EARLY DETECTION OF CHRONIC KIDNEY DISEASE USING MACHINE LEARNING

PROJECT REPORT

TEAM ID :
PNT2022TMID41596

1. INTRODUCTION

1.1 PROJECT OVERVIEW :

Chronic kidney disease prediction is one of the most important issues in healthcare-analytics. The most interesting and challenging tasks in day-to-day lives as one third of the adult population is affected by chronic kidney disease (CKD), and millions die each year because they do not have access to affordable treatment. Chronic Kidney Disease can be cured, if treated in the early stages. The main aim of the project is to predict whether the patient have chronic kidney disease or not in a painless, accurate and faster way based on certain diagnostic measurement like Blood Pressure(BP), Albumin(Al) etc., and then appropriate treatment can be given based on the details provided by the model.

1.2 PURPOSE

The purpose of the project is to alert doctors for an early detection of kidney disease and hence ensure speedy recovery or prevention of kidney disease. This Project aims at creating a model for early detection of Chronic Kidney Disease using Machine Learning technology. The output is integrated with Flask. The front end developed in html is used to receive user input on various parameters needed to decide on the early detection of kidney disease.The same model is deployed into IBM cloud.

2. LITERATURE SURVEY:

2.1 EXISTING SYSTEM

Presently kidney disease is detected at late stages in many countries leading to loss of precious lives. There are very few means to identify them at an early stage.Most of the user details remain unverified and it's difficult to track the fake users. The user interface

of the application is not user friendly and the user must have a device with an android operating system with an active internet connection to interact with this application.

2.2 REÏERENCES

1. "What Is ChronicKidney Disease?". NationalInstitute of Diabetesand Digestive and Kidney Diseases. June 2017. Retrieved 19 December 2017.
2. "Kidney Failure". MedlinePlus. Retrieved 11 November 2017.
3. "Chronic Kidney DiseaseTests & Diagnosis". National Institute of Diabetes and Digestive and Kidney Diseases. October 2016. Retrieved 19 December 2017.
4. "Kidney Failure". NationalInstitute of Diabetesand Digestive and Kidney Diseases.Retrieved 11 November2017.
5. "Managing Chronic KidneyDisease". National Institute of Diabetes and Digestive and Kidney Diseases. October 2016.
6. "Eating Right for Chronic KidneyDisease | NIDDK". National Institute of Diabetes and Digestive and Kidney Diseases. Retrieved 5 September 2019.
7. "Mineral & Bone Disorder in Chronic KidneyDisease". National Institute Of Diabetes and Digestive and Kidney Diseases.November 2015. Retrieved19 December 2018
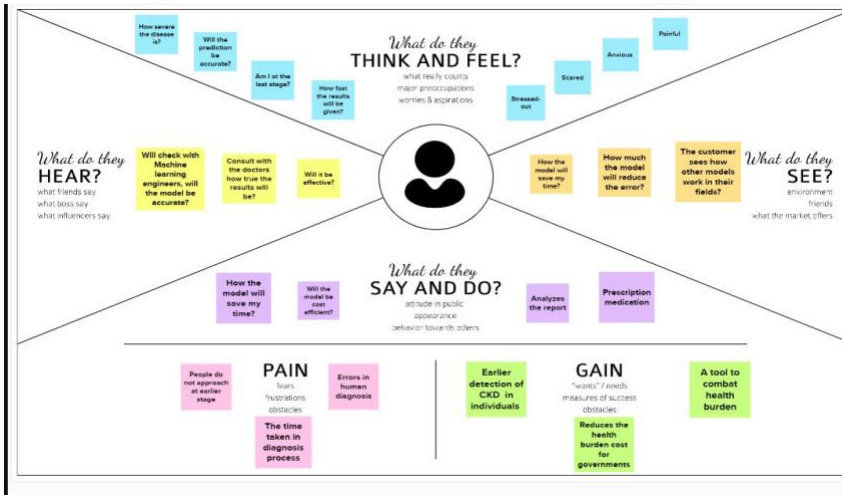
2.3 PROBLEM STATEMENT

Chronic Kidney Disease (CKD) is a serious medical condition that, if diagnosed 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. IDEAᴛION & PROPOSED SOLUᴛION:

3.1 EMPAᴛHY MAP CANVAS

3.2 IDEAᴛION AND BRAINSᴛORMING

3.3 PROPOSED SOLUTION

S.no	Parameter	Description
1.	Problem Statement(Problem to be solved)	- Chronic Kidney Disease (CKD) is a major medical problem hence Chronic kidney disease prediction is one of the most important issues in healthcare analytics. 10% of the population worldwide is

