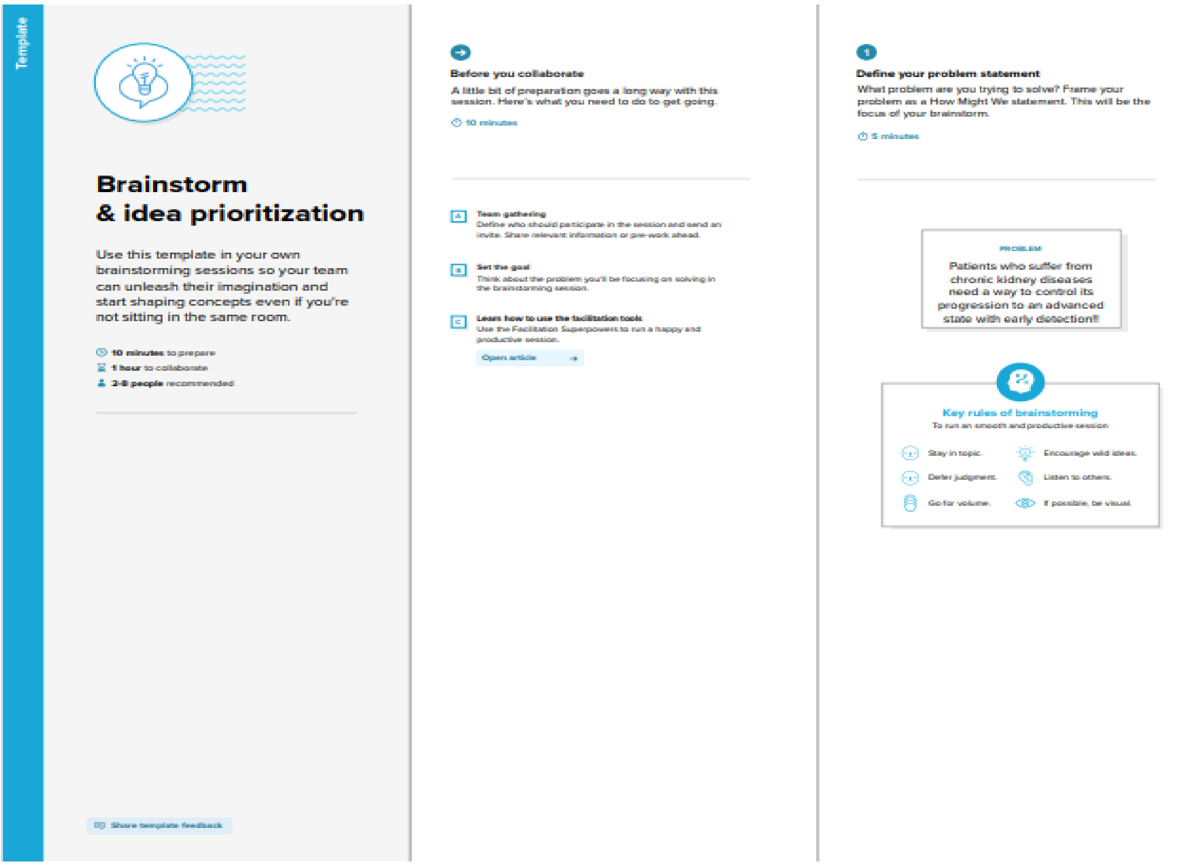
Depression of thinking future results

GAIN
"wants" / needs
measures of success
obstacles

Finally detection of disease was done

Got the results of detection quickly

3.2 Ideation & Brainstorming



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

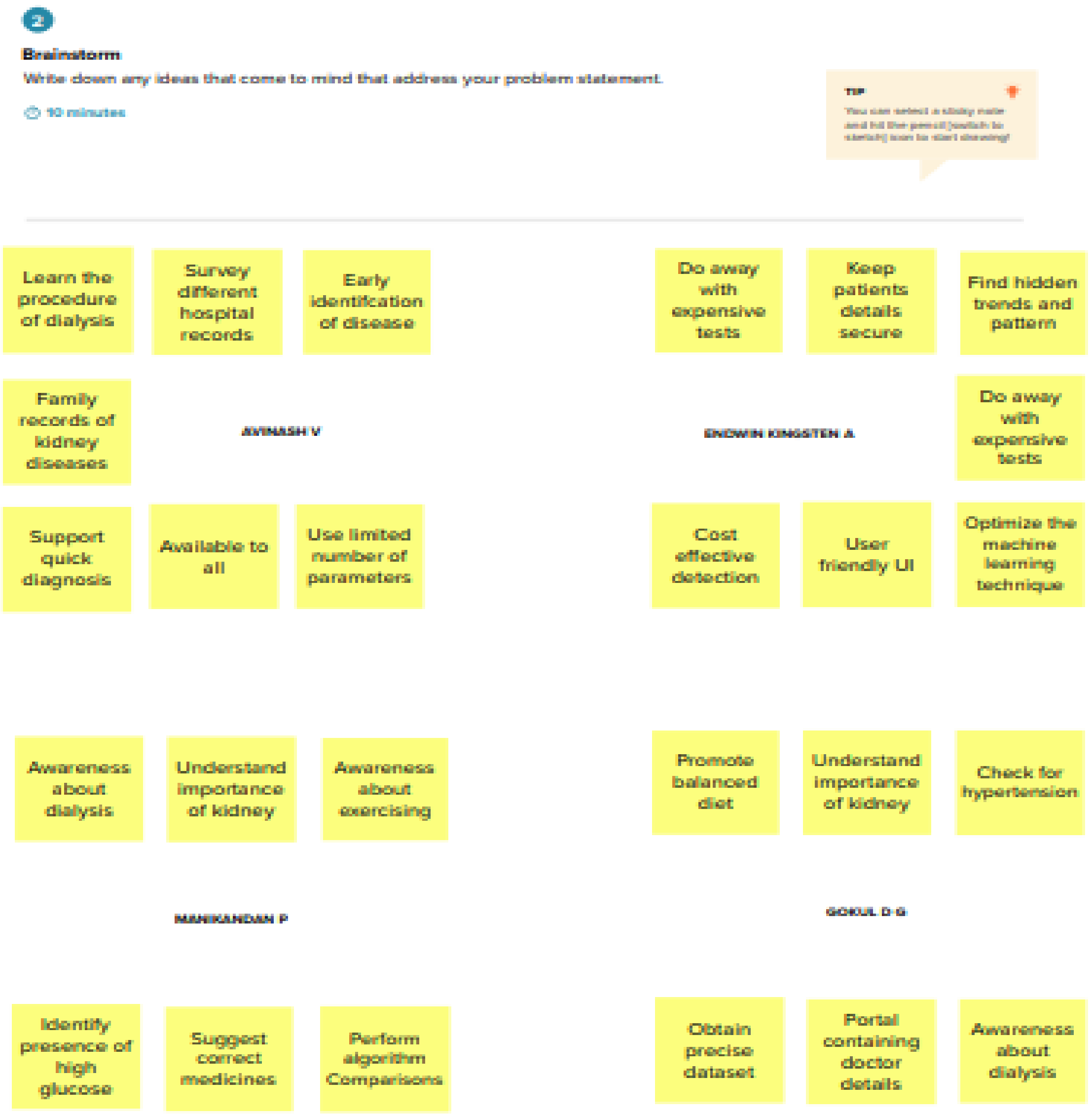
PROBLEM

Patients who suffer from chronic kidney diseases need a way to control its progression to an advanced state with early detection!!

