

Early Detection Of Chronic Kidney Disease Using Machine Learning

ABSTRACT:

Every year, an increasing number of patients are diagnosed with late stages of renal disease. Chronic Kidney Disease, also known as Chronic Renal Disease, is characterized by abnormal kidney function or a breakdown of renal function that progresses over months or years. Chronic kidney disease is often found during screening of persons who are known to be at risk for kidney issues, such as those with high blood pressure or diabetes, and those with a blood family who has chronic kidney disease (CKD). As a result, early prognosis is critical in battling the disease and providing effective therapy. Only early identification and continuous monitoring can avoid serious kidney damage or renal failure. Machine Learning (ML) plays a significant part in the healthcare system, and it may efficiently aid and help with decision support in medical institutions. The primary goals of this research are to design and suggest a machine learning method for predicting CKD. Support Vector Machine (SVR), Random Forest (LR), Artificial Neural Network (ANN), and Decision Tree are four master teaching methodologies investigated (DT). The components are built using chronic kidney disease datasets, and the outcomes of these models are compared to select the optimal model for prediction

1. INTRODUCTION

1.1 Project Overview

A kind of artificial intelligence is machine learning (ML) (AI). Its heart is algorithmic procedures, which allow the machine to solve issues without the need for specialist computer programming. The widespread use of ML in the medical industry promotes medical innovation, lowers medical expenses, and improves medical quality. However, further research on using ML to solve clinical problems in nephrology is needed. Understanding the aim and technique of ML application, as well as the present As the technologies are continuously improving, aviation systems have begun adopting smart technologies to develop unmanned aerial vehicles (UAVs) equipped with cameras, which can reach distant areas to identify aftereffects of natural disasters on human life, infrastructure, and transmission lines by capturing images and videos. Data acquired from these UAVs helps to identify the facial expressions of victims, the intensity of their situation and their needs in a post disaster scenario. It helps to take actions and carry out necessary operations to tackle devastating scenarios. Raw images obtained from camera-equipped UAVs are processed and neural network-based feature extraction techniques are applied to analyze the intensity. State of its use in nephrology, is required to properly address and overcome these issues. Machine learning has previously been used to identify human body state, evaluate disease-related aspects, and diagnose a variety of disorders. The term machine learning (ML) is very popular these days, and a lot of clinical prediction model studies have employed this type of technology. While the capacity to capture vast volumes of information on individual patients is transforming the healthcare business, the enormous volume of data being gathered is impractical for humans to comprehend. Machine learning allows healthcare practitioners to advance toward individualized care, often known as precision medicine, by automatically finding patterns and reasoning about data. The integration of machine learning, health informatics, and predictive analytics provides prospects to alter clinical decision support systems and assist improve patient outcomes. Chronic Kidney Disease refers to the kidneys' inability to fulfil their normal blood filtration role and other functions (CKD). The term "chronic" refers to the progressive deterioration of kidney cells over time. This is a severe renal failure in which the kidney no longer filters blood and there is a significant fluid accumulation in the body. This causes an abnormally high level of potassium and calcium salts in the body. High quantities of these salts in the body cause a variety of additional problems. The primary function of the kidneys is to filter excess water and wastes from the blood. This mechanism must work properly to balance the salts and minerals in our bodies. The

proper salt balance is required to manage blood pressure, activate hormones, and create red blood cells, among other things. A high calcium concentration causes bone problems and cystic ovaries in women. CKD can also cause a sudden sickness or an allergy to specific medications. Acute is the medical term for this condition

1.2 PURPOSE

Non communicable illnesses are the leading cause of early death, and CKD is the leading non communicable disease. Chronic Kidney Disease is a major concern for the global health care system. People with CKD must focus on implementing proven, costeffective therapies to as many people as possible while taking into consideration restricted needs, human and financial resources. Chronic kidney disease (CKD) is now wreaking havoc on society and is spreading at an alarming rate. Various efforts have been undertaken to advance early therapy to prevent the condition from progressing to chronic disease. Recent research suggests that some of the negative outcomes can be avoided with early identification and treatment.

2 .LITERATURE SURVEY

2.1 EXISTING SYSTEM

In the paper ‘Landslide Detection Using Random Classifier’ [1], the author Meylin Herrera aims to develop an automatic landslide detection method from satellite image in combination with machine learning. The method consist of cloud-free images and determination of suitable feature for image segmentation and image classification. For image segmentation the method uses two approaches, first approach involves over segmentation of dataset and second involves algorithm using NDVI. The use of platforms such as Google Earth Engine (GEE) allows public access to dataset .The method uses entirely open source technology allowing its applicability and reusability.

In the paper ‘Developing a Hybrid Model For Disaster Prediction Using Machine Learning’ [4], the author develop a hybrid natural disaster predicting model to predict the upcoming disaster scenario. The model uses artificial neural network (ANNs) as a machine learning technique for prediction purposes. The training dataset of ANNs includes historic natural disaster record and meteorological data. To build the model appropriate data mining and machine learning techniques are tested to enhance the accuracy and reliability of the prediction. In the paper ‘Average Rolling based Real time Calamity Detection using Deep Learning’ [9], the author develops a natural disaster detection system using satellite images and generate an alert automatically. The system can be implemented first, to process satellite image to prediction purposes and second to process satellite images for quick detection of affected regions during natural calamities. The system use satellite camera to predict the disaster. The proposed system can help the rescue community to predict the disaster in advance and to detect natural calamity such as landslides, flood etc. This system helps the research community to predict disaster in advance and to optimize the damages during disaster.

LITERATURE SURVEY

The kidneys are positioned in the abdominal cavity, on each side of the spine. They generally weigh around 5 times their body weight yet receives only 20% of the blood flow from the heart. The urine generated by each kidney drains into the urinary bladder, which is positioned in the pelvic area, via a distinct urethra. The kidney is the most essential organ in the human body because it manages fluid levels, electrolyte balance, and other elements that maintain the body's internal environment stable

and comfortable. Kidney diseases are conditions that impact the kidney's functioning. Renal disorders can lead to kidney failure in its advanced stages. Kidney diseases are conditions that impact the kidney's functioning. Kidneys can be injured, which means they can't accomplish what they should. This is known as chronic kidney disease (CKD). Anyone can get chronic kidney disease. In medical research, nephrologists primarily employ two primary tests to identify CKD. A blood test to determine glomerular filtration rate (GFR) and a urine test to determine albumin [1]. Genetics, hypertension, diabetes, obesity, age, and other factors can all have an impact on CKD. Important facts and advancements concerning CKD are described in worldwide kidney disease development guidelines and standard foundations such as the US National Kidney Foundation Kidney Disease Outcomes Quality Initiative (KDOQI) and KDIGO (Kidney Disease Improving Global Outcome). The renal patient is detected by two tests, according to the KDIGO CKD and English National Institute for Health and Care Excellence (NICE) CKD recommendations, these are blood tests to examine how well the kidneys filter the blood to remove creatinine, a normal muscle breakdown byproduct. In comparison, a urine test will show that protein is still present in the urine. Protein (albumin) is a blood component that is normally not transferred into the urine by the kidney filter. When albumin is detected in the urine, it shows that the kidney filters are compromised and may indicate chronic renal disease. Chronic Kidney Disease (CKD) is defined as kidney damage or a glomerular filtration rate (GFR) of less than 60 ml/min/1.73 m² for more than 3 months with serious health consequences.

Nephrologists in medical research generally use two major tests to diagnose CKD.

GFR is determined by a blood test, whereas albumin is determined by a urine test. CKD can be influenced by genetics, hypertension, diabetes, obesity, age, and other variables. Important CKD facts and developments are given in international kidney disease development recommendations and standard foundations such as the US National Kidney Foundation Kidney Disease Outcomes Quality Initiative (KDOQI) and KDIGO (Kidney Disease Improving Global Outcome). According to the KDIGO CKD and English National Institute for Health and Care Excellence (NICE) CKD guidelines, the renal patient is recognized by two blood tests that measure how well the kidneys filter the blood to eliminate creatinine, a natural consequence of muscle breakdown. A urine test, on the other hand, will reveal that protein is still present in the urine. Protein (albumin) is a blood component that is generally not excreted by the kidney filter. When albumin is found in the urine, it indicates that the kidney filters are faulty and may suggest chronic renal illness. Chronic Kidney Disease (CKD) is defined as kidney damage or a glomerular filtration rate (GFR) of less than 60 ml/min/1.73 m² for more than 3 months, which has major health repercussions. N.A. Almansour et al. intend to aid in the

prevention of CKD by using machine learning techniques to detect CKD at an early stage, focusing on applying multiple machine learning classification algorithms to a dataset of 400 patients and 24 CKD-related variables. As classification algorithms, they employ Artificial Neural Network and support vector machine, and any missing values in the dataset are replaced by the relevant characteristics. The best collected parameters and characteristics were used to construct the final model of the two suggested strategies. Experiment results showed that ANN performed better than SVM, with accuracy of 99.75 percent and 97.75 percent, respectively.

2.1 REFERENCES:

- 1) G. Chen et al., “Prediction of Chronic Kidney Disease Using Adaptive Hybridized Deep Convolutional Neural Network on the Internet of Medical Things Platform,” IEEE Access, vol. 8, pp. 100497–100508, 2020, doi: 10.1109/ACCESS.2020.2995310.
- 2) P. T. Coates et al., “KDIGO 2020 Clinical Practice Guideline for Diabetes Management in Chronic Kidney Disease,” Kidney Int., vol. 98, no. 4, pp. S1– S115, 2020, doi: 10.1016/j.kint.2020.06.019.
- 3) L. Chen, “Overview of clinical prediction models,” Ann. Transl. Med., vol. 8, no. 4, pp. 71–71, 2020, doi: 10.21037/atm.2019.11.121.
- 4) H. Kriplani, B. Patel, and S. Roy, Prediction of chronic kidney diseases using deep artificial neural network technique, vol. 31. Springer International Publishing, 2019.
- 5) T. O. Ayodele, “Atherosclerotic Cardiovascular Disease,” Atheroscler. Cardiovasc. Dis., 2012, doi: 10.5772/7111.

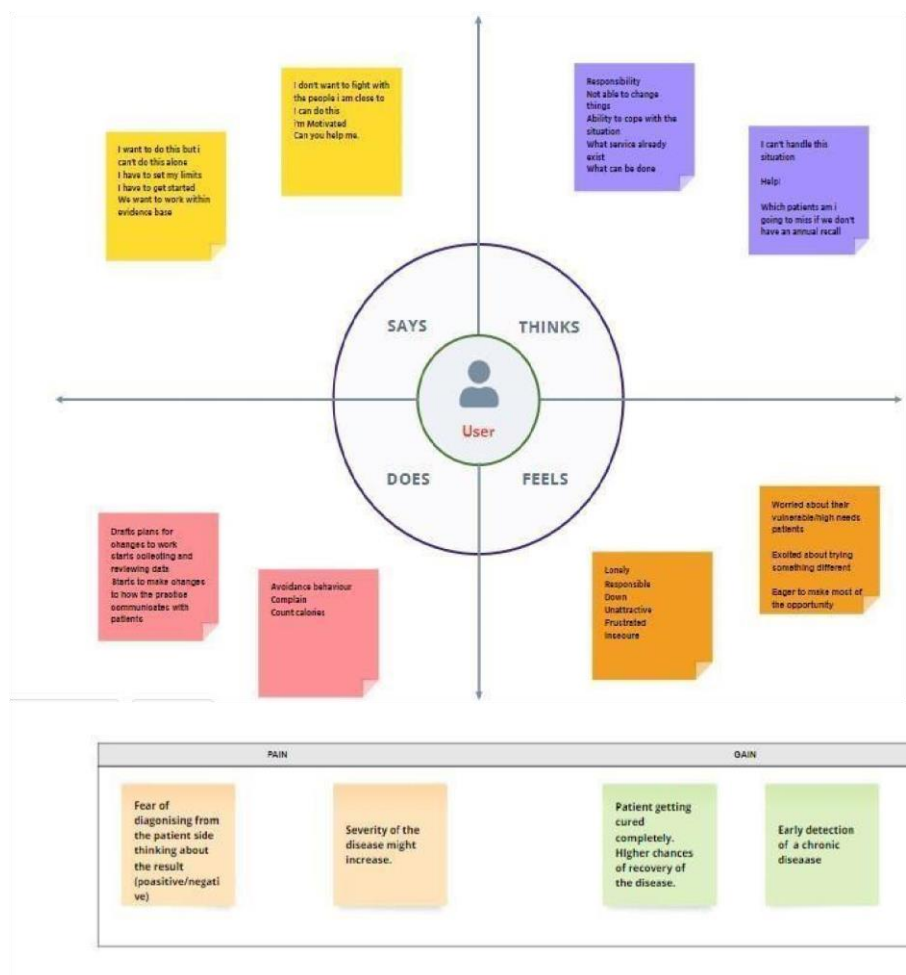
- 6) Abdi, “Three types of Machine Learning Algorithms List of Common Machine Learning Algorithms,” no. November 2016, doi: 10.13140/RG.2.2.26209.10088.
- 7) S. Y. Yashfi et al., “Risk Prediction of Chronic Kidney Disease Using Machine Learning Algorithms,” 2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020, 2020, doi: 10.1109/ICCCNT49239.2020.9225548.
- 8) N. A. Almansour et al., “Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study,” *Comput. Biol. Med.*, vol. 109, no. October 2018, pp. 101–111, 2019, doi:10.1016/j.combiomed.2019.04.017.