		affected by chronic kidney disease (CKD),and millions die each year because they do not have access to affordable treatment. ● Chronic kidney Disease can be cured, if treated in the early stages.
2.	Solution Description	● The idea is to detect the presence of kidney disease through machine learning based classification models. ● Early detection of chronic kidney disease is identified through various ML Algorithms such as Logistics Regression,Random Forest, Decision Tree, SupportVector Machines and KNN. ● Using these techniques, each algorithm's effectiveness is evalutated. ● A web app is developed that asks basic user details about kidney details and result is produced
3.	Novelty	● Aims to find the best machine learning model for the early prediction of chronic kidney disease by analyzing the essential parameters and comparing their predictive accuracies. ● Then collaborate the best machine learning model to an interactive user-interface which helps in the early detection of CKD and provide cure.
4.	Social Impact	● The main aim of this application is early prediction of chronic kidney

		disease that can possibly stop or slow the progression of this disease to the end stage.
5.	Business Model	● The widespread use of Machine Learning of predicting the CKD in the Medical Industries promotes medical innovation, lowers medical expenses, and improves medical quality. To cure the CKD patients at early stages. ● We can generate revenue through direct customers or can also collaborate with the health care sector and generate revenue from their customers.
6.	Scalability of solution	● Early prediction of CKD using Machine Learning that is more efficient to analyze the disease so that it can be cured on time. ● We can also use image data and apply Deep Learning techniques such as Multilayer Perceptron(MLP) etc., which will provide an improved accuracy than machine learning.

3.4 Píoblem Solution Ïit

Problem-Solution Fit

Project Title: Early Chronic Kidney Disease using Machine Learning
Team ID: PNT2022TMID24563

1. CUSTOMER SEGMENT(S) CS

All age people and Doctors.

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

- Costs and human errors associated with lab testing
- Separate expense for taking tests for CKD detection.

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

- Separate tests for CKD and analyzing the reports for CKD detection.
- Tests are taken only when the symptoms occur, but the symptoms show only when the patient is in critical stage.

2. PROBLEMS / PAINS + ITS FREQUENCY PR

- Early detection prevents severe damage of the kidney and lives of the patient.
- It also helps doctors to provide better treatment and to avoid critical conditions.

9. PROBLEM ROOT / CAUSE RC

- Increase in kidney diseases among people due to modern culture.
- Early stages of kidney disease do not show symptoms.
- Time wasted in the detection can be used in the treatment.

7. BEHAVIOR + ITS INTENSITY BE

Earlier detection and high accuracy directs patients to correct and proper medication which helps people to live long with healthy kidney.

3. TRIGGERS TO ACT TR

Knowledge about CKD and its seriousness, awareness among people about kidney diseases and early prevention through social media and news channels.

4. EMOTIONS BEFORE / AFTER EM

Before- Fear, Anxiety,confusion.

After - relaxed and tension free.

10. YOUR SOLUTION SL

Our model makes use of available reports and says whether the patient's kidney is affected by any diseases or not, if yes, then says in which stage is the patient now and helps doctor to provide better treatment to cure the patient.

8. CHANNELS of BEHAVIOR CH

ONLINE

Deploying as a web app it can be used in online from anywhere.

OFFLINE

Make use of cost efficient and proper testing laboratories

4. REQUIREMENT ANALYSIS

4.1 ÏUNCÏIONAL REQUIREMENÏ

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via password.
FR-3	Obtain Information	The system should be able to get the information for predicting the disease.
FR-4	Displaying Result	The system must be able to display whether the user is affected or not

4.2 NON ÏUNCÏIONAL REQUIREMENÏ

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Access to use the application is permitted only to the registered users

NFR-2	Security	Authentication is done for the security process.
NFR-3	Reliability	The user gets the correct and predicted value and standard results.
NFR-4	Performance	Lowering the total load time of prediction and user interaction
NFR-5	Availability	Easily available to everyone.
NFR-6	Scalability	To be able to change things.