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer judgment.	Listen to others.
Go for volume.	If possible, be visual.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

AVINASH V

- Learn the procedure of dialysis
- Survey different hospital records
- Early identifcation of disease
- Family records of kidney diseases
- Support quick diagnosis
- Available to all
- Use limited number of parameters

ENDWIN KINGSTEN A

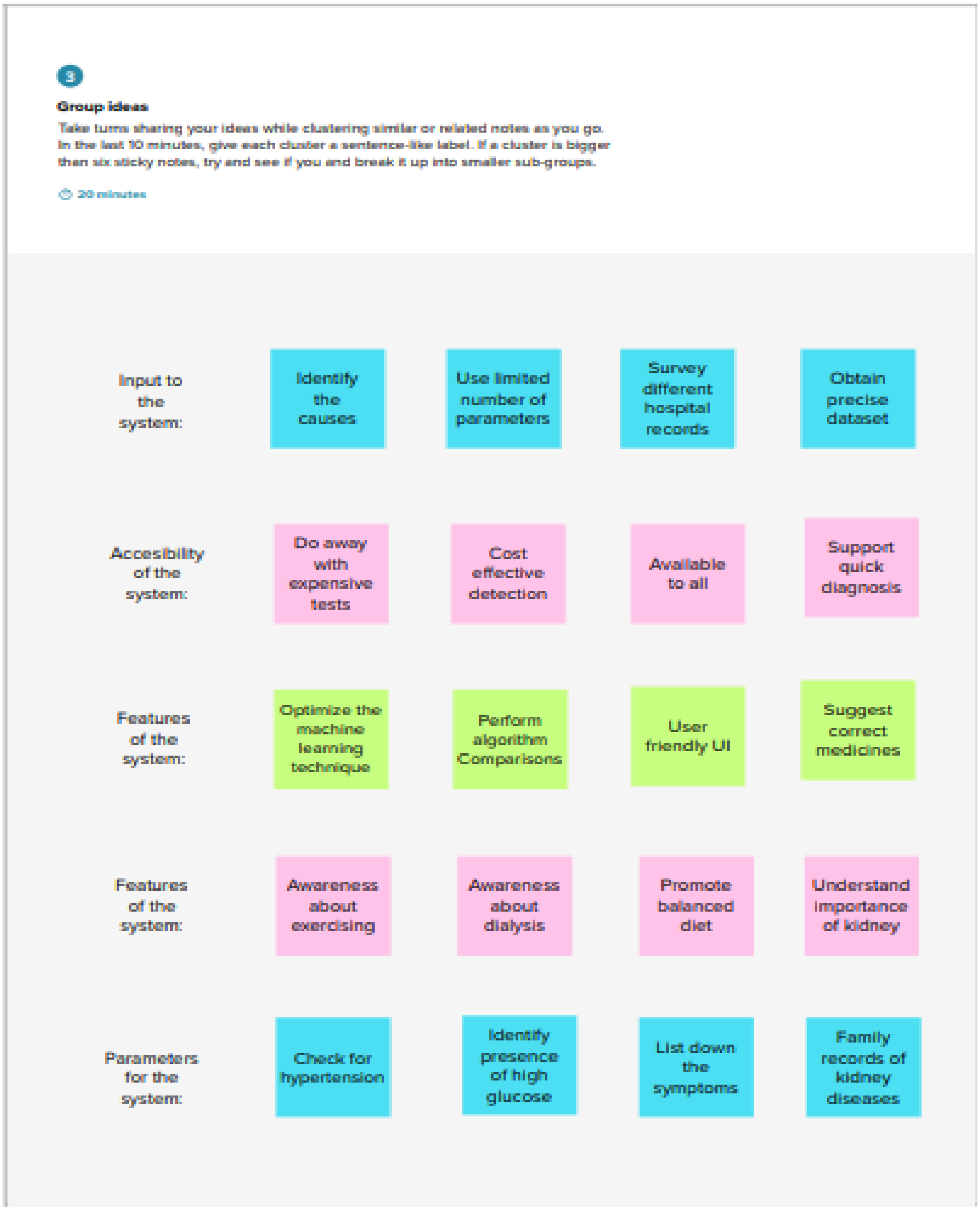
- Do away with expensive tests
- Keep patients details secure
- Find hidden trends and pattern
- Do away with expensive tests
- Cost effective detection
- User friendly UI
- Optimize the machine learning technique

MANIKANDAN P

- Awareness about dialysis
- Understand importance of kidney
- Awareness about exercising
- Identify presence of high glucose
- Suggest correct medicines
- Perform algorithm Comparisons

GOKUL D G

- Promote balanced diet
- Understand importance of kidney
- Check for hypertension
- Obtain precise dataset
- Portal containing doctor details
- Awareness about dialysis



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

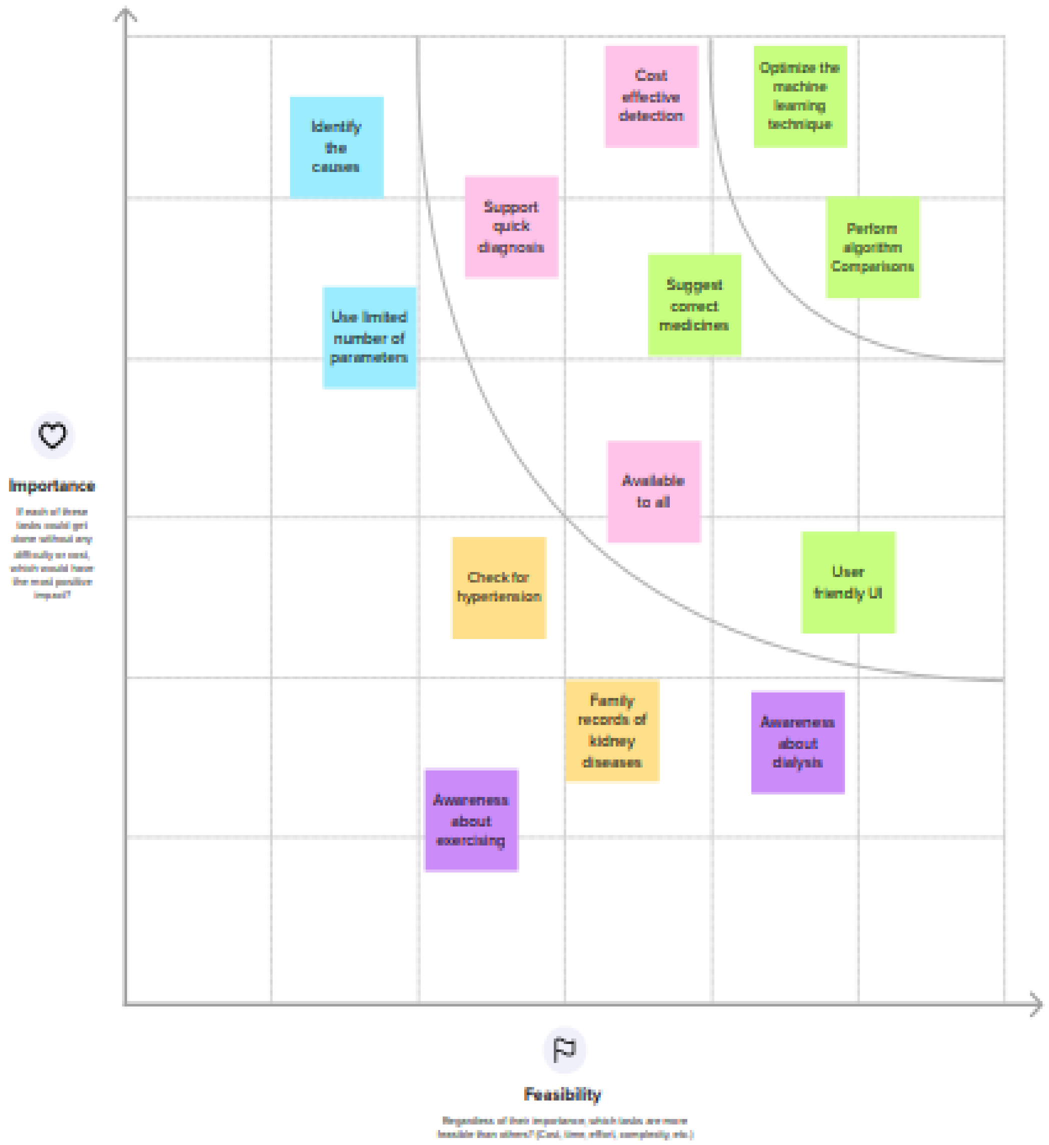
Input to the system:	Identify the causes	Use limited number of parameters	Survey different hospital records	Obtain precise dataset
Accesibility of the system:	Do away with expensive tests	Cost effective detection	Available to all	Support quick diagnosis
Features of the system:	Optimize the machine learning technique	Perform algorithm Comparisons	User friendly UI	Suggest correct medicines
Features of the system:	Awareness about exercising	Awareness about dialysis	Promote balanced diet	Understand importance of kidney
Parameters for the system:	Check for hypertension	Identify presence of high glucose	List down the symptoms	Family records of kidney diseases