3. IDEATION & PROPOSED SOLUTION


3. PROPOSED SYSTEM

The main intention is to detect natural disaster from satellite image. The process involves testing and validation of dataset. This includes image pre-processing, image segmentation and image classification. The pre-processing involves extraction of terrain image using Google Earth Engine. The image segmentation is implemented using k-means and classification algorithms. The image classification involves training dataset and testing

3.1 EMPATHY MAP CONVAS



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.

[Open article](#) →

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

QUESTION

How might we [your problem statement]?

Key rules of brainstorming
To run a 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 collect a sticky note and let the group decide on sticky notes to start drawing!

Aswath Selvam

collection of liver related samples and blood serotypes

ordering the collected samples on data sheets

preparing the data sheets

splitting the dataset

Periyakaruppan

include scikit learn and use ML algos

calculate accuracy and predict scores

Based on requirement, update dataset

calculating accuracy and predicting scores

Finally, deploy model

Venkatesh

By plotting inputs, visualize the data

Validate and evaluate model

Train the model

Yashwant

Based on test samples, graph the data

Analyze the result

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 can break it up into smaller sub-groups.

⌚ 20 minutes

Data collection and processing

collection of liver related samples and blood samples

preparing the data sheets

splitting the dataset

Visualizing the data by plotting inputs

Datasets and model

Based on requirement, update dataset

include scikit learn and use ML algos

models like KNN, etc

Training the model

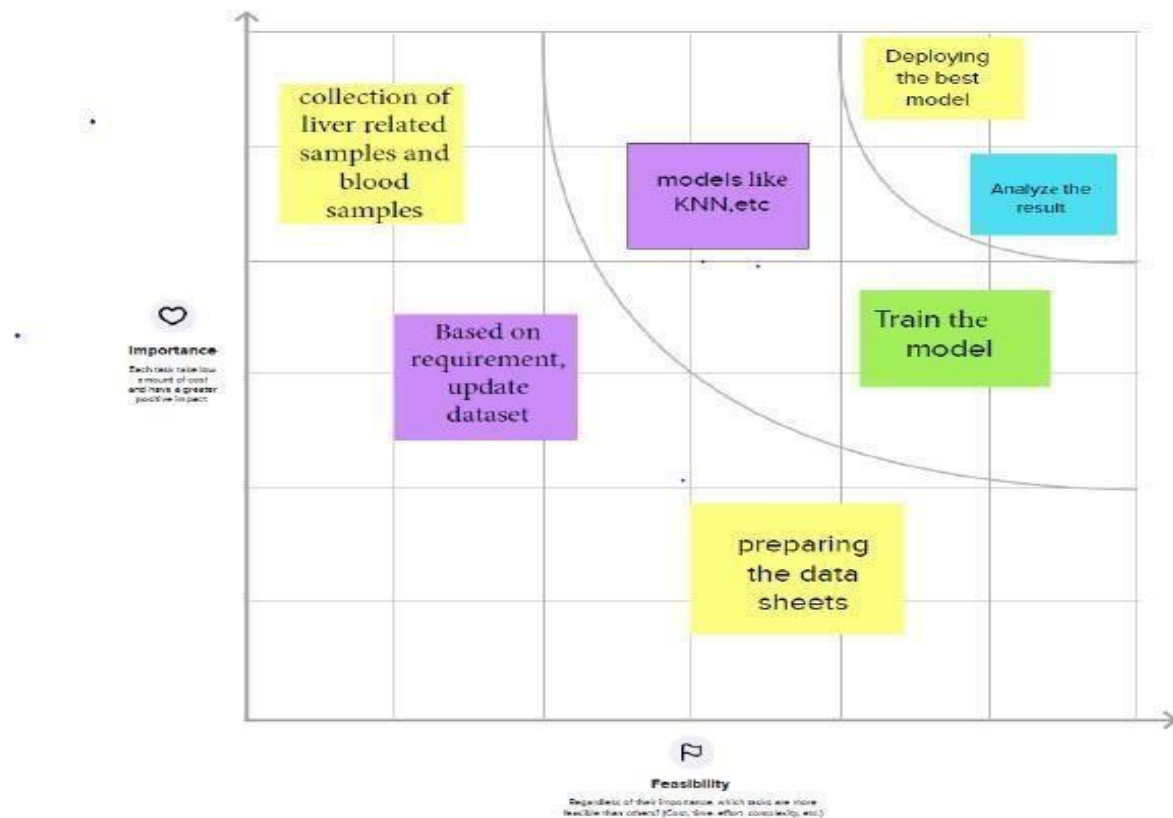
Analyzing and Deployment

calculating accuracy and predicting scores

Analyze the result

Based on test samples, graph the data

Deploying the best model



3.3 PROPOSED SOLUTION

The purpose of this tool is to provide a structured process for identifying a problem, understanding the root causes, ascertaining solution steps, and progress monitoring. With a solution template, you can organize development content that you want to reuse for customer-specific solutions.

Solution templates enable you to easily start the development of customerspecific solutions, for example, for a specific industry.

The term business model refers to a company's plan for making a profit. It identifies the products or services the business plans to sell, its identified target market, and any anticipated expenses. Business models are important for both new and established businesses.

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Early Detection of Chronic Kidney Disease using Machine Learning
2.	Idea / Solution description	By the use of a hybrid model that perfectly unites different computations, processes, or procedures from equivalent or different spaces of data or areas of usage fully intended to enhance each other
3.	Novelty / Uniqueness	Hard and soft exudates, as well as other diverse situations like haemorrhage and microaneurysms individually, are not Distinguished by any system. Models like RESNET-50, Xception etc.,
		which are pre-trained and are highly complex.
4.	Social Impact / Customer Satisfaction	<ul style="list-style-type: none"> • Early detection of the disease • Efficient prediction mechanism with faster results. • Easy to use and understand
5.	Business Model (Revenue Model)	<ul style="list-style-type: none"> • Early detection of the disease • Efficient prediction mechanism with faster results. • Easy to use and understand
6.	Scalability of the Solution	The model is scalable from the architecture and dataset training perspective. We can train huge amounts of image data by converting them into .npy / .npz file format which would

		facilitate easy storing, retrieving and processing.
--	--	---

PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem.

Problem-Solution Fit - this occurs when you have evidence that customers care about certain jobs, pains, and gains. At this stage you've proved the existence of a problem and have designed a value proposition that addresses your customers' jobs, pains and gains.

Problem-Solution fit canvas 2.0			Early Detection of Chronic Kidney Disease Using Machine Learning		
Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Patients who are facing issues related to kidneys. Elderly people, are more prone to get kidney disease. Diabetic Patients Alcoholic addicted Patients		6. CUSTOMER CONSTRAINTS CC Patients are afraid about risk of using new technology They are limiting themselves as they are not aware of the test accuracies		5. AVAILABLE SOLUTIONS AS Currently in the Medical field, the tests that are performed to detect chronic kidney disease are: 1. Ultra Sound Scan 2. MRI Scan 3. CT Scan
	2. JOBS-TO-BE-DONE / PROBLEMS J&P Problems related to identifying the chronic kidney disease Accuracy of patients test results Time taken to produce test results		9. PROBLEM ROOT CAUSE RC The root cause of the problem is inaccurate results. The test takes much time to evaluate the results.		7. BEHAVIOUR BE They take costly Scans because they had no other choice. They blindly trust the inaccurate test results and become more anxious and sad.
Identify strong TR & EM	3. TRIGGERS TR Their dilemma or confusion of whether they really have chronic kidney disease or not!		10. YOUR SOLUTION SL Predicts Faster and accurately. Time and Cost of Test is drastically reduced Helps to take treatment at right time.		8. CHANNELS of BEHAVIOUR CH They consider taking tests costing lower from any of the online labs.
	4. EMOTIONS: BEFORE / AFTER EM BEFORE: Anxious about their medical condition. AFTER: Determined and able to follow doctor's advice on hat to do next to improve their condition				6.3 OFFLINE They take many tests in offline labs and wait for enormous time to gets results

4. REQUIREMENT ANALYSIS

4.1 SOFTWARE IMPLEMENTATION

- **Django:**

Front End User is requesting the image of the location. Based on the entered location the satellite image is collected for prediction.

- **Tensor flow:**

Back End The fetched image is analysed based on the factors slope, NDVI, etc. and these images are processed for further scanning for predicting the output. It is passing back the API into Django.

- **SQLite:**

Database The image analysis and prediction is managed by the database. It involves analysis of database and updation of database. Using the satellite image analyse the database based on API and allows data handling among several datasets.

- **Google Earth Engine (GEE)** The Google Earth Engine is used to pre-process the dataset. It is a cloud based platform and its parallel processing capacity makes it efficient to run spatial reduction over large image collections. It also helps in analysis of image sets of pre and post events.