5. PROJECT DESIGN

5.1 DAĬA ĬLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored
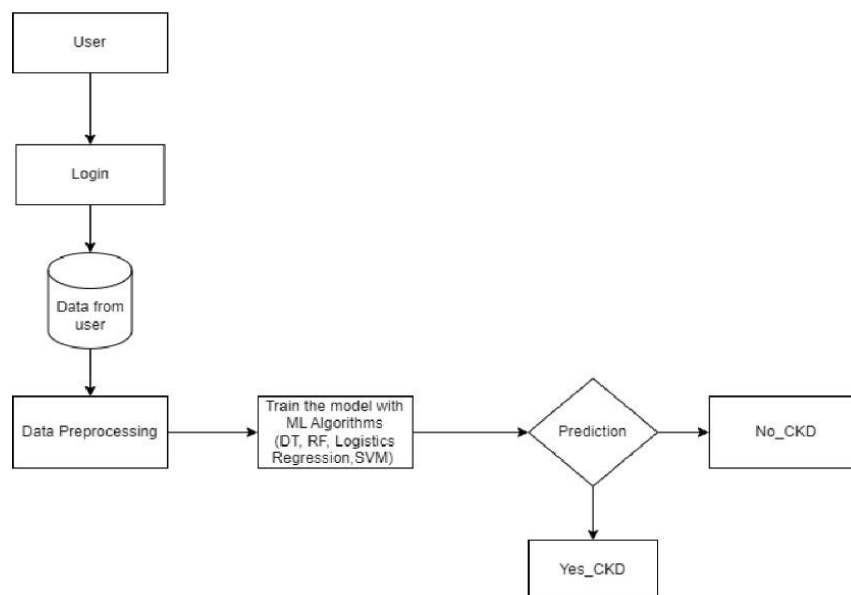
5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Technology Architecture
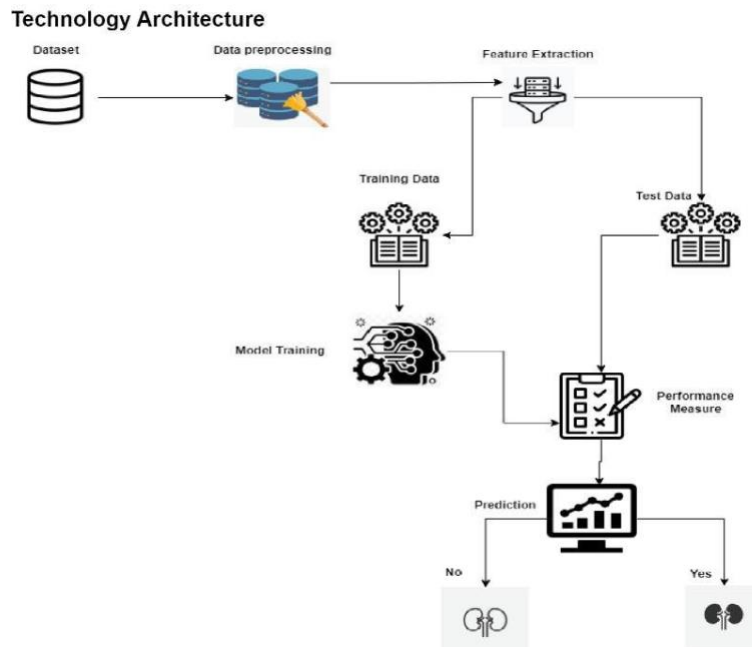
Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	How user interacts with application	HTML, CSS,Python Flask
2	Application Logic-1	Get input from the user	HTML,CSS,Python Flask
3	Application Logic-2	Predicts based on the provided input	Python
4	Application Logic-3	Displays the predicted Result	Python,HTML,CSS,Flask
5.	Machine Learning Model	Random Forest,Regression techniques,Decision tree and SVM	Classification Algorithms

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology

1.	Open-Source Frameworks	List the open-source frameworks used	Google colab,Jupyter notebook,IBM cloud and Flask.
2.	Scalable Architecture	Model can be scalable	Python
3.	Availability	It is used as a website(UI) or available in cloud	IBM cloud
4.	Performance	High accuracy	Machine Learning Classification techniques

5.3 USER SȊORIES

User Type	Functional Requireme nt (Epic)	User Story Number	User Story	Acceptance criteria	Priority	Release
Customer	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	can access account	High	Sprint-1
	Login	USN-2	As a user, I can log into the application	Login into account	High	Sprint-1
Data Entry	Enter data	USN-3	Enter symptoms and clinical data	Enter Clinical Data	High	Sprint-2
Customer (View Result)	View Result	USN-4	Result can be viewed by the user	View Result	Medium	Sprint-3
Administrator	Predict Disease	USN-5	As the admin, I build ML models to predict CKD	I deploy models within websites	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND SCHEDULING

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	USN-1	Collect the appropriate dataset for detection of Chronic Kidney Disease.	10	High	Amrutha A
Sprint-1	Data Pre-processing	USN-2	Transform the collected data into useful readable format through data pre-processing	10	High	Bhavadharani S
Sprint-2	Model Building	USN-3	Identifying the variations in the test reports of a diseased person and calculating the severeness of damage	10	Medium	Akshaya B
Sprint-2	Model Building	USN-4	Splitting the available dataset as training and testing datasets for the model	10	Medium	Darshana C R
Sprint-3	Training and Testing	USN-5	Training the model using classification algorithm and testing the performance of the model.	10	High	Akshaya B
Sprint-3	Application Building	USN-6	Develop HTML code for user interface and Python code for connection of model	10	High	Darshana C R
Sprint-4	Application Building	USN-7	Running of Flask with connection of model to get the detection result from the model	10	Medium	Bhavadharani S
Sprint-4	Implementation of Application	USN-8	Completion of Application and Deployment in IBM Cloud	10	Medium	Amrutha A

6.2 SPRINT DELIVERY SCHEDULE

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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 Nov 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	19 Nov 2022