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⏱ 20 minutes



Identify the causes

Use limited number of parameters

Support quick diagnosis

Cost effective detection

Suggest correct medicines

Optimize the machine learning technique

Perform algorithm Comparisons

Available to all

Check for hypertension

User friendly UI

Family records of kidney diseases

Awareness about dialysis

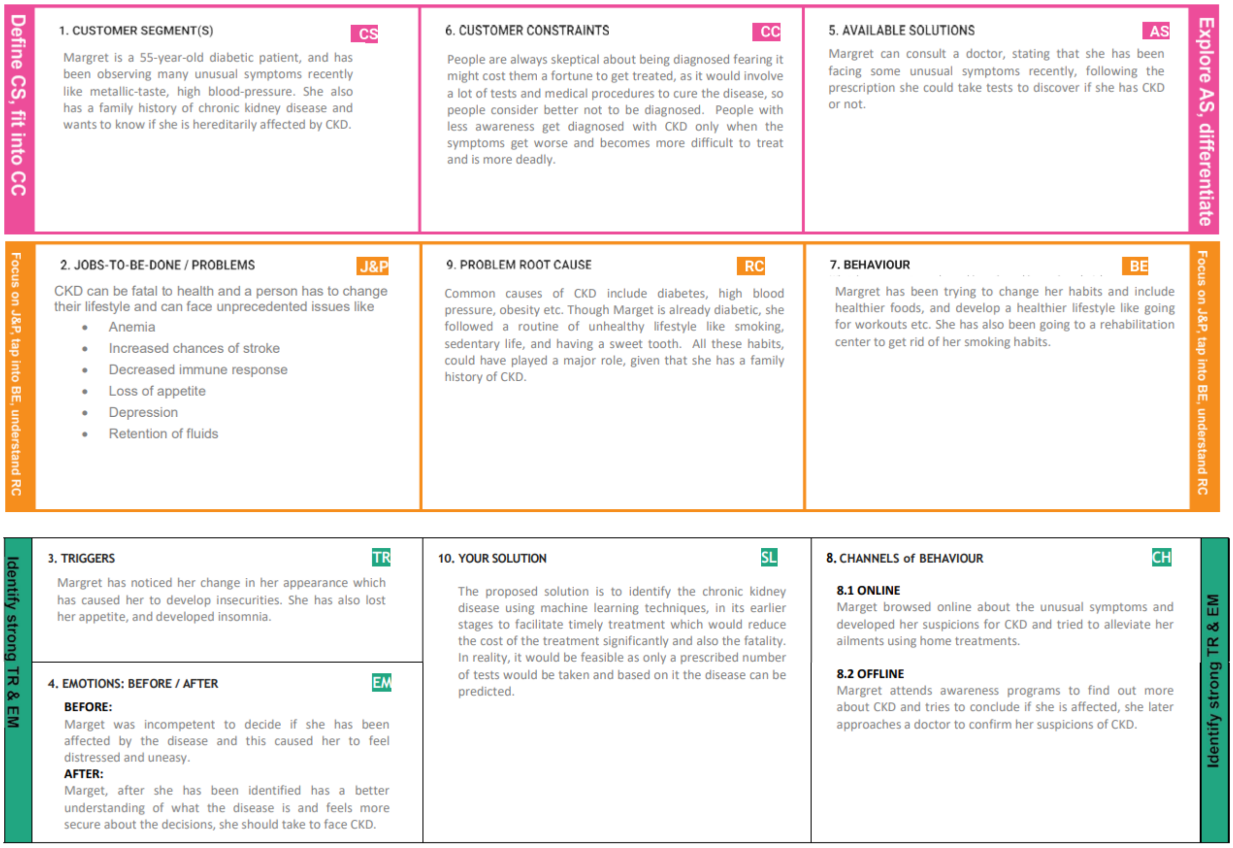
Awareness about exercising

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Chronic kidney disease (CKD) means your kidneys are damaged and can’t filter blood the way they should. The main risk factors for developing kidney disease are diabetes, high blood pressure, heart disease, and a family history of kidney failure. To detect CKD Earlier using machine learning.
2.	Idea / Solution description	Medical tests that are taken for different purposes could contain valuable information concerning kidney diseases. The attributes of various medical tests are investigated to distinguish which features may contain helpful information about the disease and we make use of that features to build the model to predict the kidney disease.
3.	Novelty / Uniqueness	By analysing different predictions models of machine learning and finding the best of it based on its accuracy.
4.	Social Impact / Customer Satisfaction	Early diagnosis is made possible and the cost for testing CKD gets reduced.
5.	Business Model (Revenue Model)	Generating income through direct customers or collaborate with healthcare centres.
6.	Scalability of the Solution	Deep Neural Network model that takes in all the features to detect whether the particular patient is affected by the chronic kidney disease or not, so accuracy is guaranteed.

3.4 Problem Solution Fit



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Margret is a 55-year-old diabetic patient, and has been observing many unusual symptoms recently like metallic-taste, high blood-pressure. She also has a family history of chronic kidney disease and wants to know if she is hereditarily affected by CKD.

6. CUSTOMER CONSTRAINTS CC

People are always skeptical about being diagnosed fearing it might cost them a fortune to get treated, as it would involve a lot of tests and medical procedures to cure the disease, so people consider better not to be diagnosed. People with less awareness get diagnosed with CKD only when the symptoms get worse and becomes more difficult to treat and is more deadly.

5. AVAILABLE SOLUTIONS AS

Margret can consult a doctor, stating that she has been facing some unusual symptoms recently, following the prescription she could take tests to discover if she has CKD or not.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

CKD can be fatal to health and a person has to change their lifestyle and can face unprecedented issues like

- Anemia
- Increased chances of stroke
- Decreased immune response
- Loss of appetite
- Depression
- Retention of fluids

9. PROBLEM ROOT CAUSE RC

Common causes of CKD include diabetes, high blood pressure, obesity etc. Though Marget is already diabetic, she followed a routine of unhealthy lifestyle like smoking, sedentary life, and having a sweet tooth. All these habits, could have played a major role, given that she has a family history of CKD.

7. BEHAVIOUR BE

Margret has been trying to change her habits and include healthier foods, and develop a healthier lifestyle like going for workouts etc. She has also been going to a rehabilitation center to get rid of her smoking habits.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

Margret has noticed her change in her appearance which has caused her to develop insecurities. She has also lost her appetite, and developed insomnia.

4. EMOTIONS: BEFORE / AFTER EM

BEFORE:
Marget was incompetent to decide if she has been affected by the disease and this caused her to feel distressed and uneasy.

AFTER:
Marget, after she has been identified has a better understanding of what the disease is and feels more secure about the decisions, she should take to face CKD.

10. YOUR SOLUTION SL

The proposed solution is to identify the chronic kidney disease using machine learning techniques, in its earlier stages to facilitate timely treatment which would reduce the cost of the treatment significantly and also the fatality. In reality, it would be feasible as only a prescribed number of tests would be taken and based on it the disease can be predicted.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
Marget browsed online about the unusual symptoms and developed her suspicions for CKD and tried to alleviate her ailments using home treatments.

8.2 OFFLINE
Margret attends awareness programs to find out more about CKD and tries to conclude if she is affected, she later approaches a doctor to confirm her suspicions of CKD.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

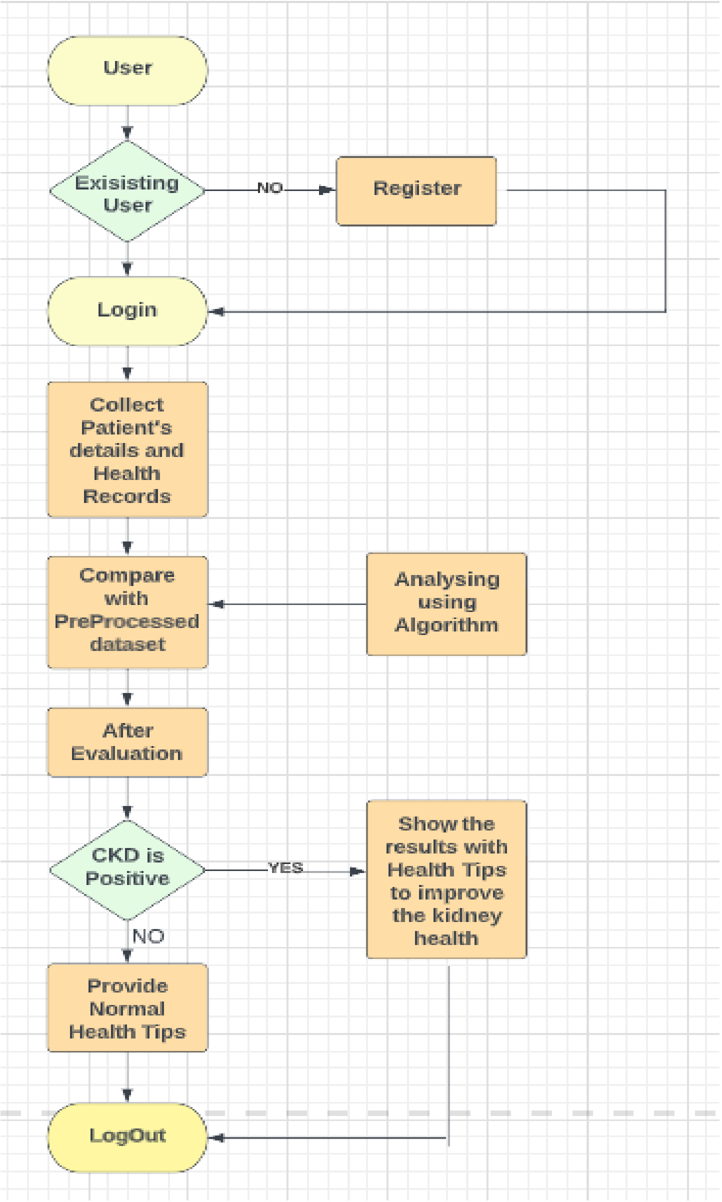
Fr.No	Functional Requirement(Epic)	SubRequirement(Story/Subtask)
FR-1	Registration	As a user, I can register for the application by entering my email, password, and confirming my password.
FR-2	Login	As a user, I can log into the application by entering email & password
FR-3	Data Cleaning	Handling the missing values Handling Outliers Label Encoding
FR-4	ML Model Building	Splitting dataset into train set and test set Choosing Appropriate Model Fitting and Evaluating the Model
FR-5	Entry Form	As a user, I can enter the data to get the result
FR-6	Train Model on IBM	Register for IBM cloud Creating Watson Studio Project Creating Cloud Object Storage Deploy Model on Watson Machine Learning
FR-7	Flask Integration	Routing and rendering html files Connection with database Requesting ML token from IBM Sending request to the deployed Model
FR-8	Result	As a user, I can get the results and navigate back to entry form