- **Python** It is used for the processing and visualization of the data. It is used for initial segmentation and generation of attribute tables for data processing

4.2 Non-Functional requirements

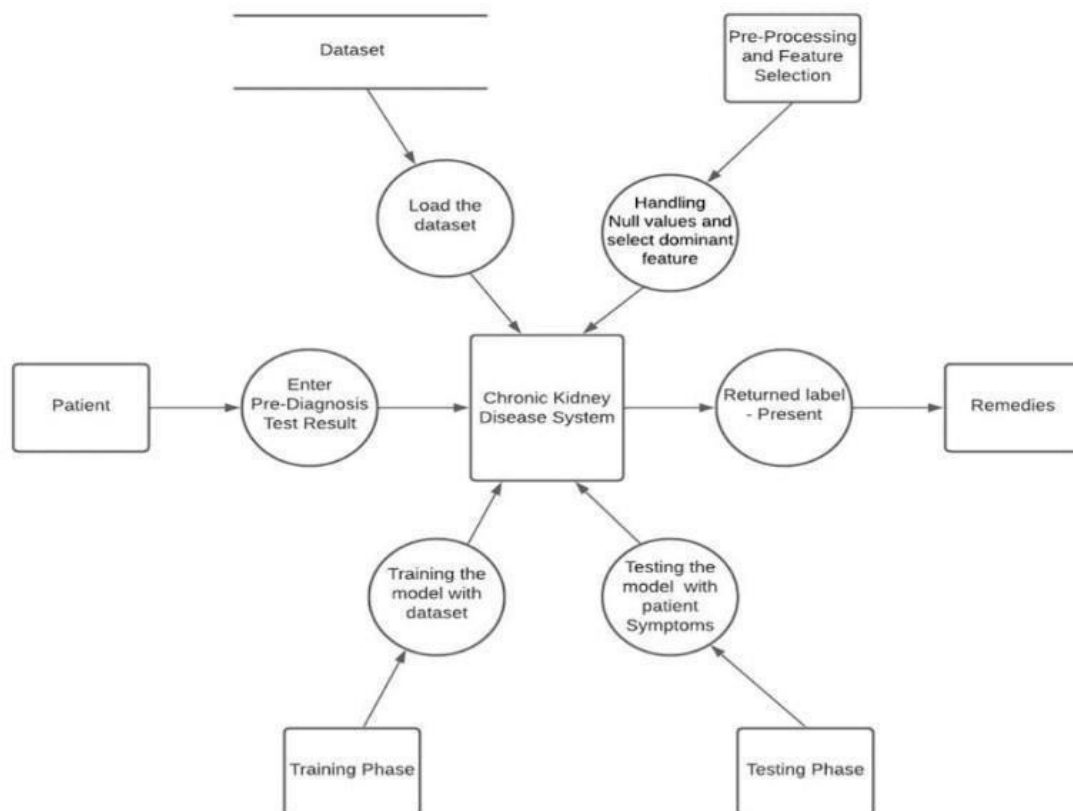
HARDWARE IMPLEMENTATION

- 1 TB Hard disk
- 8 GB RAM
- Processor Intel i5 or above
- CPU speed 2 GHz

5. PROJECT DESIGN

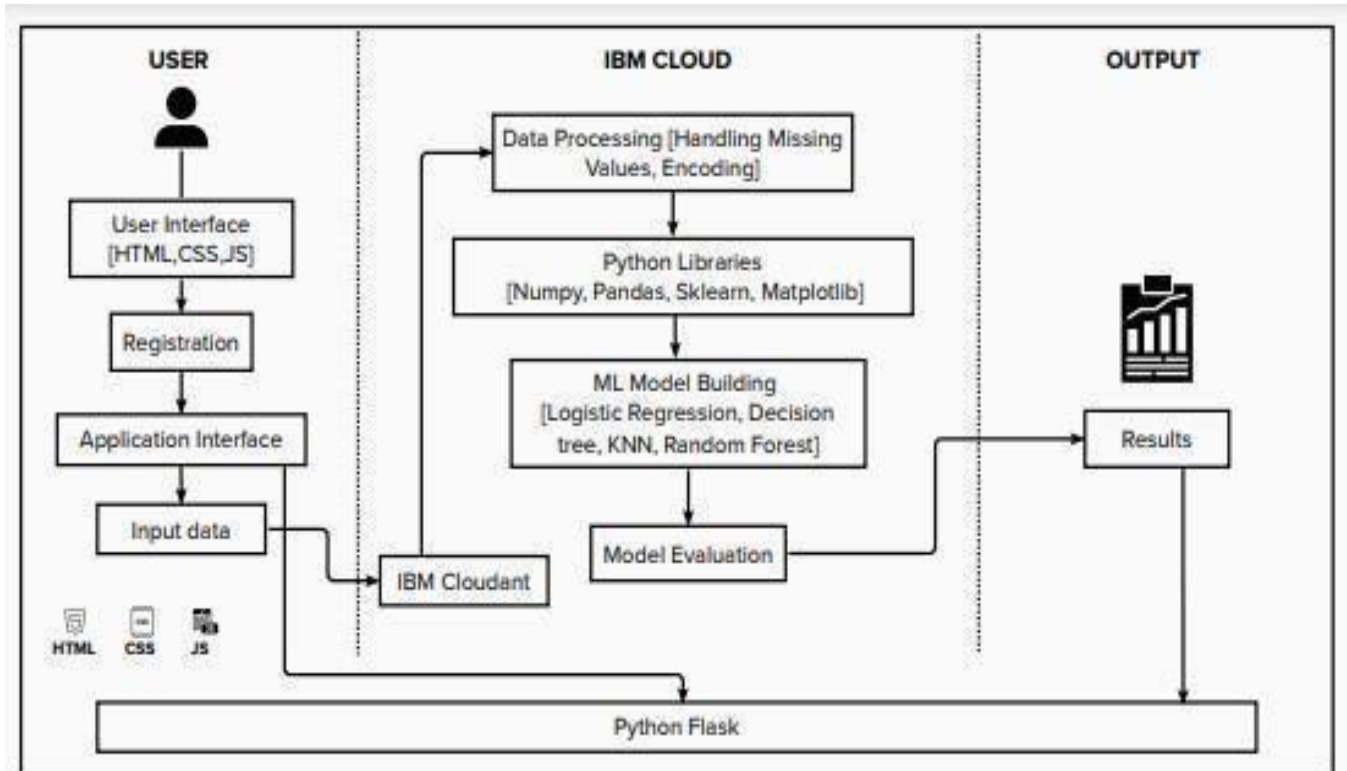
5.1 DATA FLOW DIAGRAM

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.

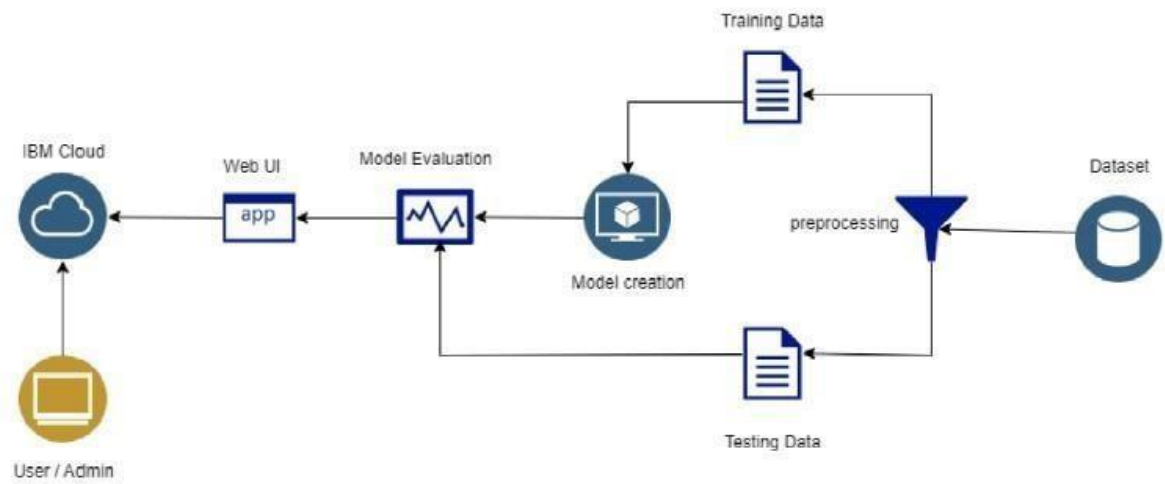


5.2 SOLUTION AND TECHNICAL ARCHITECTURE

S .No	Component	Description	Technology
1.	User Interface	An Interface for the user to interact with the prediction model.	HTML, CSS, JavaScript
2.	User Registration	User can register in the web application	HTML forms



TECHNICAL ARCHITECTURE



USER STORIES

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the diagnosis tool using my email and password	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email on registering for the diagnosis tool	I will receive confirmation email	High	Sprint-1
		USN-3	As a user, I can register for the application through my Gmail	I can register and access the dashboard with my Gmail Login	Low	Sprint-4
	Login	USN-4	As a user, I can log into the application by entering my credentials	I can login and access past records	High	Sprint-1
	Dashboard	USN-5	As a user, I can see my past records and activities	I can access the functionalities diagnosing tool	High	Sprint-3
	Entry form	USN-6	As a user, I must enter my pre-diagnostic test results	I can use the form to input test results	High	Sprint-2
	Report	USN-7	As a user, I can view the report generated by the tool	I can view negative/ positive results produced after diagnosis	High	Sprint-3
	Remedies	USN-8	As a user, I will receive remedies to treat my symptoms	I can cure my symptoms with the remedies suggested	Medium	Sprint-3
Customer Care Executive	Queries	USN-9	As a customer care executive, I must assist users that face problems through Q&A	I will provide 24/7 support for the tool	Low	Sprint-4
	Feedback	USN-10	As a customer care executive, I should get input for the tool's enhancement from users	I must work on improving tool's performance	Low	Sprint-4
Administrator	Feature importance	USN-11	As an administrator, I should identify the most significant factors that lead to CKD based on the present trend	I must identify important features	High	Sprint-2
	Train model	USN-12	As an administrator, I must use the most suitable ML model for detection of CKD	I should efficiently train the ML model	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING

Sprints are the backbone of any good Agile development team. And the better prepared you are before a sprint, the more likely you are to hit your goals. Spring planning helps to refocus attention, minimize surprises, and (hopefully) guarantee better code gets shipped. The main event during agile methodology is the sprint, the stage where ideas turn into innovation and valuable products come to life. On one hand, agile sprints can be highly effective and collaborative. At the same time, they can be chaotic and inefficient if they lack proper planning and guidance. And for this reason, making a sprint schedule is one of the most important things you can do to ensure that your efforts are successful.

6.2 SPRINT DELIVERY SCHEDULE

Product Backlog, Sprint Schedule, and Estimation

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	Registration	USN-1	New user enters into the System He/ She can register into the Application by entering mail Id and Password.	8	High	FATHIMA AMANI LAVANYA MALINI SHRUTHI
Sprint-1		USN-2	The user will receive conformation Email	5	High	FATHIMA AMANI LAVANYA MALINI SHRUTHI

Sprint-1	Login	USN-3	After Successful registration the user can Log into the application by entering the registered Mail Id and Password	8	High	FATHIMA AMANI LAVANYA MALINI SHRUTHI
----------	-------	-------	---	---	------	---

Sprint-2	Dashboard	USN-4	User can get into the Dashboard only when the Verification Successful. After the user can access the displayed information in the Dashboard.	8	High	FATHIMA AMANI LAVANYA MALINI SHRUTHI
Sprint-2	Diagnosis Form	USN-5	As a user, I must enter my prediagnostic test results to give as required.	10	High	FATHIMA AMANI LAVANYA MALINI SHRUTHI
Sprint-3	Report	USN-6	As a user, I can view the report generated by the tool	7	High	FATHIMA AMANI LAVANYA MALINI SHRUTHI

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Quality Assurance	USN-8	As a user they have some credibility issues while using application.	8	High	FATHIMA AMANI LAVANYA MALINI SHRUTHI
Sprint-3	Train Model	USN-9	As an administrator, I must use the most suitable ML model for detection of CKD	9	High	FATHIMA AMANI LAVANYA MALINI SHRUTHI

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

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

Velocity:

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

Sprint 1 AV = Sprint duration/velocity = $13/6 = 2.16$

Sprint 2 AV = Sprint duration/velocity = $8/6 = 1.33$

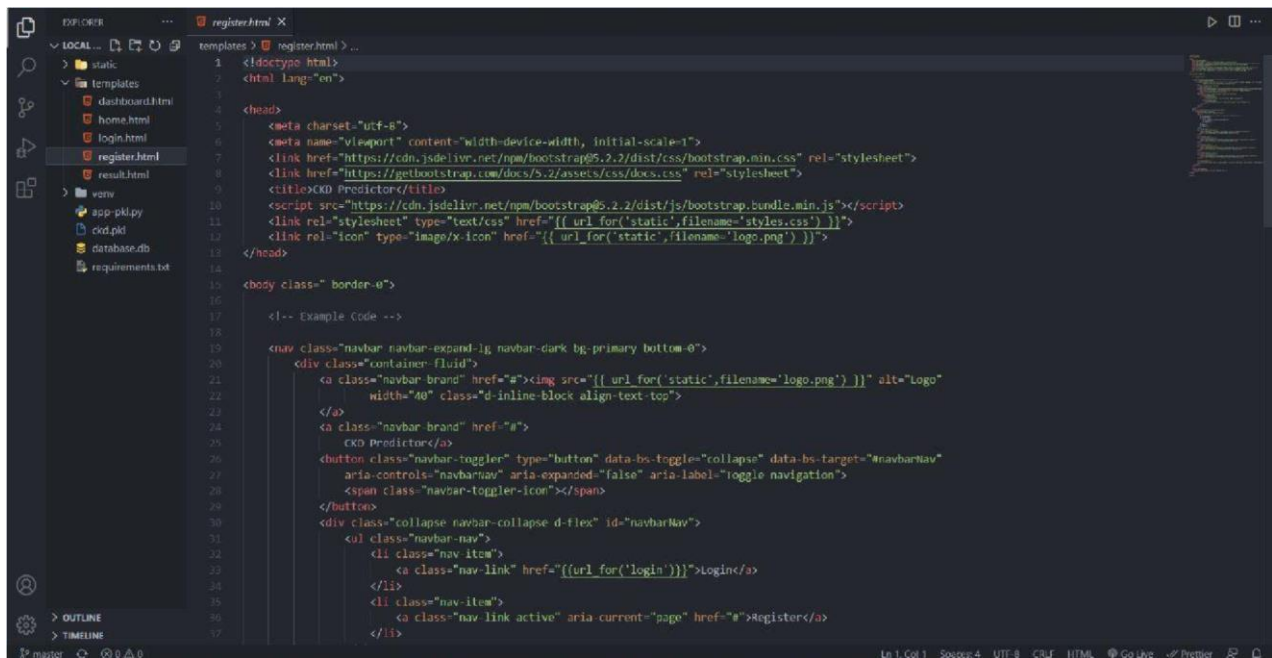
Sprint 3 AV = Sprint duration/velocity = $11/6 = 1.83$

Sprint 4 AV = Sprint duration/velocity = $5/6 = 0.83$

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

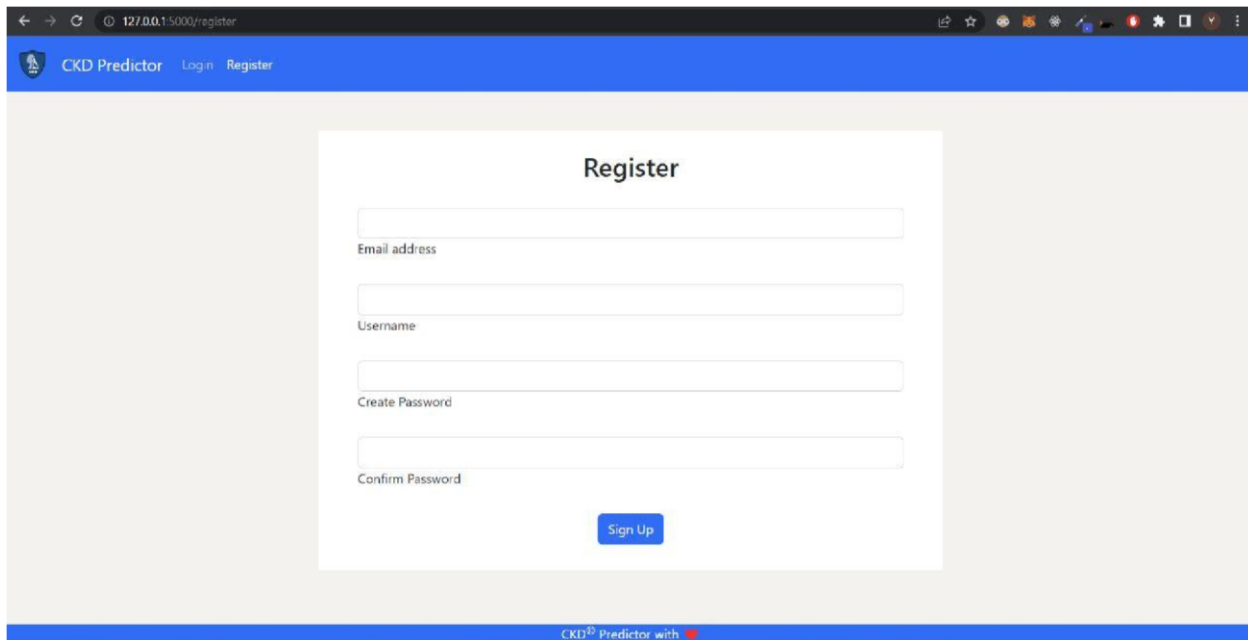
7.1 User Registration and login

Register.html:



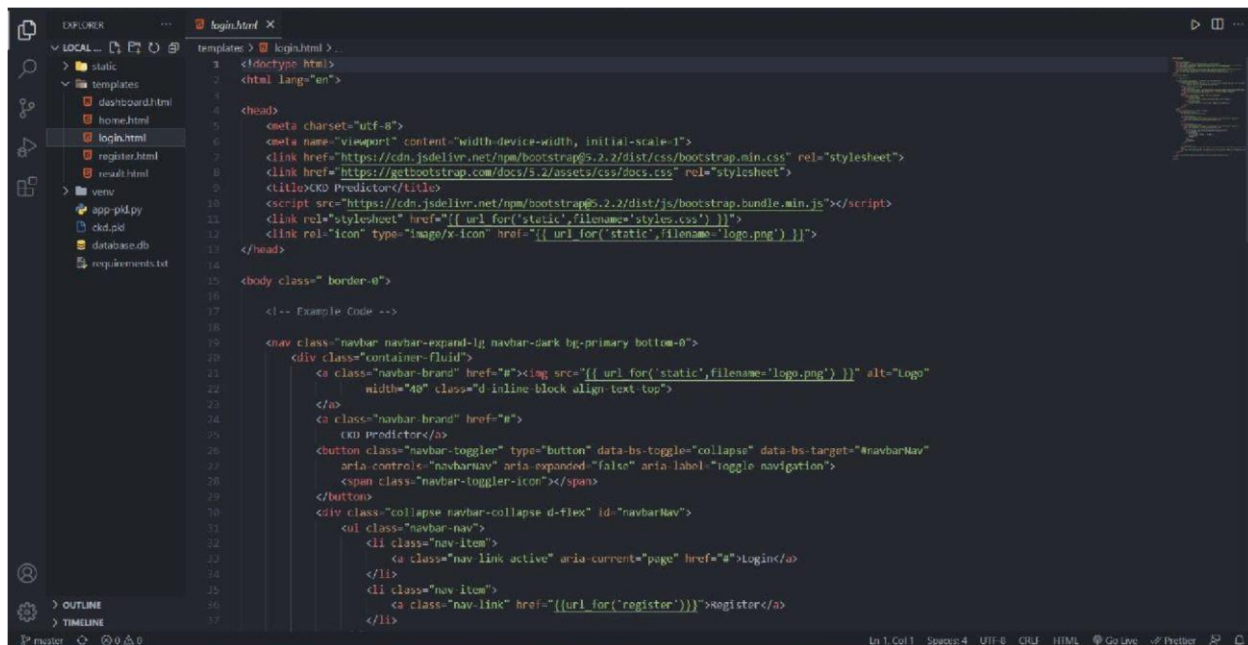
```
1 <doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
8   <link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">
9   <title>OCD Predictor</title>
10  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>
11  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='styles.css') }}">
12  <link rel="icon" type="image/x-icon" href="{{ url_for('static', filename='logo.png') }}">
13 </head>
14
15 <body class="border-0">
16
17   <!-- Example Code -->
18
19   <nav class="navbar navbar-expand-lg navbar-dark bg-primary bottom-0">
20     <div class="container-fluid">
21       <a class="navbar-brand" href="#">
23     </a>
24       <a class="navbar-brand" href="#">
25         OCD Predictor</a>
26       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
27         aria-controls="navbarNav" aria-expanded="false" aria-label="toggle navigation">
28         <span class="navbar-toggler-icon"></span>
29     </button>
30       <div class="collapse navbar-collapse d-flex id="navbarNav">
31         <ul class="navbar-nav">
32           <li class="nav-item">
33             <a class="nav-link" href="{{ url_for('login') }}">Login</a>
34           </li>
35           <li class="nav-item">
36             <a class="nav-link active" aria-current="page" href="#">Register</a>
37           </li>
38         </ul>
39       </div>
40     </div>
41   </nav>
```


Log in page



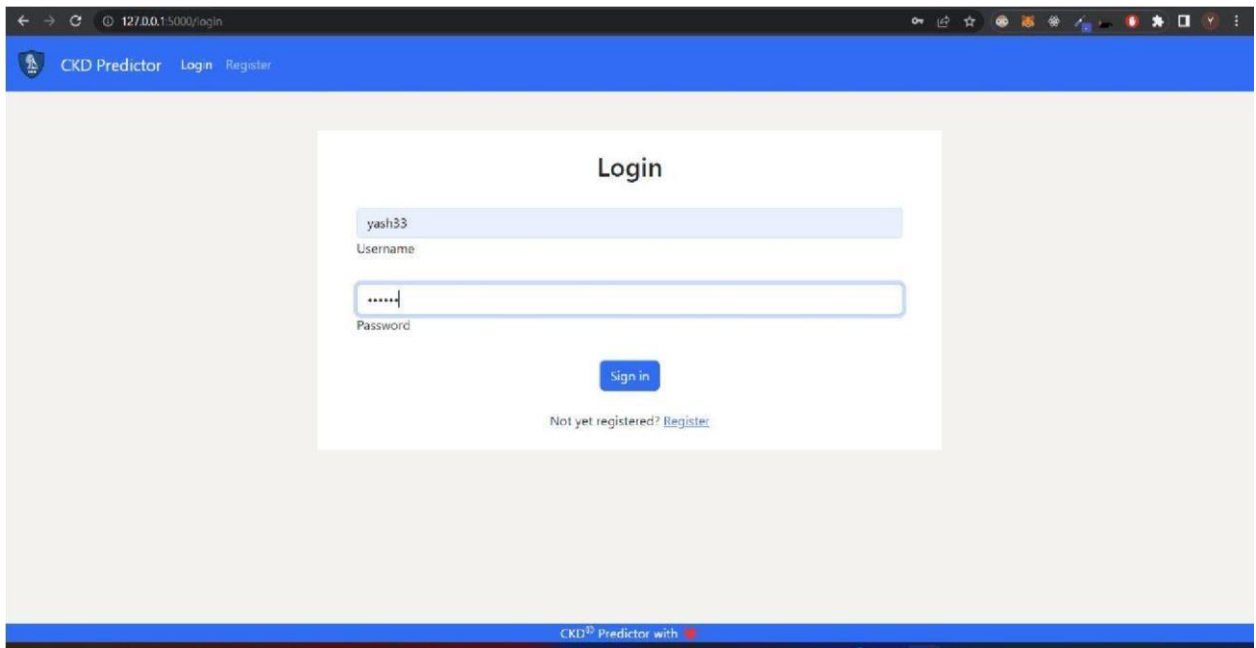
The screenshot shows a web browser at the address 127.0.0.1:5000/register. The page has a blue header with the text "CKD Predictor" and links for "Login" and "Register". The main content area is a white box titled "Register" containing four input fields: "Email address", "Username", "Create Password", and "Confirm Password". A blue "Sign Up" button is at the bottom of the form. The browser's status bar at the bottom indicates "CKD Predictor with ❤️".