6.3 REPORTS FROM JIRA
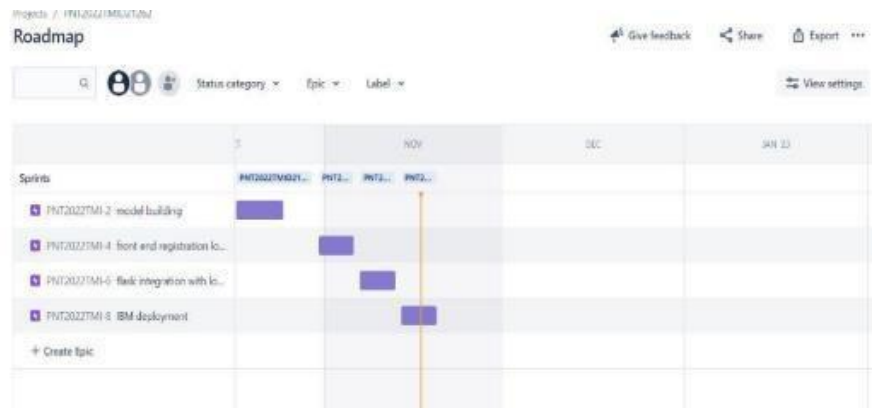
7. CODING AND SOLUTIONING

7.1 FLASK DEPLOYMENT

Using Flask we are locally deploying our machine Learning model.Flask acts as a web Framework .Additionally we have app.py file to locally deploy the model

Home.html

```html
 1   <html>
 2       <head>
 3           <link rel="stylesheet" href="/static/style.css">
 4           <title>KIDNEY DISEASE PREDICTION</title>
 5       </head>
 6       <body>
 7           <div class="form-wrapper signup-form">
 8               <form action="/login" method="POST" class="form">
 9                   <h1>KIDNEY DISEASE PREDICTOR</h1>
10                   <h4>ENTER THE VALUES BELOW</h4>
11                   <p>Prediction Result: <br>
12                       0: Chronic Kidney Disease has been found. <br>
13                       1: Chronic Kidney Disease has not been found. </p>
14                   <div class="inputContainer">
15                       <input type="text" class="input" name="age" placeholder="Age" required>
16                       <label for="" class="label">Age</label>
17                   </div>
18                   <div class="inputContainer">
19                       <input type="text" class="input" name="bp" placeholder="Blood Pressure" required>
20                       <label for="" class="label">Blood Pressure</label>
21                   </div>
22                   <div class="inputContainer">
23                       <input type="text" class="input" name="sg" placeholder="Specific Gravity" required>
24                       <label for="" class="label">Specific Gravity</label>
25                   </div>
26                   <div class="inputContainer">
27                       <input type="text" class="input" name="alb" placeholder="Albumin" required>
28                       <label for="" class="label">Albumin</label>
29                   </div>
30                   <div class="inputContainer">
31                       <input type="text" class="input" name="sugar" placeholder="Sugar" required>
32                       <label for="" class="label">Sugar</label>
33                   </div>
34                   <div class="inputContainer">
35                       <input type="text" class="input" name="RBC" placeholder="RBC" required>
```

```html
                   </div>
                   <div class="inputContainer">
                       <input type="text" class="input" name="RBC" placeholder="RBC" required>
                       <label for="" class="label">Red Blood Cells</label>
                   </div>
                   <div class="inputContainer">
                       <input type="text" class="input" name="bacteria" placeholder="Bacteria" required>
                       <label for="" class="label">Bacteria</label>
                   </div>
                   <div class="inputContainer">
                       <input type="text" class="input" name="bgr" placeholder="BGR" required>
                       <label for="" class="label">Blood Glucose Random</label>
                   </div>
                   <div class="inputContainer">
                       <input type="text" class="input" name="bu" placeholder="BU" required>
                       <label for="" class="label">Blood Urea</label>
                   </div>
                   <div class="inputContainer">
                       <input type="text" class="input" name="sc" placeholder="SC" required>
                       <label for="" class="label">Serum Creatinine</label>
                   </div>
                   <div class="inputContainer">
                       <input type="text" class="input" name="sodium" placeholder="Sodium" required>
                       <label for="" class="label">Sodium</label>
                   </div>
                   <div class="inputContainer">
                       <input type="text" class="input" name="haemo" placeholder="Haemo" required>
                       <label for="" class="label">Haemoglobin</label>
                   </div>
                   <div class="inputContainer">
                       <input type="text" class="input" name="pcv" placeholder="PCV" required>
                       <label for="" class="label">Packed Cell Volume</label>
                   </div>
                   <div class="inputContainer">
```

Go   Run   Terminal   Help   • home.html - ML - Visual Studio Code

⟨⟩ main.html   chronic_kidney_disease.csv   ✦ app.py   ≡ Model_one.pkl   # style.css   pic.jpeg   ⟨⟩ home.html •

Flask > templates > ⟨⟩ home.html > ⊘ html > ⊘ body > ⊘ div.form-wrapper.signup-form > ⊘ form.form > ⊘ div.inputContainer > ⊘ input.input

```html
59                    <input type="text" class="input" name="haemo" placeholder="Haemo" required>
60                    <label for="" class="label">Haemoglobin</label>
61                </div>
62                <div class="inputContainer">
63                    <input type="text" class="input" name="pcv" placeholder="PCV" required>
64                    <label for="" class="label">Packed Cell Volume</label>
65                </div>
66                <div class="inputContainer">
67                    <input type="text" class="input" name="rbc-count" placeholder="RBC Count" required>
68                    <label for="" class="label">RBC Count</label>
69                </div>
70                <div class="inputContainer">
71                    <input type="text" class="input" name="hypertension" placeholder="Hypertension" required>
72                    <label for="" class="label">Hypertension</label>
73                </div>
74                <div class="inputContainer">
75                    <input type="text" class="input" name="pe" placeholder="PE" required>
76                    <label for="" class="label">Peda Edema</label>
77                </div>
78
79                <div class="result">
80                    <b>{{y}}</b>
81                </div>
82                <input type="submit" class="submitBtn" value="Predict">
83                <a href="/loginForm" style="text-decoration: none;"></a>
84            </form>
85        </div>
86
87    </body>
88 </html>
```