4.2 Non-Functional Requirements

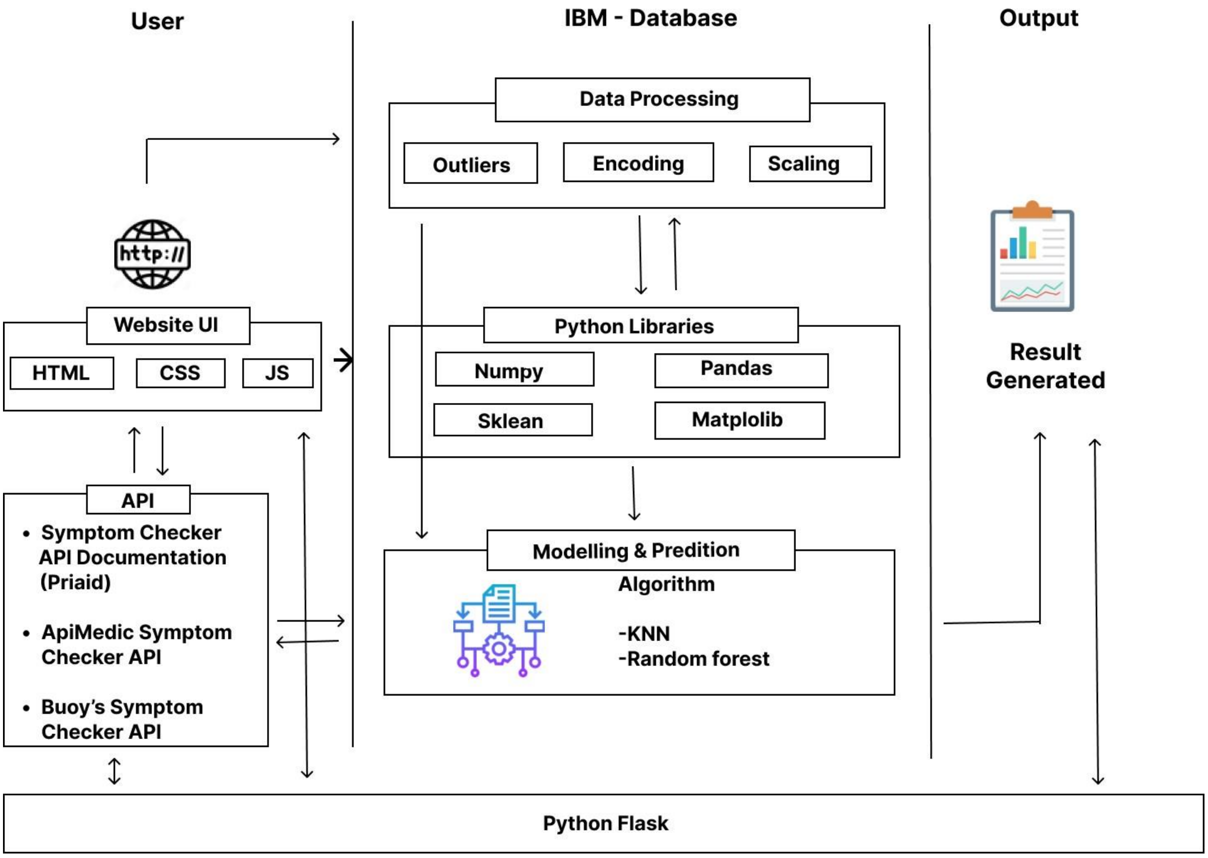
NFr.No	**Non-Functional Requirement**	**Description**
NFR-1	Usability	Easy to use and learn
NFR-2	Security	User data integrity
NFR-3	Reliability	Reliable transport of data
NFR-4	Performance	Less Response time
NFR-5	Availability	No Down time
NFR-6	Scalability	Can handle large number of users at a time

5. PROJECT DESIGN

5.1 Data Flow Diagram


User
Exisisting User
NO
Register
Login
Collect Patient's details and Health Records
Compare with PreProcessed dataset
Analysing using Algorithm
After Evaluation
CKD is Positive
YES
Show the results with Health Tips to improve the kidney health
NO
Provide Normal Health Tips
LogOut

5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirements(Epic)	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
User (Webuser)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password	I can create a new account ,store my data and can use the service	High	Sprint-1
	Login	USN-2	As a user, I can log into the application by entering email & password	I can login to my account and access the data	High	Sprint-1
	Data Cleaning	USN-3	Handling the missing values Handling Outliers Label Encoding	I can add/remove the wanted & unwanted data	High	Sprint-2
	Entry Form	USN-4	As a user, I can enter the data to get the result	I can enter the data to get the result	High	Sprint-3
	Result	USN-5	As a user, I can get the results and navigate back to entry form	I can get the results and navigate back to entry form	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

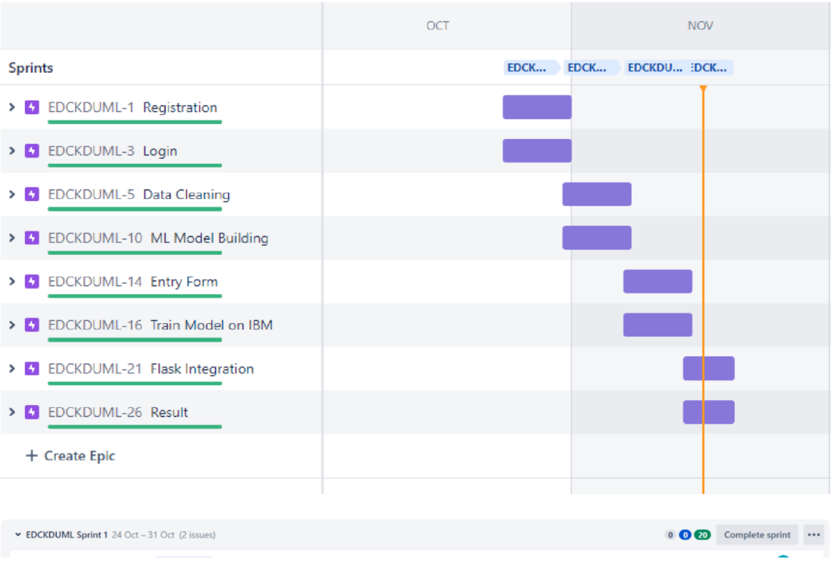
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	10	High	Avinash
	Login	USN-2	As a user, I can log into the application by entering email & password	10	High	Gokul
Sprint-2	Data Cleaning	USN-3	Handling the missing values	4	High	Edwin Kingsten
		USN-4	Handling Outliers	3	High	Edwin Kingsten
		USN-5	Label Encoding	3	High	Edwin Kingsten
	ML Model Building	USN-6	Splitting dataset into train set and test set	2	High	Edwin Kingsten
		USN-7	Choosing Appropriate Model	3	High	Edwin Kingsten
		USN-8	Fitting and Evaluating the Model	5	High	Edwin Kingsten
Sprint-3	Entry Form	USN-9	As a user, I can enter the data to get the result	10	High	Gokul
	Train Model on IBM	USN-10	Register for IBM cloud	2	High	Manikandan
		USN-11	Creating Watson Studio Project	2	High	Manikandan
		USN-12	Creating Cloud Object Storage	2	High	Manikandan
		USN-13	Deploy Model on Watson Machine Learning	4	High	Manikandan
Sprint-4	Flask Integration	USN-14	Routing and rendering html files	3	High	Avinash
		USN-15	Connection with database	3	High	Avinash
		USN-16	Requesting ML token from IBM	5	High	Manikandan
		USN-17	Sending request to the deployed Model	5	High	Manikandan

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
	Result	USN-18	As a user, I can get the results and navigate back to entry form	4	High	Avinash

6.2 Sprint Delivery Schedule

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	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	15 ov 2022

6.3 Reports From JIRA



7.CODING & SOLUTIONING

7.1 Login

Users are able to login to the system by filling in the details like their username and password, if the entered details are correct it will leave the message like ‘ Logged in successfully’ ; if not it will leave the message like ‘ incorrect username / password’.

Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=PT+Sans:ital@1&display=swap"
rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@1,700&display=swap"
rel="stylesheet">
</head>
<style>
    html{
        border: 0;
        margin: 0;
        width: 100vw;
        height: 100vh;
        background-image: url("{{url_for('static', filename='clean-medical-background_53876-
116875.webp')}}");
        background-size: cover;
    }
    .header{
        color: #392968;
        text-align: center;
        font-size: 50px;
        font-family: 'Lato', sans-serif;
    }
    .box{
        width: 300px;
        margin: auto auto auto auto;
        border-radius: 50px;
        text-align: center;
        padding-top: 10px;
        padding-bottom: 10px;
```

```
}
body{
   border: 0;
   margin: 0;
   height: 100vh;
   display: flex;
   flex-direction: column;
}
.inner-box{
   width: 250px;
   margin: auto;
}
.lil{
   padding: 8px;
   margin: 10px;
   border-radius: 5px;
   border: #30D5C8 solid;
   width: 200px;
   font-family: 'Lato', sans-serif;
   font-weight: bolder;
   color: #392968;
   font-size: 18px;
}
.li{
   border-radius: 5px;
   padding: 8px;
   margin: 10px 10px 20px 10px;
   background-color: white;
   border: 3px turquoise solid;
   color: #392968;
   font-size: 20px;
   font-family: 'Lato', sans-serif;
   font-weight: bolder;

}


.li:hover{
   border-radius: 5px;
   border:3px white solid;
   background-color: #30D5C8;
   color: white;
}

a{
   text-decoration: none;
}
.link{
   color: #392968;
   font-size: 20px;
```

```
        font-family: 'PT Sans', sans-serif;
        font-weight: bolder;
    }
    .hes{
        color: #392968;
        font-size: 22px;
        margin-bottom: 8px;
        font-family: 'Lato', sans-serif;
        font-weight: bolder;
    }
    .error{
        position: absolute;
        bottom: 0;
        border: 3px solid red;
        color: red;
        border-radius: 6px;
        padding: 10px;
        background-color: white;
        font-family: 'PT Sans', sans-serif;
        font-size: 20px;
        font-weight: bolder;
        left: 50%;
        transform: translateX(-50%);
    }
    .none{
        visibility: hidden;
    }
</style>
<body>
    {% if date %}
    <div class="error" >{{ message }}</div>
    {% endif %}
   <div class="header">
        <p>Chronic Kidney Disease Detector</p>
   </div>
   <div class = "box">
        <form action="testing2" method="post" class="inner-box">
            <div class="hes">Login</div>
            <input  class = "lil" name="email" type="email" required placeholder="Email
Address">
            <input  class = "lil" name="password" type="password" required
placeholder="Password">
            <button class = "li" >Sign in</button>
        </form>
        <a class = "link" href="registration">Go to Registration Page</a>
    </div>
    <script>
        setTimeout(()=>{
            error = document.querySelector(".error")
            if (error){
```

```
            error.classList.add("none")}
        },5000)
    </script>
</body>
</html>
```

7.2 Registration

Users can sign up into the system by entering their details. If the account already exists it will leave the message like ‘Account already exists’ / ‘ Enter the valid email address’ / ‘name must contain only characters and numbers’. If users entered the correct details then it will send the message like ‘ You are successfully registered click signin !’

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=PT+Sans:ital@1&display=swap"
rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@1,700&display=swap"
rel="stylesheet">
    <style>
    html{
        border: 0;
        margin: 0;
        width: 100vw;
        height: 100vh;
        background-image: url("{{url_for('static', filename='clean-medical-background_53876-
116875.webp')}}");
        background-size: cover;
    }
    .header{
        color: #392968;
        text-align: center;
        font-size: 50px;
        font-family: 'Lato', sans-serif;
        height: 20%;
    }
    body{
```

```
    display: flex;
    flex-direction: column;
    margin: 0;
    height: 100%;
}
.box{
    height: 80%;
    width: 100%;
    height: 300px;
    margin: 100px auto auto auto;
    display: flex;
    flex-direction: column;
    align-items: center;
}
a{
    text-decoration: none;
}
.inputs{
    display: block;
    padding: 8px;
    margin: 8px;
    border-radius: 5px;
    border: #30D5C8 solid;
    width: 200px;
    font-family: 'Lato', sans-serif;
    font-weight: bolder;
    color: #392968;
    font-size: 18px;
}
.sign-up{
    display: block;
    padding: 8px 8px 8px 8px;
    margin: 8px 8px 16px 8px;
    border-radius: 5px;
    border: #30D5C8 solid;
    width:auto;
    font-family: 'Lato', sans-serif;
    font-weight: bolder;
    color: #392968;
    font-size: 18px;
}
.link{
    color: #392968;
    font-size: 20px;
    font-family: 'PT Sans', sans-serif;
    font-weight: bolder;
}
.hes{
    color: #392968;
    font-size: 22px;
```

```
        margin-bottom: 8px;
        font-family: 'Lato', sans-serif;
        font-weight: bolder;
    }
    .error{
        position: absolute;
        bottom: 0;
        border: 3px solid red;
        color: red;
        border-radius: 6px;
        padding: 10px;
        background-color: white;
        font-family: 'PT Sans', sans-serif;
        font-size: 20px;
        font-weight: bolder;
        left: 50%;
        transform: translateX(-50%);
    }
    .none{
        visibility: hidden;
    }
    </style>
</head>
<body>
    {% if date %}
    <div class="error" >{{ message }}</div>
    {% endif %}
    <div class="header">
        <p>Chronic Kidney Disease Detector</p>
   </div>
    <form action="testing" method="post" class="box">
        <div class="hes">Register</div>
        <input class="inputs" name="name" minlength="4" type="text" required
placeholder="Name">
        <input class="inputs" name="email" type="email" required
placeholder="Email">
        <input class="inputs" name="password" minlength="8" type="password"
required placeholder="Password">
        <input class="inputs" name="re-enter" type="password" required
placeholder="Re-Enter Password">
        <button class="sign-up">Sign Up</button>
        <a class="link" href="login">Go to Login Page</a>
    </form>
    <script>
        setTimeout(()=>{
            error = document.querySelector(".error")
            error.classList.add("none")
        },5000)
    </script>
```

```
</body>
</html>
```

7.3 Entry form

The user enters the credentials like Blood pressure , Blood urea , Diabetes Mellitus status, Haemoglobin count , Hyper tension status, Pus cell , Packed cell volume, Pedal Edema , Red blood cell count and Specific gravity to detect the disease.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=PT+Sans:ital@1&display=swap"
rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@1,700&display=swap"
rel="stylesheet">
    <style>
        html{
        border: 0;
        margin: 0;
        width: 100vw;
        height: 100vh;
        background-image: url("{{url_for('static', filename='doctor-looks-kidney-hologram-
checks-test-result-virtual-interface-analyzes-data-kidney-disease-stones-innovative-
technologies-medicine-future_99433-6534.webp')}}");
        background-size: cover;
        }
        body{
            border: 0;
            margin: 0;
            height: 100%;
            width: 100%;
            display: flex;
            flex-direction: column;
        }
        .header{
            height: 20%;
            width: 100%;
        }
        .body{
```

```
    height: 80%;
    width: 80%;
    margin: auto;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}

.inputs{
    display: flex;
    flex-direction: row;
    flex-wrap: wrap;
    align-items: center;
    justify-content: center;
}
.s-input{
    width: 150px;
    display: block;
    margin: 5px;
    padding: 8px;
    border-radius: 5px;
    border: red solid;
    font-family: 'Lato', sans-serif;
    font-weight: bolder;
    font-size: 18px;
    color: #392968;
}
.post-btn{
    display: block;
    padding: 8px 8px 8px 8px;
    margin: 8px 8px 16px 8px;
    border-radius: 5px;
    border: #30D5C8 solid;
    width:auto;
    font-family: 'PT Sans', sans-serif;
    font-weight: bolder;
    color: #392968;
    font-size: 18px;
}
.post-btn:hover{
    border: white solid;
    color: white;
    background-color: #30D5C8;
}
::placeholder{
    color: #a9a9a9;
}
.items{
    padding: 8px;
```