Login.html



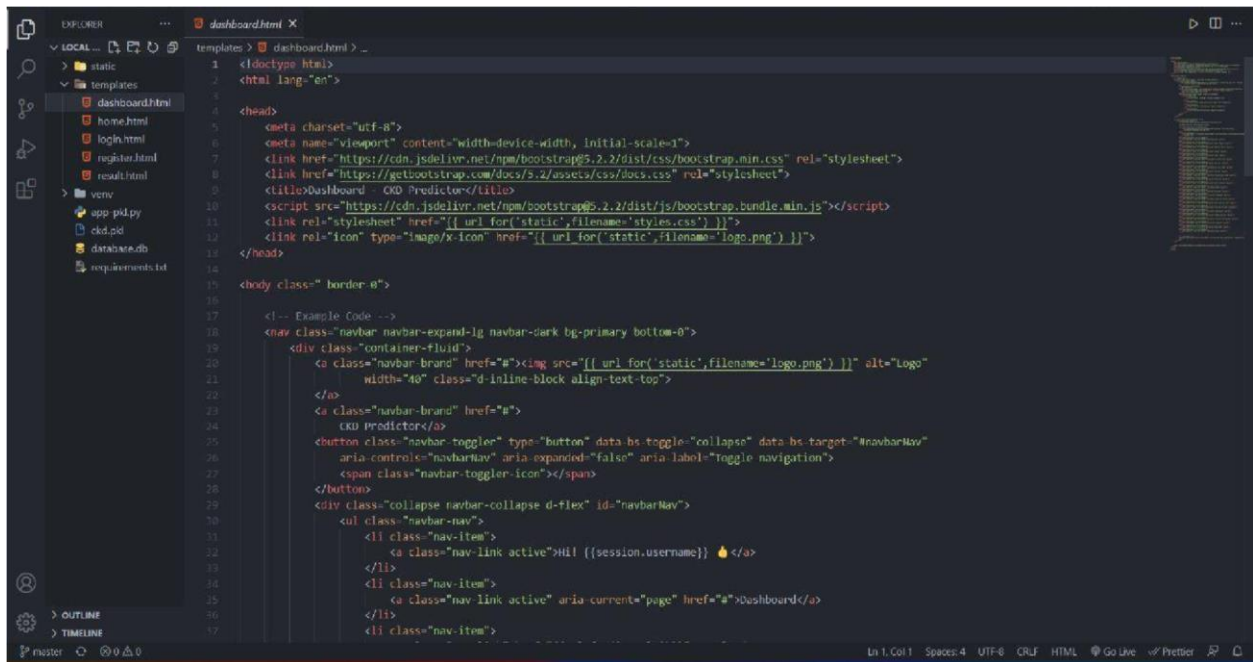
The screenshot shows a code editor with the file "login.html" open. The code is HTML and includes Bootstrap 5.2.2 CSS and JS. It features a navigation bar with a logo, a toggle button, and a list of links: "Login" and "Register". The code is as follows:

```
1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
8   <link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">
9   <title>CKD Predictor</title>
10  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>
11  <link rel="stylesheet" href="{{ url_for('static', filename='styles.css') }}">
12  <link rel="icon" type="image/x-icon" href="{{ url_for('static', filename='logo.png') }}">
13 </head>
14
15 <body class="border-0">
16
17   <!-- Example Code -->
18
19   <nav class="navbar navbar-expand-lg navbar-dark bg-primary bottom-0">
20     <div class="container-fluid">
21       <a class="navbar-brand" href="#">
23     </a>
24       <a class="navbar-brand" href="#">
25         CKD Predictor</a>
26       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
27         aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
28         <span class="navbar-toggler-icon"></span>
29     </button>
30       <div class="collapse navbar-collapse d-flex id="navbarNav">
31         <ul class="navbar-nav">
32           <li class="nav-item">
33             <a class="nav-link active" aria-current="page" href="#">Login</a>
34           </li>
35           <li class="nav-item">
36             <a class="nav-link" href="{{ url_for('register') }}">Register</a>
37           </li>
38         </ul>
39       </div>
40     </div>
41   </nav>
```