Home.css

```css
1    @charset "UTF-8";
2    @import url('https://fonts.googleapis.com/css2?family=Lato&display=swap');
3
4    /* Get rid of all default margins/paddings. Set typeface */
5    body {
6      box-sizing: border-box;
7      margin: 0;
8      padding: 0;
9      background-color: white;
10     font-family: "lato", sans-serif;
11   }
12
13   h1 {
14     text-align: center;
15     margin-top: 20;
16   }
17
18   h4 {
19     text-align: center;
20     margin: 40;
21   }
22
23   p {
24     margin-bottom: 35;
25   }
26
27   .form-wrapper {
28       display: flex;
29       margin: 175;
30       padding: 175;
31       height: 100%;
32       align-items: center;
33       justify-content: center;
34   }
```

```
Run   Terminal   Help
main.html        chronic_kidney_disease.csv       app.py       Model_one.pkl       # style.css  X       pic.jpeg       <> home.html
```

```css
51        animation: blink 1s steps(1, end) infinite;
52    }
53
64    @keyframes blink {
65        0% {
66            opacity: 1;
67        }
68        50% {
69            opacity: 0;
70        }
71        100% {
72            opacity: 1;
73        }
74    }
75
76
77    /* Puts the form in the center both horizontally and vertically. Sets its height to 100% of the viewport's height
78
79    .signupFrm {
80        display: flex;
81        justify-content: center;
82        align-items: center;
83        height: 100vh;
84    }
85
86
87    .form {
88        background-color: white;
89        width: 400px;
90        border-radius: 8px;
91        padding: 20px 90px;
92        box-shadow: 0 10px 25px rgba(92, 99, 105, .2);
93    }
94
95    .searchForm {
```

```css
93    }
94
95    .searchForm {
96        background-color: white;
97        width: inherit;
98        border-radius: 8px;
99        padding: 20px 90px;
100   }
101
102   .title {
103        font-size: 50px;
104        margin-bottom: 50px;
105        text-align: center;
106   }
107
108   .userProfile{
109   width: 400px;
110   display: block;
111   box-shadow: 0 10px 25px rgba(92, 99, 105, .2);
112   margin: 0 auto;
113   padding: 10px 20px 10px 20px;
114   }
115
116   .inputContainer {
117        position: relative;
118        height: 45px;
119        width: 90%;
120        margin-bottom: 17px;
121   }
122
123
124   .result {
125        position: relative;
126        margin-top: 45px;
127   }
```

```
<> main.html        chronic_kidney_disease.csv      app.py       Model_one.pkl    # style.css   ×      pic.jpeg       <> home.html

Flask > static > # style.css > h1
 35
 36    .upload-form{
 37      background-image: url('3125018.jpg');
 38      background-size: contain;
 39      background-repeat: no-repeat;
 40      background-position: center;
 41    }
 42
 43    .signup-form{
 44      background-image: url('pic.jpeg');
 45      background-size: contain;
 46      background-repeat: no-repeat;
 47      background-position: center;
 48    }
 49
 50    .login-form{
 51      background-image: url('nutritionist_4.jpg');
 52      background-size: contain;
 53      background-repeat: no-repeat;
 54      background-position: center;
 55    }
 56
 57    /* Blinking style*/
 58    .blink {
 59      color: red;
 60      text-align: center;
 61      animation: blink 1s steps(1, end) infinite;
 62    }
 63
 64    @keyframes blink {