```
            background-color: white;
            width: 150px;
            border-radius: 2px;
            cursor: pointer;
        }
        .items:hover{
            background-color: red;
            color: white;
        }
        .hid{
            position: absolute;
            top:100%;
            visibility: hidden;
        }
        .drop:focus + .hid{
            visibility: visible;
        }
        .hid:hover{
            visibility: visible;
        }
        .h-input{
            width: 166px;
            display: block;
            margin: 0 5px 5px 5px;
            border-radius: 6px;
            border: red solid;
            font-family: 'Lato', sans-serif;
            font-weight: bolder;
            font-size: 18px;
            color: #392968;
            background-color: white;
            z-index: 10;
        }

    </style>
</head>
<body>
    <div class="header"></div>
    <form action="results" method="post" class="body">
        <div class="inputs">
            <input step="any" required name="bp" type="number" class="s-input" placeholder="Blood Pressure">
            <input step="any" required name="bu" type="number" class="s-input" placeholder="Blood Urea">

            <!-- <select name="dm" class="s-input select">
                <option value="" selected disabled>Diabetes Mellitus</option>
                <option value="1">Yes</option>
                <option value="0">No</option>
            </select> -->
            <div style="position: relative;">
                <input name="dm" type="text" class="drop drop-2 s-input select" readonly required placeholder="Diabetes Mellitus">
                <div class="hid h-input">
                    <div class="items items-2">Yes</div>
                    <div class="items items-2">No</div>
                </div>
            </div>

            <input step="any" required name="hemo" type="number" class="s-input" placeholder="Hemoglobin">

            <!-- <select name="htn" class="s-input select">
                <option value="" selected disabled>Hypertension</option>
                <option value="1">Yes</option>
                <option value="0">No</option>
            </select> -->

            <div style="position: relative;">
                <input name="htn" type="text" class="drop drop-1 s-input select" readonly required placeholder="Hypertension">
                <div class="hid h-input">
                    <div class="items items-1">Yes</div>
                    <div class="items items-1">No</div>
                </div>
            </div>

            <!-- <select name="pc" class="s-input select">
                <option value="" selected disabled>Pus Cell</option>
                <option value="0">Normal</option>
                <option value="1">Abnormal</option>
            </select> -->

            <div style="position: relative;">
                <input name="pc" type="text" class="drop drop-3 s-input select" readonly required placeholder="Pus Cell">
                <div class="hid h-input">
                    <div class="items items-3">Normal</div>
                    <div class="items items-3">Abnormal</div>
                </div>
            </div>

            <input step="any" required name="pcv" type="number" class="s-input" placeholder="Packed Cell Volume">

            <!-- <select name="pe" class = "s-input select">
                <option value="" selected disabled >Pedal Edema</option>
                <option value="1">Yes</option>
                <option value="0">No</option>
            </select> -->

            <div style="position: relative;">
                <input name="pe" type="text" class="drop drop-4 s-input select" readonly
required placeholder="Pedal Edema">
                <div class="hid h-input">
                    <div class="items items-4">Yes</div>
                    <div class="items items-4">No</div>
                </div>
            </div>

            <input step="any" required name="rc" type="number" class="s-input"
placeholder="Red Blood Cells Count">
            <input step="any" required name="sg" type="number" class="s-input"
placeholder="Specific Gravity">
        </div>
        <button class="post-btn" >Get Results</button>
    </form>
    <script>
        items1 = document.querySelectorAll(".items-1")
        drop1 = document.querySelector(".drop-1")
        items1.forEach(element => {
            element.addEventListener("click",()=>{
                drop1.value = element.innerHTML
            })
        });

        items2 = document.querySelectorAll(".items-2")
        drop2 = document.querySelector(".drop-2")
        items2.forEach(element => {
            element.addEventListener("click",()=>{
                drop2.value = element.innerHTML
            })
        });

        items3 = document.querySelectorAll(".items-3")
        drop3 = document.querySelector(".drop-3")
        items3.forEach(element => {
            element.addEventListener("click",()=>{
                drop3.value = element.innerHTML
            })
        });

        items4 = document.querySelectorAll(".items-4")
        drop4 = document.querySelector(".drop-4")
        items4.forEach(element => {
            element.addEventListener("click",()=>{
                drop4.value = element.innerHTML
            })
        });
    </script>
</body>
</html>
```

7.4 Results

Users can get their respective results based on their disease detection essentials . They will get result as " Positive / Negative" to detect that disease . If the result is " Positive" then the patient (user) is affected by the disease and if the result is "Negative" then the patient (user) is not affected by the disease.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=PT+Sans:ital@1&display=swap"
rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@1,700&display=swap"
rel="stylesheet">
    <style> html{
        border: 0;
        margin: 0;
        width: 100vw;
        height: 100vh;
        background-image: url("{{url_for('static', filename='abstract-blue-background-simple-
design-for-your-website-free-vector.webp')}}");
        background-size: cover;
    }
    body{
        margin: 0;
        border: 0;
        height: 100%;
        width: 100%;
        display: flex;
    }
    .div{
        display: block;
        margin: auto;
        text-align: center;
    }
    .result{
        font-family: 'Lato', sans-serif;
        font-size: 40px;
        font-weight: bolder;
        color: white;
        margin-bottom: 20px;
    }
    .navi{
        font-family: 'PT Sans', sans-serif;
        font-size: 30px;
        font-weight: bolder;
        color: whitesmoke;
    }
    a{
        text-decoration: none;
        color: whitesmoke;
    }

    a:hover{
        color:plum;
    }

    </style>

</head>
<body>
    <div class="div">
        <div class="result">
            {{ result }}
        </div>

        <div class="navi">
            <a href="entryform">Go Back To Entry Form</a>
        </div>
    </div>

</body>
</html>
```

8. TESTING

8.1 Test Cases

Test Scenarios

Login

1. Verify user is able to see the go to registration Page
2. User can log in to their account they are registered with
3. If user's Email or password is wrong
4. If the user email is not registered

Registration

1. Verify user is able to see the go to Login Page.
2. User can register their account.
3. If the email is not valid.
4. If password doesnt meet the requirements.
5. If password doesnt match the confirm password.

Entry Form

1.User can enter the details.

2. If any details are missed.
3. Clicking get results.
4. User wait for results.

Results

1. User can see the results.
2. User can see go to entry form.

8.2 User acceptance testing

The project has been tested extensively with a number of users. The users found the interface very easy to use. The Web pages were colorful and attractive. There were no unnecessary details in the web page. It was clean and simple that any new user could master. The data input format was also simple. The user can login to the system by entering username and password . Various inputs have been given by the users to test the consistency of the model. The model proved itself and all the users accepted the model as reliable and convenient.

9. RESULTS

9.1 Performance Metrics

The application that we have developed has better performance in speed. The user interface simple and can be used by any people. The results will be gotten by the user within seconds which reduces the unwanted fear and tension during the testing period. The application developed also performed well with no glitches or lag found during the testing phase.

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

- This application gives the instant result of the disease stage in case of symptoms appear.
- This early detection serves facility for early treatment for the Chronic kidney disease.
- This detector is more efficient than the existing system which makes the system reliable among large number of users.
- This detector gets less data credentials to detect the disease which will not defect the actual efficiency of the detector.

10.2 Disadvantages

- High implementation costs.

11. CONCLUSION

CKD in early stages occurs frequently in the studied population. The proposed diagnostic algorithm seems to be a powerful tool to identify subjects at risk of CKD. The system detects the disease at instance after the credentials were given which helps the user to start the treatment as soon as possible. Further developments can be done from this detector as it is compatible with alterations and can be easily installed .

12. FUTURE ENHANCEMENTS

The developed system can further developed with various disease detection . This system detects the CKD which can be integrated with another different system and work alongside with it. This system can be developed to detect the other kidney related issues like stone detection and more.