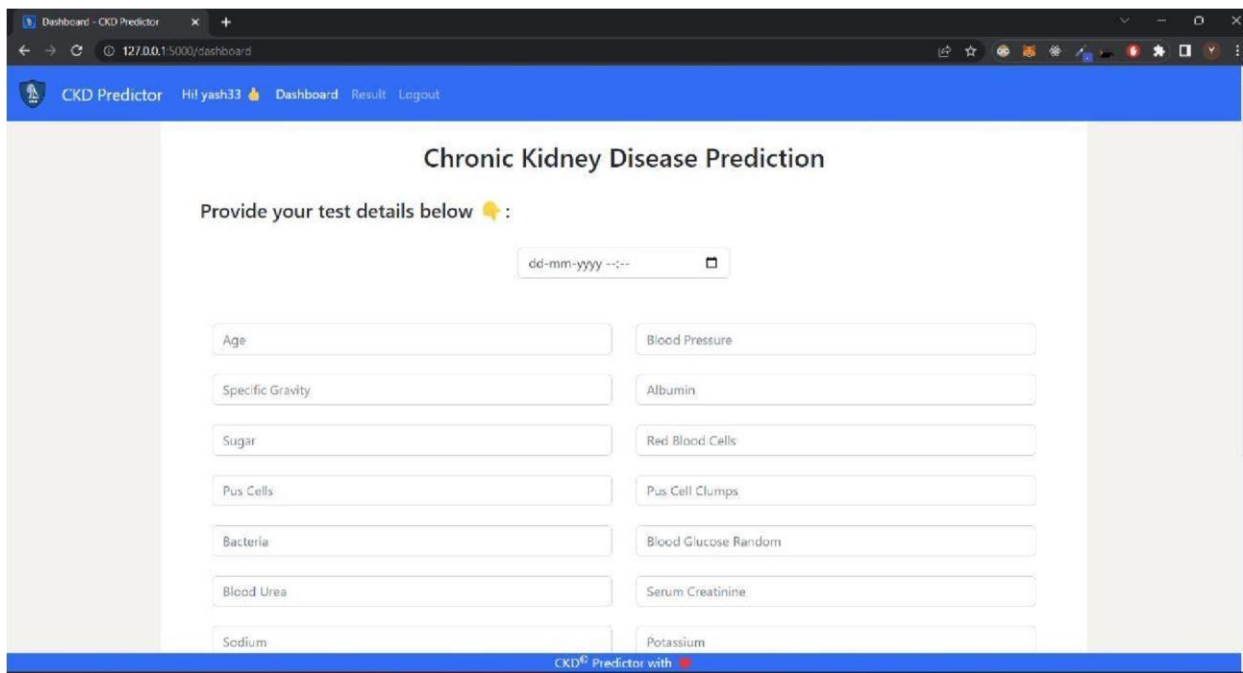



7.2 Dashboard and Result

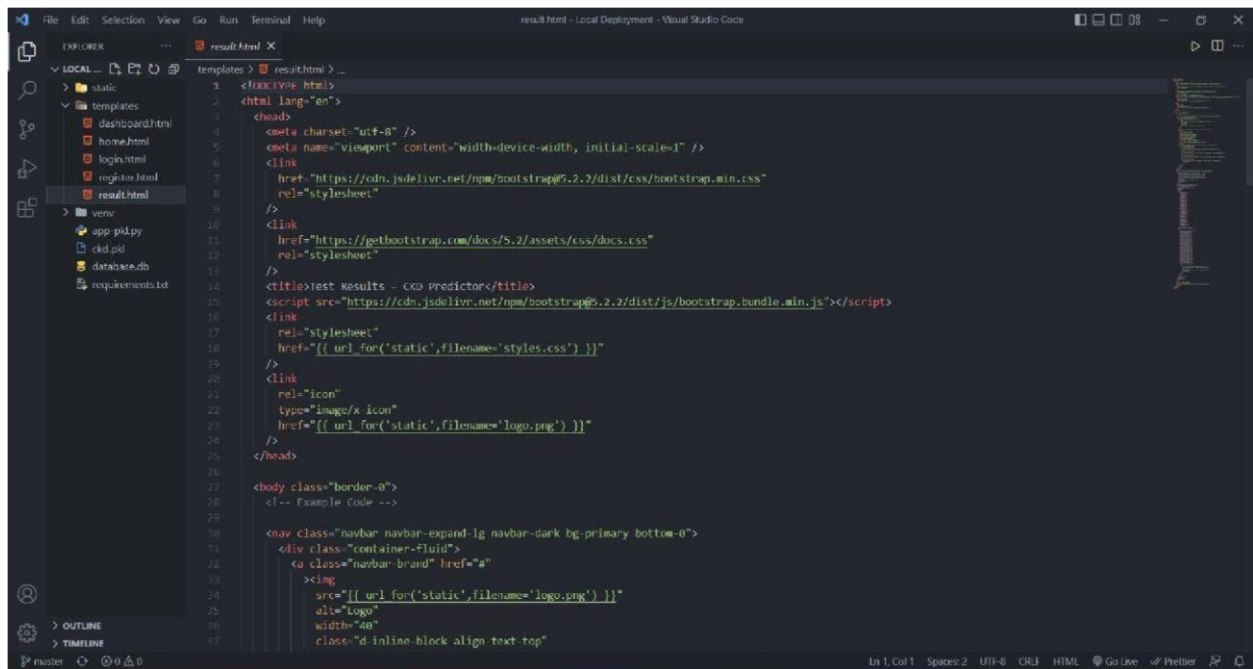
Dashboard.html

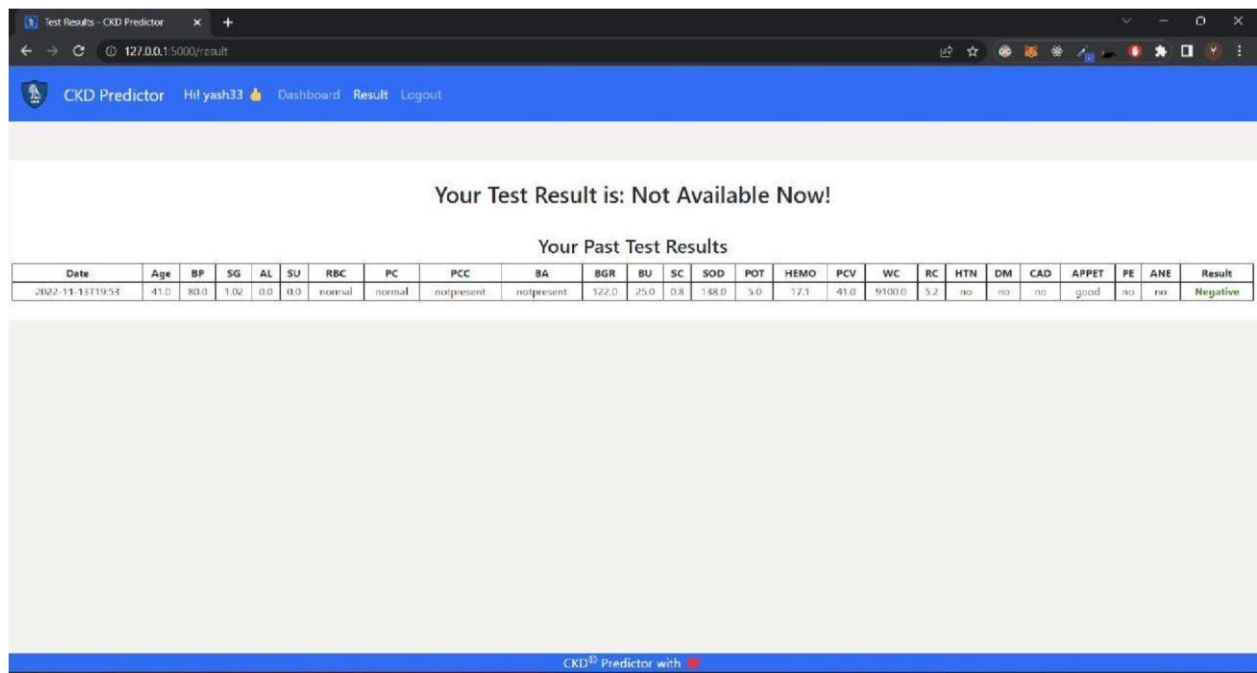



```
EXPLORER
LOCAL DEPLOYMENT
static
templates
  dashboard.html
  home.html
  login.html
  register.html
  result.html
  venw
  app-pd.py
  ckd.py
  database.db
  requirements.txt
  templates
    dashboard.html
      97
      98
      99
      100
      101
      102
      103
      104
      105
      106
      107
      108
      109
      110
      111
      112
      113
      114
      115
      116
      117
      118
      119
      120
      121
      122
      123
      124
      125
      126
      127
      128
      129
      130
      131
      132
      133
      134
      135
      136
      137
      138
      139
      140
      141
      142
      143
      144
      145
      146
      147
      148
      149
      150
      151
      152
      153
      154
      155
      156
      157
      158
      159
      160
      161
      162
      163
      164
      165
      166
      167
      168
      169
      170
      171
      172
      173
      174
      175
      176
      177
      178
      179
      180
      181
      182
      183
      184
      185
      186
      187
      188
      189
      190
      191
      192
      193
      194
      195
      196
      197
      198
      199
      200
      201
      202
      203
      204
      205
      206
      207
      208
      209
      210
      211
      212
      213
      214
      215
      216
      217
      218
      219
      220
      221
      222
      223
      224
      225
      226
      227
      228
      229
      230
      231
      232
      233
      234
      235
      236
      237
      238
      239
      240
      241
      242
      243
      244
      245
      246
      247
      248
      249
      250
      251
      252
      253
      254
      255
      256
      257
      258
      259
      260
      261
      262
      263
      264
      265
      266
      267
      268
      269
      270
      271
      272
      273
      274
      275
      276
      277
      278
      279
      280
      281
      282
      283
      284
      285
      286
      287
      288
      289
      290
      291
      292
      293
      294
      295
      296
      297
      298
      299
      300
      301
      302
      303
      304
      305
      306
      307
      308
      309
      310
      311
      312
      313
      314
      315
      316
      317
      318
      319
      320
      321
      322
      323
      324
      325
      326
      327
      328
      329
      330
      331
      332
      333
      334
      335
      336
      337
      338
      339
      340
      341
      342
      343
      344
      345
      346
      347
      348
      349
      350
      351
      352
      353
      354
      355
      356
      357
      358
      359
      360
      361
      362
      363
      364
      365
      366
      367
      368
      369
      370
      371
      372
      373
      374
      375
      376
      377
      378
      379
      380
      381
      382
      383
      384
      385
      386
      387
      388
      389
      390
      391
      392
      393
      394
      395
      396
      397
      398
      399
      400
      401
      402
      403
      404
      405
      406
      407
      408
      409
      410
      411
      412
      413
      414
      415
      416
      417
      418
      419
      420
      421
      422
      423
      424
      425
      426
      427
      428
      429
      430
      431
      432
      433
      434
      435
      436
      437
      438
      439
      440
      441
      442
      443
      444
      445
      446
      447
      448
      449
      450
      451
      452
      453
      454
      455
      456
      457
      458
      459
      460
      461
      462
      463
      464
      465
      466
      467
      468
      469
      470
      471
      472
      473
      474
      475
      476
      477
      478
      479
      480
      481
      482
      483
      484
      485
      486
      487
      488
      489
      490
      491
      492
      493
      494
      495
      496
      497
      498
      499
      500
      501
      502
      503
      504
      505
      506
      507
      508
      509
      510
      511
      512
      513
      514
      515
      516
      517
      518
      519
      520
      521
      522
      523
      524
      525
      526
      527
      528
      529
      530
      531
      532
      533
      534
      535
      536
      537
      538
      539
      540
      541
      542
      543
      544
      545
      546
      547
      548
      549
      550
      551
      552
      553
      554
      555
      556
      557
      558
      559
      560
      561
      562
      563
      564
      565
      566
      567
      568
      569
      570
      571
      572
      573
      574
      575
      576
      577
      578
      579
      580
      581
      582
      583
      584
      585
      586
      587
      588
      589
      590
      591
      592
      593
      594
      595
      596
      597
      598
      599
      600
      601
      602
      603
      604
      605
      606
      607
      608
      609
      610
      611
      612
      613
      614
      615
      616
      617
      618
      619
      620
      621
      622
      623
      624
      625
      626
      627
      628
      629
      630
      631
      632
      633
      634
      635
      636
      637
      638
      639
      640
      641
      642
      643
      644
      645
      646
      647
      648
      649
      650
      651
      652
      653
      654
      655
      656
      657
      658
      659
      660
      661
      662
      663
      664
      665
      666
      667
      668
      669
      670
      671
      672
      673
      674
      675
      676
      677
      678
      679
      680
      681
      682
      683
      684
      685
      686
      687
      688
      689
      690
      691
      692
      693
      694
      695
      696
      697
      698
      699
      700
      701
      702
      703
      704
      705
      706
      707
      708
      709
      710
      711
      712
      713
      714
      715
      716
      717
      718
      719
      720
      721
      722
      723
      724
      725
      726
      727
      728
      729
      730
      731
      732
      733
      734
      735
      736
      737
      738
      739
      740
      741
      742
      743
      744
      745
      746
      747
      748
      749
      750
      751
      752
      753
      754
      755
      756
      757
      758
      759
      760
      761
      762
      763
      764
      765
      766
      767
      768
      769
      770
      771
      772
      773
      774
      775
      776
      777
      778
      779
      780
      781
      782
      783
      784
      785
      786
      787
      788
      789
      790
      791
      792
      793
      794
      795
      796
      797
      798
      799
      800
      801
      802
      803
      804
      805
      806
      807
      808
      809
      810
      811
      812
      813
      814
      815
      816
      817
      818
      819
      820
      821
      822
      823
      824
      825
      826
      827
      828
      829
      830
      831
      832
      833
      834
      835
      836
      837
      838
      839
      840
      841
      842
      843
      844
      845
      846
      847
      848
      849
      850
      851
      852
      853
      854
      855
      856
      857
      858
      859
      860
      861
      862
      863
      864
      865
      866
      867
      868
      869
      870
      871
      872
      873
      874
      875
      876
      877
      878
      879
      880
      881
      882
      883
      884
      885
      886
      887
      888
      889
      890
      891
      892
      893
      894
      895
      896
      897
      898
      899
      900
      901
      902
      903
      904
      905
      906
      907
      908
      909
      910
      911
      912
      913
      914
      915
      916
      917
      918
      919
      920
      921
      922
      923
      924
      925
      926
      927
      928
      929
      930
      931
      932
      933
      934
      935
      936
      937
      938
      939
      940
      941
      942
      943
      944
      945
      946
      947
      948
      949
      950
      951
      952
      953
      954
      955
      956
      957
      958
      959
      960
      961
      962
      963
      964
      965
      966
      967
      968
      969
      970
      971
      972
      973
      974
      975
      976
      977
      978
      979
      980
      981
      982
      983
      984
      985
      986
      987
      988
      989
      990
      991
      992
      993
      994
      995
      996
      997
      998
      999
      1000
      1001
      1002
      1003
      1004
      1005
      1006
      1007
      1008
      1009
      1010
      1011
      1012
      1013
      1014
      1015
      1016
      1017
      1018
      1019
      1020
      1021
      1022
      1023
      1024
      1025
      1026
      1027
      1028
      1029
      1030
      1031
      1032
      1033
      1034
      1035
      1036
      1037
      1038
      1039
      1040
      1041
      1042
      1043
      1044
      1045
      1046
      1047
      1048
      1049
      1050
      1051
      1052
      1053
      1054
      1055
      1056
      1057
      1058
      1059
      1060
      1061
      1062
      1063