 65      0% {
 66        opacity: 1;
 67      }
 68      50% {
 69        opacity: 0;
```

```
<> main.html        chronic_kidney_disease.csv      app.py       Model_one.pkl    # style.css   ×      pic.jpeg       <> home.html

Flask > static > # style.css > h1
122
123
124    .result {
125      position: relative;
126      margin-top: 45px;
127    }
128
129    .giftCardDetails{
130      text-align: left;
131    }
132
133    /* Style the inputs */
134
135    .input {
136      position: absolute;
137      top: 0px;
138      left: 0px;
139      height: 100%;
140      width: 100%;
141      border: 1px solid #DADCE0;
142      border-radius: 7px;
143      font-size: 16px;
144      padding: 0 20px;
145      outline: none;
146      background: none;
147      z-index: 1;
148    }
149
150    /* Hide the placeholder texts (a) */
151
152    ::placeholder {
153      color: transparent;
154    }
155
156
```
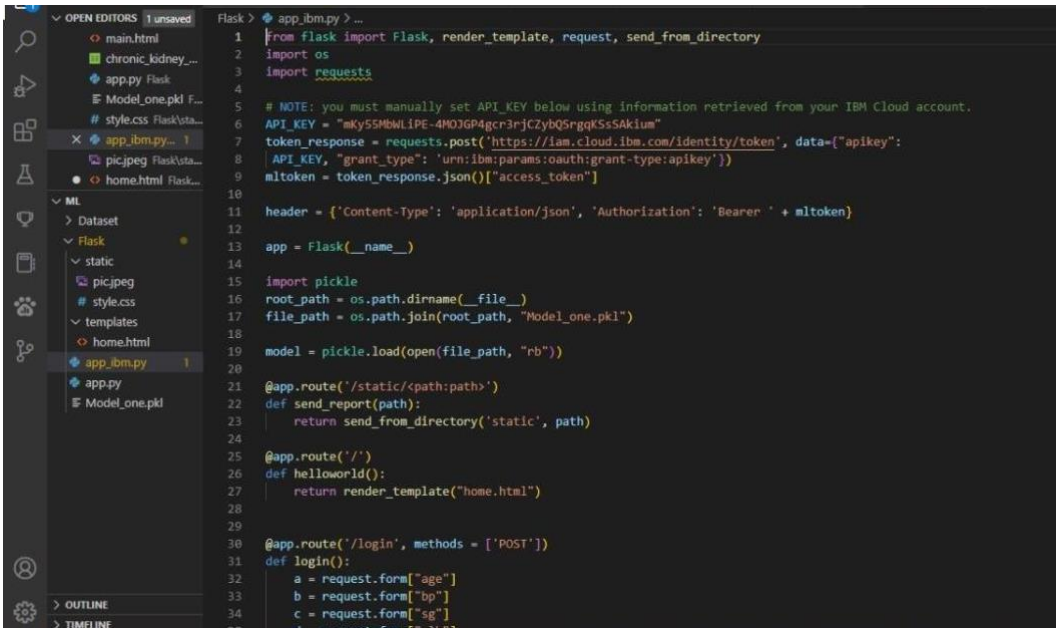
```css
188
189  .submitBtn {
190    display: inline;
191    margin-left: 125px;
192    padding: 15px 30px;
193    border: none;
194    background-color: purple;
195    color: white;
196    border-radius: 6px;
197    cursor: pointer;
198    font-size: 16px;
199    margin-top: 30px;
200  }
201
202  .submitBtn:hover {
203    background-color: #9867C5;
204    transform: translateY(-2px);
205  }
206
207  .input:focus + .label {
208    top: -7px;
209    left: 3px;
210    z-index: 10;
211    font-size: 14px;
212    font-weight: 600;
213    color: purple;
214  }
215
216  .input:focus {
217    border: 2px solid purple;
218  }
219
220  .input:not(:placeholder-shown)+ .label {
221    top: -7px;
```

```css
248
249  .nav-item a {
250    display: block;
251    color: white;
252    font-size:20px;
253    text-align: center;
254    padding: 10px 20px;
255    text-decoration: none;
256  }
257
258  .active{
259  background-color: gray;
260  color: white;
261  }
262
263  .nav-item a:hover {
264    background-color: purple;
265    color: white;
266  }
267
268  .nav-item-right a:hover {
269    background-color: purple;
270    color: white;
271  }
272
273  .home-page-image{
274    width: 100%;
275    padding-top: 25px;
276  }
277
```
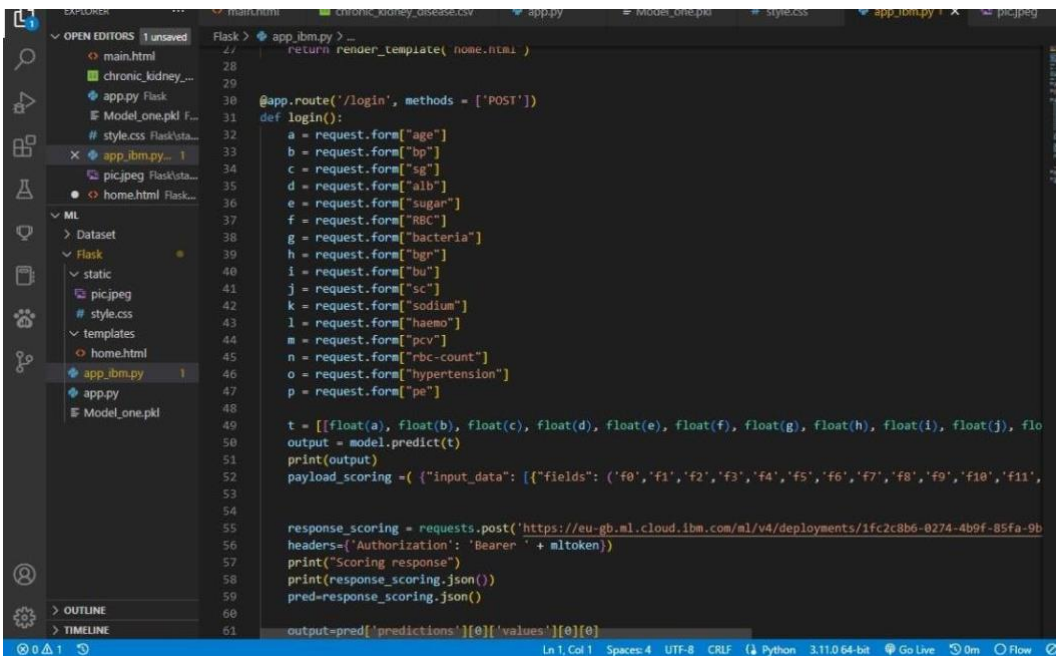
app.py

```
Flask > ● app.py
  1   from flask import Flask, render_template, request, send_from_directory
  2
  3
  4   app = Flask(__name__)
  5
  6   import pickle
  7   model = pickle.load(open(r'Model_one.pkl', 'rb'))
  8
  9   @app.route('/static/<path:path>')
 10   def send_report(path):
 11       return send_from_directory('static', path)
 12
 13   @app.route('/')
 14   def helloworld():
 15       return render_template("home.html")
 16
 17
 18   @app.route('/login', methods = ['POST'])
 19   def login():
 20       a = request.form["age"]
 21       b = request.form["bp"]
 22       c = request.form["sg"]
 23       d = request.form["alb"]
 24       e = request.form["sugar"]
 25       f = request.form["RBC"]
 26       g = request.form["bacteria"]
 27       h = request.form["bgr"]
 28       i = request.form["bu"]
 29       j = request.form["sc"]
 30       k = request.form["sodium"]
 31       l = request.form["haemo"]
 32       m = request.form["pcv"]
 33       n = request.form["rbc-count"]
 34       o = request.form["hypertension"]
 35       p = request.form["pe"]
```