13. APPENDIX

13.1 Source code:

Index.py

```
from flask import Flask, request, render_template, redirect
import requests
import mysql.connector as mariadb
from mysql.connector import Error

user = None

API_KEY = "Im5Owvbo4HbMZgEjLP9zMlCv-d_kwEGOkQgNoDH5C38M"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
error = 0
exists = 0
reg = 0
miss = 0

app = Flask(__name__)

@app.route("/")
def redirct():
    return redirect("/login")
@app.route("/login")
def login():
    message = ""
    global error
    global reg
    date = False
    if error:
        date = True
        message = "Wrong Password"
        error = 0
    else:
        date = False

    if reg:
        date = True
        message = "Email Not Registered"
        reg = 0

    return render_template('login.html',date = date,message=message)
```

```
@app.route("/registration")
def registration():
    message = ""
    date = False
    global exists
    global miss
    if exists:
        date = True
        message = "Account Already Registered"
        exists = 0
    else:
        date = False
    if miss:
        miss = 0
        date = True
        message = "Password Not Match"
    return render_template('registration.html',date=date,message = message)




@app.route("/testing",methods = ["POST"])
def redir():
    data = request.form.to_dict()
    try:
        if data["password"] != data["re-enter"]:
            print("confirmation wrong")
            global miss
            miss = 1
            return redirect("/registration")
        conn = mariadb.connect(
            user="sql12575355",
            password="QUz4kAunlf",
            host="sql12.freesqldatabase.com",
            database="sql12575355"

        )
        cursor = conn.cursor()
        checkquery = "SELECT * FROM USERS WHERE EMAIL =
'{}'".format(data["email"])
        cursor.execute(checkquery)
        flag = cursor.fetchall()
        if flag:
            global exists
            exists = 1
            return redirect("/registration")
        else:
            print("empty")
            query = "INSERT INTO USERS
```

```
VALUES('{}','{}','{}')".format(data["name"],data["email"],data["password"])
            cursor.execute(query)

    except Exception as e:
        return redirect("/registration",code = 500)
    else:
        conn.commit()
        conn.close()
    return redirect("/login")





@app.route("/testing2",methods = ["POST"])
def redire():
    data = request.form.to_dict()
    try:
        conn = mariadb.connect(
            user="sql12575355",
            password="QUz4kAunlf",
            host="sql12.freesqldatabase.com",
            database="sql12575355"

        )
        cursor = conn.cursor()
        checkquery = "SELECT * FROM USERS WHERE EMAIL =
'{}'".format(data["email"])
        cursor.execute(checkquery)
        out = cursor.fetchone()
        if not out:
            global reg
            reg = 1
            return redirect("/login")
        if out[2] != data["password"]:
            print(out[1],data["password"])
            global error
            error = 1
            return redirect("/login")
    except Error as e:
        return str(e)
    else:
        conn.commit()
        conn.close()
        global user
        user = 1
    return redirect("/entryform")
```

```
@app.route("/entryform")
def entryform():
    if user:
        return render_template("entryform.html")
    else:
        return redirect("/login")





@app.route("/results",methods = ["POST"])
def results():
    data = request.form.to_dict()
    bp = float(data["bp"])
    bu = float(data["bu"])
    if data["dm"] == "Yes":
        dm = 1
    else:
        dm = 0
    hemo = float(data["hemo"])
    if data["htn"] == "Yes":
        htn = 1
    else:
        htn = 0
    if data["pc"] == "Abnormal":
        pc = 1
    else:
        pc = 0
    pcv = float(data["pcv"])
    if data["pe"] == "Yes":
        pe = 1
    else:
        pe = 0
    rc = float(data["rc"])
    sg = float(data["sg"])
    datas = [bp , bu , dm , hemo , htn , pc , pcv , pe , rc , sg]
    print(datas)
    payload_scoring = {"input_data": [{"fields": ["bp" , "bu" , "dm"  , "hemo" , "htn" ,
"pc" ,  "pcv" , "pe" , "rc" , "sg"], "values": [datas]}]}

    response_scoring = requests.post('https://eu-
de.ml.cloud.ibm.com/ml/v4/deployments/kidney_pred/predictions?version=2022-11-05',
json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    result = response_scoring.json()["predictions"][0]["values"][0][0]
    if result:
        result = "Positive"
    else:
        result = "Negative"
    return render_template("results.html",result = result)




if __name__ == "__main__":
    app.run(debug=True)
```

Login.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=PT+Sans:ital@1&display=swap"
rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@1,700&display=swap"
rel="stylesheet">
</head>
<style>
    html{
        border: 0;
        margin: 0;
        width: 100vw;
        height: 100vh;
        background-image: url("{{url_for('static', filename='clean-medical-background_53876-
116875.webp')}}");
        background-size: cover;
    }
    .header{
        color: #392968;
        text-align: center;
        font-size: 50px;
        font-family: 'Lato', sans-serif;
    }
    .box{
        width: 300px;
        margin: auto auto auto auto;
```

```
   border-radius: 50px;
   text-align: center;
   padding-top: 10px;
   padding-bottom: 10px;
}
body{
   border: 0;
   margin: 0;
   height: 100vh;
   display: flex;
   flex-direction: column;
}
.inner-box{
   width: 250px;
   margin: auto;
}
.lil{
   padding: 8px;
   margin: 10px;
   border-radius: 5px;
   border: #30D5C8 solid;
   width: 200px;
   font-family: 'Lato', sans-serif;
   font-weight: bolder;
   color: #392968;
   font-size: 18px;
}
.li{
   border-radius: 5px;
   padding: 8px;
   margin: 10px 10px 20px 10px;
   background-color: white;
   border: 3px turquoise solid;
   color: #392968;
   font-size: 20px;
   font-family: 'Lato', sans-serif;
   font-weight: bolder;

}


.li:hover{
   border-radius: 5px;
   border:3px white solid;
   background-color: #30D5C8;
   color: white;
}

a{
   text-decoration: none;
```

```
    }
    .link{
        color: #392968;
        font-size: 20px;
        font-family: 'PT Sans', sans-serif;
        font-weight: bolder;
    }
    .hes{
        color: #392968;
        font-size: 22px;
        margin-bottom: 8px;
        font-family: 'Lato', sans-serif;
        font-weight: bolder;
    }
    .error{
        position: absolute;
        bottom: 0;
        border: 3px solid red;
        color: red;
        border-radius: 6px;
        padding: 10px;
        background-color: white;
        font-family: 'PT Sans', sans-serif;
        font-size: 20px;
        font-weight: bolder;
        left: 50%;
        transform: translateX(-50%);
    }
    .none{
        visibility: hidden;
    }
</style>
<body>
    {% if date %}
    <div class="error" >{{ message }}</div>
    {% endif %}
   <div class="header">
        <p>Chronic Kidney Disease Detector</p>
   </div>
   <div class = "box">
        <form action="testing2" method="post" class="inner-box">
            <div class="hes">Login</div>
            <input  class = "lil" name="email" type="email" required placeholder="Email 
Address">
            <input  class = "lil" name="password" type="password" required 
placeholder="Password">
            <button class = "li" >Sign in</button>
        </form>
        <a class = "link" href="registration">Go to Registration Page</a>
    </div>
    <script>
        setTimeout(()=>{
            error = document.querySelector(".error")
            if (error){
            error.classList.add("none")}
        },5000)
    </script>
</body>
</html>
```

Registration.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=PT+Sans:ital@1&display=swap"
rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@1,700&display=swap"
rel="stylesheet">
    <style>
    html{
        border: 0;
        margin: 0;
        width: 100vw;
        height: 100vh;
        background-image: url("{{url_for('static', filename='clean-medical-background_53876-
116875.webp')}}");
        background-size: cover;
    }
    .header{
        color: #392968;
        text-align: center;
        font-size: 50px;
        font-family: 'Lato', sans-serif;
        height: 20%;
    }
    body{
        display: flex;
        flex-direction: column;
        margin: 0;
        height: 100%;
    }
```

```
.box{
   height: 80%;
   width: 100%;
   height: 300px;
   margin: 100px auto auto auto;
   display: flex;
   flex-direction: column;
   align-items: center;
}
a{
   text-decoration: none;
}
.inputs{
   display: block;
   padding: 8px;
   margin: 8px;
   border-radius: 5px;
   border: #30D5C8 solid;
   width: 200px;
   font-family: 'Lato', sans-serif;
   font-weight: bolder;
   color: #392968;
   font-size: 18px;
}
.sign-up{
   display: block;
   padding: 8px 8px 8px 8px;
   margin: 8px 8px 16px 8px;
   border-radius: 5px;
   border: #30D5C8 solid;
   width:auto;
   font-family: 'Lato', sans-serif;
   font-weight: bolder;
   color: #392968;
   font-size: 18px;
}
.link{
   color: #392968;
   font-size: 20px;
   font-family: 'PT Sans', sans-serif;
   font-weight: bolder;
}
.hes{
   color: #392968;
   font-size: 22px;
   margin-bottom: 8px;
   font-family: 'Lato', sans-serif;
   font-weight: bolder;
}
.error{
        position: absolute;
        bottom: 0;
        border: 3px solid red;
        color: red;
        border-radius: 6px;
        padding: 10px;
        background-color: white;
        font-family: 'PT Sans', sans-serif;
        font-size: 20px;
        font-weight: bolder;
        left: 50%;
        transform: translateX(-50%);
    }
    .none{
        visibility: hidden;
    }
    </style>
</head>
<body>
    {% if date %}
    <div class="error" >{{ message }}</div>
    {% endif %}
    <div class="header">
        <p>Chronic Kidney Disease Detector</p>
   </div>
    <form action="testing" method="post" class="box">
        <div class="hes">Register</div>
        <input class="inputs" name="name" minlength="4" type="text" required placeholder="Name">
        <input class="inputs" name="email" type="email" required placeholder="Email">
        <input class="inputs" name="password" minlength="8" type="password" required placeholder="Password">
        <input class="inputs" name="re-enter" type="password" required placeholder="Re-Enter Password">
        <button class="sign-up">Sign Up</button>
        <a class="link" href="login">Go to Login Page</a>
    </form>
    <script>
        setTimeout(()=>{
            error = document.querySelector(".error")
            error.classList.add("none")
        },5000)
    </script>
</body>
</html>
```