      1064
      1065
      1066
      1067
      1068
      1069
      1070
      1071
      1072
      1073
      1074
      1075
      1076
      1077
      1078
      1079
      1080
      1081
      1082
      1083
      1084
      1085
      1086
      1087
      1088
      1089
      1090
      1091
      1092
      1093
      1094
      1095
      1096
      1097
      1098
      1099
      1100
      1101
      1102
      1103
      1104
      1105
      1106
      1107
      1108
      1109
      1110
      1111
      1112
      1113
      1114
      1115
      1116
      1117
      1118
      1119
      1120
      1121
      1122
      1123
      1124
      1125
      1126
      1127
      1128
      1129
      1130
      1131
      1132
      1133
      1134
      1135
      1136
      1137
      1138
      1139
      1140
      1141
      1142
      1143
      1144
      1145
      1146
      1147
      1148
      1149
      1150
      1151
      1152
      1153
      1154
      1155
      1156
      1157
      1158
      1159
      1160
      1161
      1162
      1163
      1164
      1165
      1166
      1167
      1168
      1169
      1170
      1171
      1172
      1173
      1174
      1175
      1176
      1177
      1178
      1179
      1180
      1181
      1182
      1183
      1184
      1185
      1186
      1187
      1188
      1189
      1190
      1191
      1192
      1193
      1194
      1195
      1196
      1197
      1198
      1199
      1200
      1201
      1202
      1203
      1204
      1205
      1206
      1207
      1208
      1209
      1210
      1211
      1212
      1213
      1214
      1215
      1216
      1217
      1218
      1219
      1220
      1221
      1222
      1223
      1224
      1225
      1226
      1227
      1228
      1229
      1230
      1231
      1232
      1233
      1234
      1235
      1236
      1237
      1238
      1239
      1240
      1241
      1242
      1243
      1244
      1245
      1246
      1247
      1248
      1249
      1250
      1251
      1252
      1253
      1254
      1255
      1256
      1257
      1258
      1259
      1260
      1261
      1262
      1263
      1264
      1265
      1266
      1267
      1268
      1269
      1270
      1271
      1272
      1273
      1274
      1275
      1276
      1277
      1278
      1279
      1280
      1281
      1282
      1283
      1284
      1285
      1286
      1287
      1288
      1289
      1290
      1291
      1292
      1293
      1294
      1295
      1296
      1297
      1298
      1299
      1300
      1301
      1302
      1303
      1304
      1305
      1306
      1307
      1308
      1309
      1310
      1311
      1312
      1313
      1314
      1315
      1316
      1317
      1318
      1319
      1320
      1321
      1322
      1323
      1324
      1325
      1326
      1327
      1328
      1329
      1330
      1331
      1332
      1333
      1334
      1335
      1336
      1337
      1338
      1339
      1340
      1341
      1342
      1343
      1344
      1345
      1346
      1347
      1348
      1349
      1350
      1351
      1352
      1353
      1354
      1355
      1356
      1357
      1358
      1359
      1360
      1361
      1362
      1363
      1364
      1365
      1366
      1367
      1368
      1369
      1370
      1371
      1372
      1373
      1374
      1375
      1376
      1377
      1378
      1379
      1380
      1381
      1382
      1383
      1384
      1385
      1386
      1387
      1388
      1389
      1390
      1391
      1392
      1393
      1394
      1395
      1396
      1397
      1398
      1399
      1400
      1401
      1402
      1403
      1404
      1405
      1406
      1407
      1408
      1409
      1410
      1411
      1412
      1413
      1414
      1415
      1416
      1417
      1418
      1419
      1420
      1421
      1422
      1423
      1424
      1425
      1426
      1427
      1428
      1429
      1430
      1431
      1432
      1433
      1434
      1435
      1436
      1437
      1438
      1439
      1440
      1441
      1442
      1443
      1444
      1445
      1446
      1447
      1448
      1449
      1450
      1451
      1452
      1453
      1454
      1455
      1456
      1457
      1458
      1459
      1460
      1461
      1462
      1463
      1464
      1465
      1466
      1467
      1468
      1469
      1470
      1471
      1472
      1473
      1474
      1475
      1476
      1477
      1478
      1479
      1480
      1481
      1482
      1483
      1484
      1485
      1486
      1487
      1488
      1489
      1490
      1491
      1492
      1493
      1494
      1495
      1496
      1497
      1498
      1499
      1500
      1501
      1502
      1503
      1504
      1505
      1506
      1507
      1508
      1509
      1510
      1511
      1512
      1513
      1514
      1515
      1516
      1517
      1518
      1519
      1520
      1521
      1522
      1523
      1524
      1525
      1526
      1527
      1528
      1529
      1530
      1531
      1532
      1533
      1534
      1535
      1536
      1537
      1538
      1539
      1540
      1541
      1542
      1543
      1544
      1545
      1546
      1547
      1548
      1549
      1550
      1551
      1552
      1553
      1554
      1555
      1556
      1557
      1558
      1559
      1560
      1561
      1562
      1563
      1564
      1565
      1566
      1567
      1568
      1569
      1570
      1571
      1572
      1573
      1574
      1575
      1576
      1577
      1578
      1579
      1580
      1581
      1582
      1583
      1584
      1585
      1586
      1587
      1588
      1589
      1590
      1591
      1592
      1593
      1594
      1595
      1596
      1597
      1598
      1599
      1600
      1601
      1602
      1603
      1604
      1605
      1606
      1607
      1608
      1609
      1610
      1611
      1612
      1613
      1614
      1615
      1616
      1617
      1618
      1619
      1620
      1621
      1622
      1623
      1624
      1625
      1626
      1627
      1628
      1629
      1630
      1631
      1632
      1633
      1634
      1635
      1636
      1637
      1638
      1639
      1640
      1641
      1642
      1643
      1644
      1645
      1646
      1647
      1648
      1649
      1650
      1651
      1652
      1653
      1654
      1655
      1656
      1657
      1658
      1659
      1660
      1661
      1662
      1663
      1664
      1665
      1666
      1667
      1668
      1669
      1670
      1671
      1672
      1673
      1674
      1675
      1676
      1677
      1678
      1679
      1680
      1681
      1682
      1683
      1684
      1685
      1686
      1687
      1688
      1689
      1690
      1691
      1692
      1693
      1694
      1695
      1696
      1697
      1698
      1699
      1700
      1701
      1702
      1703
      1704
      1705
      1706
      1707
      1708
      1709
      1710
      1711
      1712
      1713
      1714
      1715
      1716
      1717
      1718
      1719
      1720
      1721
      1722
      1723
      1724
      1725
      1726
      1727
      1728
      1729
      1730
      1731
      1732
      1733
      1734
      1735
      1736
      1737
      1738
      1739
      1740
      1741
      1742
      1743
      1744
      1745
      1746
      1747
      1748
      1749
      1750
      1751
      1752
      1753
      1754
      1755
      1756
      1757
      1758
      1759
      1760
      1761
      1762
      1763
      1764
      1765
      1766
      1767
      1768
      1769
      1770
      1771
      1772
      1773
      1774
      1775
      1776
      1777
      1778
      1779
      1780
      1781
      1782
      1783
      1784
      1785
      1786
      1787
      1788
      1789
      1790
      1791
      1792
      1793
      1794
      1795
      1796
      1797
      1798
      1799
      1800
      1801
      1802
      1803
      1804
      1805
      1806
      1807
      1808
      1809
      1810
      1811
      1812
      1813
      1814
      1815
      1816
      1817
      1818
      1819
      1820
      1821
      1822
      1823
      1824
      1825
      1826
      1827
      1828
      1829
      1830
      1831
      1832
      1833
      1834
      1835
      1836
      1837
      1838
      1839
      1840
      1841
      1842
      1843
      1844
      1845
      1846
      1847
      1848
      1849
      1850
      1851
      1852
      1853
      1854
      1855
      1856
      1857
      1858
      1859
      1860
      1861
      1862
      1863
      1864
      1865
      1866
      1867
      1868
      1869
      1870
      1871
      1872
      1873
      1874
      1875
      1876
      1877
      1878
      1879
      1880
      1881
      1882
      1883
      1884
      1885
      1886
      1887
      1888
      1889
      1890
      1891
      1892
      1893
      1894
      1895
      1896
      1897
      1898
      1899
      1900
      1901
      1902
      1903
      1904
      1905
      1906
      1907
      1908
      1909
      1910
      1911
      1912
      1913
      1914
      1915
      1916
      1917
      1918
      1919
      1920
      1921
      1922
      1923
      1924
      1925
      1926
      1927
      1928
      1929
      1930
      1931
      1932
      1933
      1934
      1935
      1936
      1937
      1938
      1939
      1940
      1941
      1942
      1943
      1944
      1945
      1946
      1947
      1948
      1949
      1950
      1951
      1952
      1953
      1954
      1955
      1956
      1957
      1958
      1959
      1960
      1961
      1962
      1963
      1964
      1965
      1966
      1967
      1968
      1969
      1970
      1971
      1972
      1973
      1974
      1975
      1976
      1977
      1978
      1979
      1980
      1981
      
```