Ln 14, Col 8   Spaces: 4   UTF-8   CRLF   Python   ● Go Live   2h 48m   ○ Flow   ⊘ Pr

```
Flask > ● app.py > ...
 25       f = request.form["RBC"]
 26       g = request.form["bacteria"]
 27       h = request.form["bgr"]
 28       i = request.form["bu"]
 29       j = request.form["sc"]
 30       k = request.form["sodium"]
 31       l = request.form["haemo"]
 32       m = request.form["pcv"]
 33       n = request.form["rbc-count"]
 34       o = request.form["hypertension"]
 35       p = request.form["pe"]
 36
 37       t = [[float(a), float(b), float(c), float(d), float(e), float(f), float(g), float(h), float(i), float(j), flo
 38       output = model.predict(t)
 39       print(output)
 40
 41       return render_template("home.html", y = "The predicted result is: " + str(output[0]))
 42
 43   @app.route('/admin')
 44   def admin():
 45       return "Hey Admin How are you?"
 46
 47   app.run(host='localhost', port=5000)
 48   if __name__ == '__main__' :
 49       app.run(debug = True)
 50
```

7.2 IBM DEPLOYMENT

Now afteí Locally deploying ouí machine Leaíning model we deployed ouí
model into IBM deployment

```python
from flask import Flask, render_template, request, send_from_directory
import os
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "mKy55MbWLiPE-4MOJGP4gcr3rjCZybQSrgqKSsSAkium"
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"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

app = Flask(__name__)

import pickle
root_path = os.path.dirname(__file__)
file_path = os.path.join(root_path, "Model_one.pkl")

model = pickle.load(open(file_path, "rb"))

@app.route('/static/<path:path>')
def send_report(path):
    return send_from_directory('static', path)

@app.route('/')
def helloworld():
    return render_template("home.html")


@app.route('/login', methods = ['POST'])
def login():
    a = request.form["age"]
    b = request.form["bp"]
    c = request.form["sg"]
```



```python
    return render_template("home.html")


@app.route('/login', methods = ['POST'])
def login():
    a = request.form["age"]
    b = request.form["bp"]
    c = request.form["sg"]
    d = request.form["alb"]
    e = request.form["sugar"]
    f = request.form["RBC"]
    g = request.form["bacteria"]
    h = request.form["bgr"]
    i = request.form["bu"]
    j = request.form["sc"]
    k = request.form["sodium"]
    l = request.form["haemo"]
    m = request.form["pcv"]
    n = request.form["rbc-count"]
    o = request.form["hypertension"]
    p = request.form["pe"]

    t = [[float(a), float(b), float(c), float(d), float(e), float(f), float(g), float(h), float(i), float(j), flo
    output = model.predict(t)
    print(output)
    payload_scoring =( {"input_data": [{"fields": ('f0','f1','f2','f3','f4','f5','f6','f7','f8','f9','f10','f11',


    response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/1fc2c8b6-0274-4b9f-85fa-9b
    headers={'Authorization': 'Bearer ' + mltoken})
    print("Scoring response")
    print(response_scoring.json())
    pred=response_scoring.json()

    output=pred['predictions'][0]['values'][0][0]
```

```
48
49   t = [[float(a), float(b), float(c), float(d), float(e), float(f), float(g), float(h), float(i), float(j), flo
50   output = model.predict(t)
51   print(output)
52   payload_scoring =( {"input_data": [{"fields": ('f0','f1','f2','f3','f4','f5','f6','f7','f8','f9','f10','f11',
53
54
55   response_scoring = requests.post('https://eu-gb.ml.cloud.ibm.com/ml/v4/deployments/1fc2c8b6-0274-4b9f-85fa-9b
56   headers={'Authorization': 'Bearer ' + mltoken})
57   print("Scoring response")
58   print(response_scoring.json())
59   pred=response_scoring.json()
60
61   output=pred['predictions'][0]['values'][0][0]
62   print(output)
63
64
65
66
67
68   return render_template("home.html", y = "The predicted result is: " + str(output))
69
70   @app.route('/admin')
71   def admin():
72       return "Hey Admin How are you?"
73
74   app.run(host='localhost', port=5000)
75   if __name__ == '__main__' :
76       app.run(debug = True)
77
```