Entry form.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=PT+Sans:ital@1&display=swap"
rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@1,700&display=swap"
rel="stylesheet">
    <style>
        html{
        border: 0;
        margin: 0;
        width: 100vw;
        height: 100vh;
        background-image: url("{{url_for('static', filename='doctor-looks-kidney-hologram-
checks-test-result-virtual-interface-analyzes-data-kidney-disease-stones-innovative-
technologies-medicine-future_99433-6534.webp')}}");
        background-size: cover;
        }
        body{
            border: 0;
            margin: 0;
            height: 100%;
            width: 100%;
            display: flex;
            flex-direction: column;
        }
        .header{
            height: 20%;
            width: 100%;
        }
        .body{
            height: 80%;
            width: 80%;
            margin: auto;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
        }
```

```
.inputs{
   display: flex;
   flex-direction: row;
   flex-wrap: wrap;
   align-items: center;
   justify-content: center;
}
.s-input{
   width: 150px;
   display: block;
   margin: 5px;
   padding: 8px;
   border-radius: 5px;
   border: red solid;
   font-family: 'Lato', sans-serif;
   font-weight: bolder;
   font-size: 18px;
   color: #392968;
}
.post-btn{
   display: block;
   padding: 8px 8px 8px 8px;
   margin: 8px 8px 16px 8px;
   border-radius: 5px;
   border: #30D5C8 solid;
   width:auto;
   font-family: 'PT Sans', sans-serif;
   font-weight: bolder;
   color: #392968;
   font-size: 18px;
}
.post-btn:hover{
   border: white solid;
   color: white;
   background-color: #30D5C8;
}
::placeholder{
   color: #a9a9a9;
}
.items{
   padding: 8px;
   background-color: white;
   width: 150px;
   border-radius: 2px;
   cursor: pointer;
}
.items:hover{
   background-color: red;
   color: white;
}
```

```
        .hid{
            position: absolute;
            top:100%;
            visibility: hidden;
        }
        .drop:focus + .hid{
            visibility: visible;
        }
        .hid:hover{
            visibility: visible;
        }
        .h-input{
            width: 166px;
            display: block;
            margin: 0 5px 5px 5px;
            border-radius: 6px;
            border: red solid;
            font-family: 'Lato', sans-serif;
            font-weight: bolder;
            font-size: 18px;
            color: #392968;
            background-color: white;
            z-index: 10;
        }

    </style>
</head>
<body>
    <div class="header"></div>
    <form action="results" method="post" class="body">
        <div class="inputs">
            <input step="any" required name="bp" type="number" class="s-input" placeholder="Blood Pressure">
            <input step="any" required name="bu" type="number" class="s-input" placeholder="Blood Urea">

            <!-- <select name="dm" class="s-input select">
                <option value="" selected disabled>Diabetes Mellitus</option>
                <option value="1">Yes</option>
                <option value="0">No</option>
            </select> -->

            <div style="position: relative;">
                <input name="dm" type="text" class="drop drop-2 s-input select" readonly required placeholder="Diabetes Mellitus">
                <div class="hid h-input">
                    <div class="items items-2">Yes</div>
                    <div class="items items-2">No</div>
                </div>
            </div>
            <input step="any" required name="hemo" type="number" class="s-input" placeholder="Hemoglobin">

            <!-- <select name="htn" class="s-input select">
                <option value="" selected disabled>Hypertension</option>
                <option value="1">Yes</option>
                <option value="0">No</option>
            </select> -->

            <div style="position: relative;">
                <input name="htn" type="text" class="drop drop-1 s-input select" readonly required placeholder="Hypertension">
                <div class="hid h-input">
                    <div class="items items-1">Yes</div>
                    <div class="items items-1">No</div>
                </div>
            </div>

            <!-- <select name="pc" class="s-input select">
                <option value="" selected disabled>Pus Cell</option>
                <option value="0">Normal</option>
                <option value="1">Abnormal</option>
            </select> -->

            <div style="position: relative;">
                <input name="pc" type="text" class="drop drop-3 s-input select" readonly required placeholder="Pus Cell">
                <div class="hid h-input">
                    <div class="items items-3">Normal</div>
                    <div class="items items-3">Abnormal</div>
                </div>
            </div>

            <input step="any" required name="pcv" type="number" class="s-input" placeholder="Packed Cell Volume">

            <!-- <select name="pe" class = "s-input select">
                <option value="" selected disabled >Pedal Edema</option>
                <option value="1">Yes</option>
                <option value="0">No</option>
            </select> -->

            <div style="position: relative;">
                <input name="pe" type="text" class="drop drop-4 s-input select" readonly required placeholder="Pedal Edema">
                <div class="hid h-input">
                    <div class="items items-4">Yes</div>
                    <div class="items items-4">No</div>
                </div>
            </div>

            <input step="any" required name="rc" type="number" class="s-input" placeholder="Red Blood Cells Count">
            <input step="any" required name="sg" type="number" class="s-input" placeholder="Specific Gravity">
        </div>
        <button class="post-btn" >Get Results</button>
    </form>
    <script>
        items1 = document.querySelectorAll(".items-1")
        drop1 = document.querySelector(".drop-1")
        items1.forEach(element => {
            element.addEventListener("click",()=>{
                drop1.value = element.innerHTML
            })
        });

        items2 = document.querySelectorAll(".items-2")
        drop2 = document.querySelector(".drop-2")
        items2.forEach(element => {
            element.addEventListener("click",()=>{
                drop2.value = element.innerHTML
            })
        });

        items3 = document.querySelectorAll(".items-3")
        drop3 = document.querySelector(".drop-3")
        items3.forEach(element => {
            element.addEventListener("click",()=>{
                drop3.value = element.innerHTML
            })
        });

        items4 = document.querySelectorAll(".items-4")
        drop4 = document.querySelector(".drop-4")
        items4.forEach(element => {
            element.addEventListener("click",()=>{
                drop4.value = element.innerHTML
            })
        });
    </script>
</body>
</html>
```

Results.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=PT+Sans:ital@1&display=swap" rel="stylesheet">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Lato:ital,wght@1,700&display=swap" rel="stylesheet">
    <style> html{
        border: 0;
        margin: 0;
        width: 100vw;
        height: 100vh;
        background-image: url("{{url_for('static', filename='abstract-blue-background-simple-design-for-your-website-free-vector.webp')}}");
        background-size: cover;
    }
    body{
        margin: 0;
        border: 0;
        height: 100%;
        width: 100%;
        display: flex;
    }
    .div{
        display: block;
        margin: auto;
        text-align: center;
    }
    .result{
        font-family: 'Lato', sans-serif;
        font-size: 40px;
        font-weight: bolder;
        color: white;
        margin-bottom: 20px;
    }
    .navi{
        font-family: 'PT Sans', sans-serif;
        font-size: 30px;
        font-weight: bolder;
        color: whitesmoke;
    }
    a{
        text-decoration: none;
        color: whitesmoke;
    }

    a:hover{
        color:plum;
    }

    </style>

</head>
<body>
    <div class="div">
        <div class="result">
            {{ result }}
        </div>

        <div class="navi">
            <a href="entryform">Go Back To Entry Form</a>
        </div>
    </div>

</body>
</html>
```

13.2 Github &Project demo link

Github : https://github.com/IBM-EPBL/IBM-Project-21744-1659790457

Demo Video:

https://drive.google.com/file/d/1G3eAxF2NT84OU4UzUCjqbHAsEBoDZq0C/view?usp=sharing