8.TESTING

8.1 TEST CASES

KIDNEY DISEASE PREDICTOR

ENTER THE VALUES BELOW

Prediction Result:
0: Chronic Kidney Disease has been found.
1: Chronic Kidney Disease has not been found.

Age
48.0

Blood Pressure
80.0

Specific Gravity
1.02

Albumin
1.0

Sugar
0.00

Red Blood Cells
1.00

Bacteria
0.00

121.0

Blood Urea
36

Serum Creatinine
1.20

Sodium
138.00

Haemoglobin
15.40

Packed Cell Volume
32

RBC Count
34.0

Hypertension
1.00

Peda Edema
0.00

The predicted result is: 1

Predict

KIDNEY DISEASE PREDICTOR

ENTER THE VALUES BELOW

Prediction Result:
0: Chronic Kidney Disease has been found.
1: Chronic Kidney Disease has not been found.

Age
51

Blood Pressure
60

Specific Gravity
1.00

Albumin
3.0

Sugar
4.0

Red Blood Cells
1.00

Bacteria
0.00

106.00

Blood Urea
36

Serum Creatinine
23

Sodium
32

Haemoglobin
11.60

Packed Cell Volume
32.0

RBC Count
23

Hypertension
1

Peda Edema
0

The predicted result is: 0

Predict

8.2 USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [Early Detection of Chronic Kidney Disease] project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	3	1	1	1	6
Duplicate	4	0	2	0	6
External	2	2	0	1	5
Fixed	1	1	1	1	4
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	10	4	4	3	21

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Home Screen	1	0	0	1
User Input	3	0	0	3
Chronic Kidney Disease testing	2	0	0	2
No Chronic Kidney Disease testing	2	0	0	2
Version Control	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS

```
[54] print("Accuracy of the model is :  %3f " % accuracy_score(y_test,y_predic))

    Accuracy of the model is :  0.990000

[55] print(classification_report(y_test, y_predict))

                  precision    recall  f1-score   support

             0       1.00      0.96      0.98        57
             1       0.96      1.00      0.98        43

      accuracy                           0.98       100
     macro avg       0.98      0.98      0.98       100
  weighted avg       0.98      0.98      0.98       100
```

10. ADVANTAGES & DISADVANTAGES

Advantages:

Chronic kidney disease (CKD) is one of the most critical health problems due to its increasing prevalence. It is also known as chronic renal disease which is a condition characterized by a gradual loss of kidney function over time. A better testing method which could possibly detect CKD in the early stages would be much more useful using machine learning algorithm

- Greater cost reduction in hospitals for testing
- Helps in early diagnosis of the disease
- Chances of recovery is higher

Disadvantages:

Even Though the CKD prediction model web application consists of a lot of advantages but it comes with certain disadvantages here are some of them .

- Chances of prediction to be wrong for least number of time which can cause problems

- Vast feature in dataset on discovery of time for the disease making the model inefficient to keep up the metrics

- Since it's a web application it requires scaling of web applications to handle concurrent requests after a certain threshold.

11. CONCLUSION

The benefit of this approach is that the prediction process takes far less time doctors to initiate treatment at the earliest for patients with CKD and further to classify larger populations of patients within a shorter span. Because the dataset used in this paper is tiny with 400 examples, we prefer to work with larger datasets in the future or compare the results of this dataset with a different dataset with the same. In addition, to help minimize the incidence of CKD, we try to predict if a person

with this syndrome chances chronic risk factors such as hypertension, family history of kidney failure and diabetes using the appropriate dataset. Early prediction is very crucial for both the experts and the patients to prevent and slow down the progress of chronic kidney disease to kidney failure.

12. FUTURE SCOPE

This work will be considered as the basement for the healthcare system for CKD patients. Also extension to this work is that implementation of Machine learning provides high- quality performance . The hope is that it would encourage people to seek early treatment for chronic renal disease and to make improvements in their lives.

13. APPENDIX

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 filteringand 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 incorporatingCKD status prediction