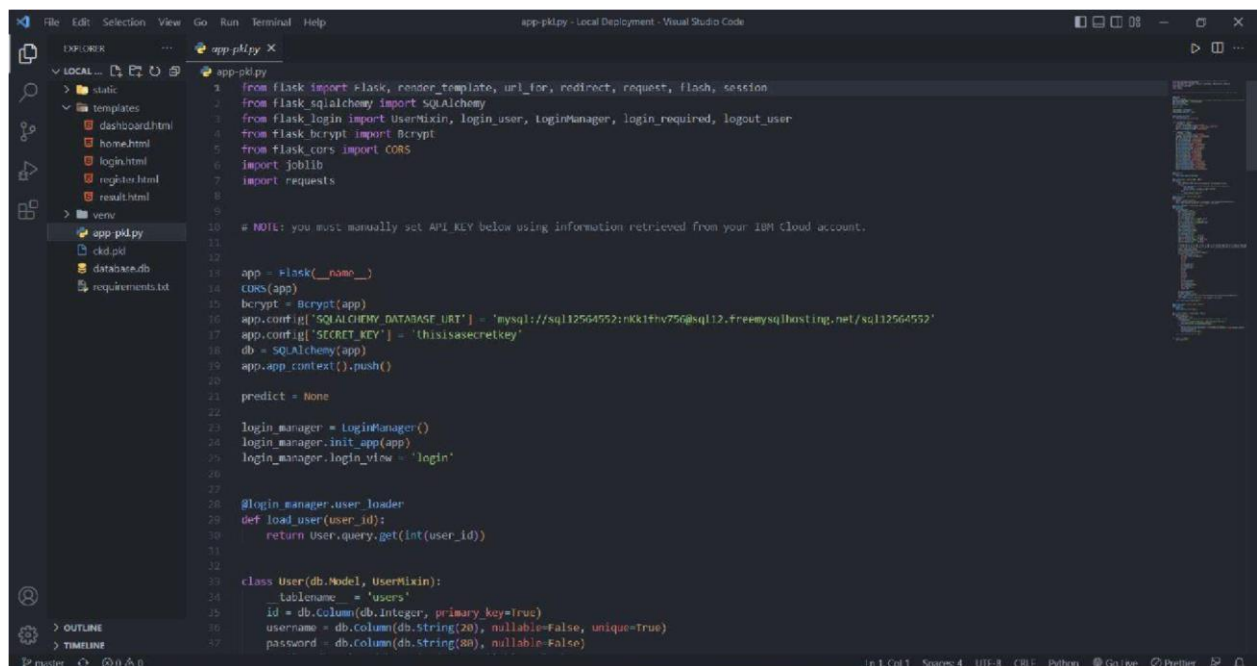


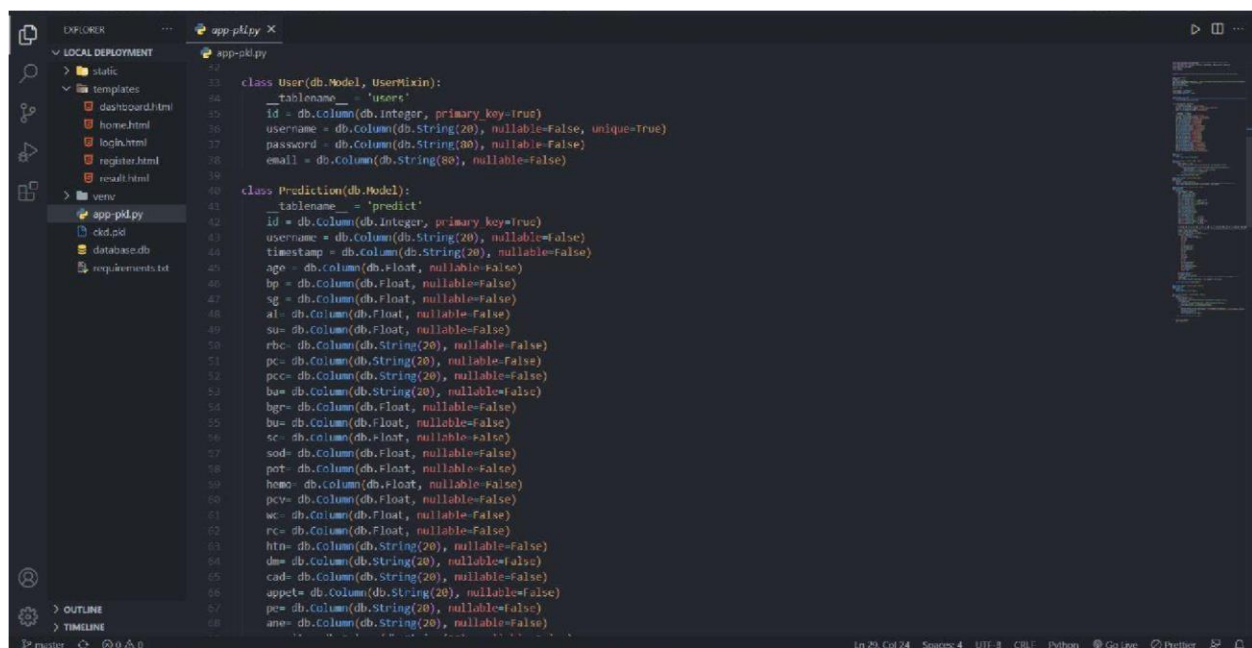
Result.html





Flask Integration and Deployment





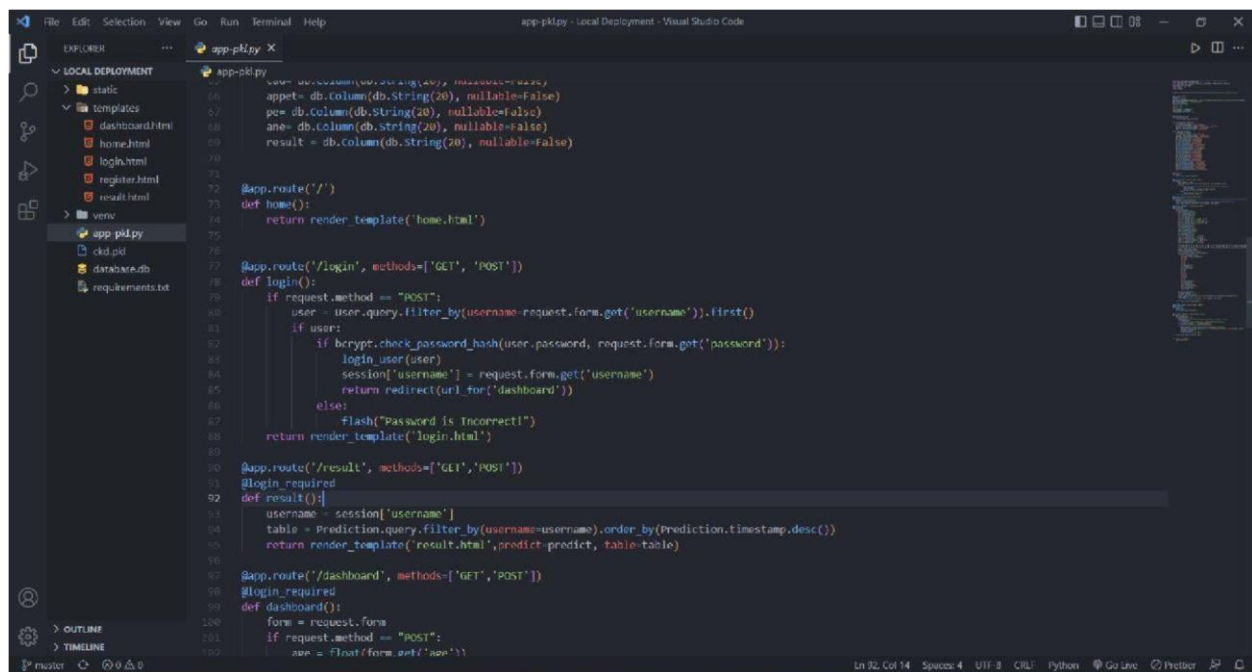
EXPLORER

LOCAL DEPLOYMENT

- static
- templates
 - dashboard.html
 - home.html
 - login.html
 - register.html
 - result.html
- venv
 - app-pk1.py
 - cdk.pkl
 - database.db
 - requirements.txt

```
13 class User(db.Model, UserMixin):
14     __tablename__ = 'users'
15     id = db.Column(db.Integer, primary_key=True)
16     username = db.Column(db.String(20), nullable=False, unique=True)
17     password = db.Column(db.String(80), nullable=False)
18     email = db.Column(db.String(80), nullable=False)
19
20 class Prediction(db.Model):
21     __tablename__ = 'predict'
22     id = db.Column(db.Integer, primary_key=True)
23     username = db.Column(db.String(20), nullable=False)
24     timestamp = db.Column(db.String(20), nullable=False)
25     age = db.Column(db.Float, nullable=False)
26     bp = db.Column(db.Float, nullable=False)
27     sg = db.Column(db.Float, nullable=False)
28     al = db.Column(db.Float, nullable=False)
29     su = db.Column(db.Float, nullable=False)
30     rhc = db.Column(db.String(20), nullable=False)
31     pc = db.Column(db.String(20), nullable=False)
32     pcc = db.Column(db.String(20), nullable=False)
33     bu = db.Column(db.String(20), nullable=False)
34     hgr = db.Column(db.Float, nullable=False)
35     bu = db.Column(db.Float, nullable=False)
36     sc = db.Column(db.Float, nullable=False)
37     sod = db.Column(db.Float, nullable=False)
38     pot = db.Column(db.Float, nullable=False)
39     homo = db.Column(db.Float, nullable=False)
40     pcv = db.Column(db.Float, nullable=False)
41     wc = db.Column(db.Float, nullable=False)
42     rc = db.Column(db.Float, nullable=False)
43     ht = db.Column(db.String(20), nullable=False)
44     dm = db.Column(db.String(20), nullable=False)
45     cad = db.Column(db.String(20), nullable=False)
46     appet = db.Column(db.String(20), nullable=False)
47     pc = db.Column(db.String(20), nullable=False)
48     ane = db.Column(db.String(20), nullable=False)
```

Ln 20, Col 24 | Scope: 4 | UTF-8 | CRLF | Python | Go Live | Prettier



EXPLORER

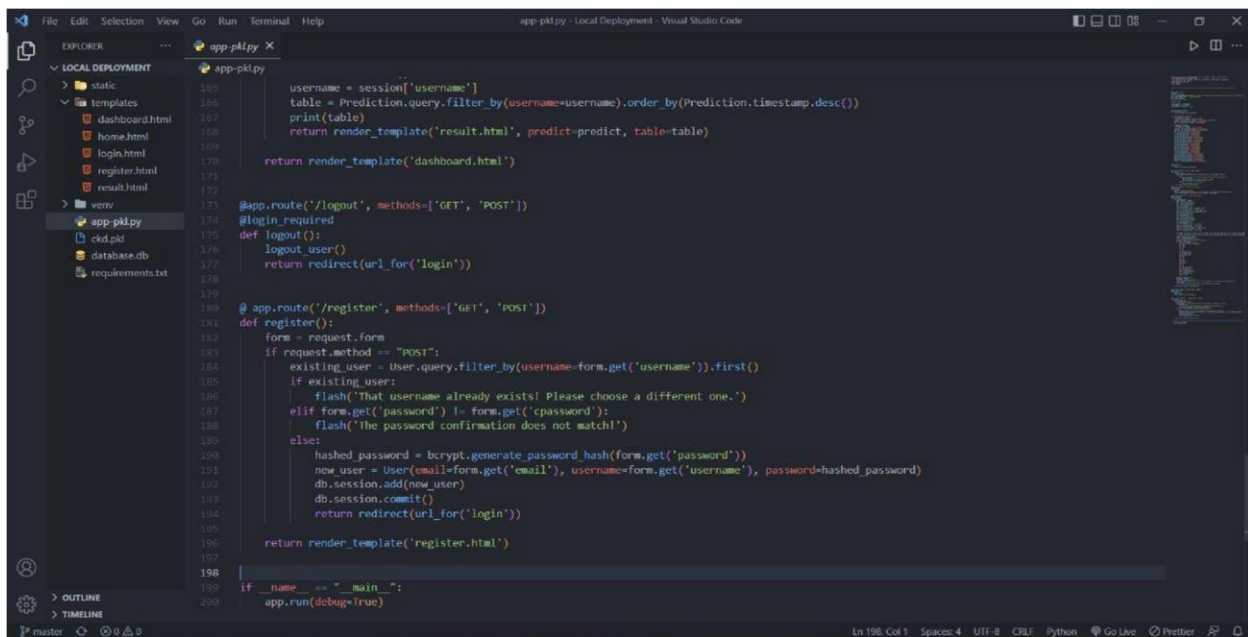
LOCAL DEPLOYMENT

- static
- templates
 - dashboard.html
 - home.html
 - login.html
 - register.html
 - result.html
- venv
 - app-pk1.py
 - cdk.pkl
 - database.db
 - requirements.txt

```
56 user = db.Column(db.String(80), nullable=False)
57 appet = db.Column(db.String(20), nullable=False)
58 pc = db.Column(db.String(20), nullable=False)
59 ane = db.Column(db.String(20), nullable=False)
60 result = db.Column(db.String(20), nullable=False)
61
62 @app.route('/')
63 def home():
64     return render_template('home.html')
65
66 @app.route('/login', methods=['GET', 'POST'])
67 def login():
68     if request.method == "POST":
69         user = user.query.filter_by(username=request.form.get('username')).first()
70         if user:
71             if bcrypt.check_password_hash(user.password, request.form.get('password')):
72                 login_user(user)
73                 session['username'] = request.form.get('username')
74                 return redirect(url_for('dashboard'))
75             else:
76                 flash("Password is Incorrect!")
77         return render_template('login.html')
78
79 @app.route('/result', methods=['GET', 'POST'])
80 @login_required
81 def result():
82     username = session['username']
83     table = Prediction.query.filter_by(username=username).order_by(Prediction.timestamp.desc())
84     return render_template('result.html', predict=predict, table=table)
85
86 @app.route('/dashboard', methods=['GET', 'POST'])
87 @login_required
88 def dashboard():
89     form = request.form
90     if request.method == "POST":
91         age = float(form.get('age'))
```

Ln 92, Col 14 | Scope: 4 | UTF-8 | CRLF | Python | Go Live | Prettier

The status bar at the bottom indicates 'Ln 94, Col 30', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', 'Go Live', 'Prettier', and 'G'.
The status bar at the bottom indicates 'Ln 126, Col 28', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', 'Go Live', 'Prettier', and 'G'.



Using pickle to integrate with flask

```
In [29]: import joblib #created a pickle file using joblib to export the model for frontend usage
         joblib.dump(model,'rfc.pkl') # model - Random Forest Classifier

Out[29]: ['rfc.pkl']

In [ ]:
```

Flask changes for ibm deployment

```
File Edit Selection View Go Run Terminal Help
app-ibm.py - IBM Deployment - Visual Studio Code

EXPLORER
  IBM DEPLOYMENT
    static
    templates
    venv
    app-ibm.py
    ckd.pid
    database.db
    requirements.txt

app-ibm.py
1 from flask import Flask, render_template, url_for, redirect, request, flash, session
2 from flask_sqlalchemy import SQLAlchemy
3 from flask_login import UserMixin, login_user, login_manager, login_required, logout_user
4 from flask_bcrypt import Bcrypt
5 from flask_cors import CORS
6 import requests
7
8
9 # NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
10 API_KEY = "M01pdcKwE1QfugulqjnlZocw0d7Rkd1i8Wk0zR0"
11 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY,
12 "grant_type": "urn:ibm:params:oauth:grant-type:apikey"})
13 mltoken = token_response.json()["access_token"]
14
15 header = {'content-type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
16
17
18 app = Flask(__name__)
19 CORS(app)
20 bcrypt = Bcrypt(app)
21 app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://sql12564552:nKk1fhv756@sql12.freemysqhosting.net/sql12564552'
22 app.config['SECRET_KEY'] = 'thisisasecretkey'
23 db = SQLAlchemy(app)
24 app.app_context().push()
25
26 predict = None
27
28 login_manager = login_manager()
29 login_manager.init_app(app)
30 login_manager.login_view = 'login'
31
32
33 @login_manager.user_loader
34 def load_user(user_id):
35     return User.query.get(int(user_id))
36
37
```

```
File Edit Selection View Go Run Terminal Help
app-ibm.py - IBM Deployment - Visual Studio Code

EXPLORER
  IBM DEPLOYMENT
    static
    templates
    venv
    app-ibm.py
    ckd.pid
    database.db
    requirements.txt

app-ibm.py
100 pot= float(form.get('pot'))
101 hemo= float(form.get('hemo'))
102 pcv= float(form.get('pcv'))
103 wc= float(form.get('wc'))
104 rc= float(form.get('rc'))
105 htn= 0 if form.get('htn')== 'no' else 1
106 dm= 0 if form.get('dm')== 'no' else 1
107 cad= 0 if form.get('cad')== 'no' else 1
108 appet= 0 if form.get('appet')== 'good' else 1
109 pe= 0 if form.get('pe')== 'no' else 1
110 ane= 0 if form.get('ane')== 'no' else 1
111
112 print(age, bp, sg, al, su, rbc, pc, pcc, ha, hgr, bu, sc, sod, pot, hemo, pcv, wc, rc, htn, dm, cad, appet, pe, ane)
113 x = [[age, bp, sg, al, su, rbc, pc, pcc, ha, hgr, bu, sc, sod, pot, hemo, pcv, wc, rc, htn, dm, cad, appet, pe, ane]]
114 # NOTE: manually define and pass the array(s) of values to be scored in the next line
115 payload_scoring = {"input_data": [{"field": [{"age", 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ha', 'hgr', 'bu', 'sc', 'sod', 'pot', 'hemo',
116 'pcv', 'wc', 'rc', 'htn', 'dm', 'cad', 'appet', 'pe', 'ane'}]}]}
117
118 response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/4b20f2e-e060-412e-875e-a336e199f1aa/predictions?vers=1',
119 json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
120 print(response_scoring)
121 predictions = response_scoring.json()
122 predict = predictions["predictions"][0]['values'][0][0]
123 print("Final prediction is:", predict)
124 res = 'Positive' if predict==1 else 'Negative'
125 new_user = Prediction(
126     username=form.get('username'),
127     timestamp = form.get('timestamp'),
128     age = age,
129     bp = bp,
130     sg = sg,
131     al= al,
132     su= su,
133     rbc= form.get('rbc'),
134     pc= form.get('pc'),
135     pcc= form.get('pcc'),
136     ha= form.get('ha'),
137     hgr= hgr,
138
```

```
146 timestamp = form.get('timestamp'),
147 age = age,
148 bp = bp,
149 sg = sg,
150 al = al,
151 su = su,
152 rbc = form.get('rbc'),
153 pc = form.get('pc'),
154 pcc = form.get('pcc'),
155 ba = form.get('ba'),
156 bgr = bgr,
157 hu = hu,
158 sc = sc,
159 sod = sod,
160 pot = pot,
161 hemo = hemo,
162 pcv = pcv,
163 wc = wc,
164 rc = rc,
165 htn = form.get('htn'),
166 dm = form.get('dm'),
167 cad = form.get('cad'),
168 appet = form.get('appet'),
169 pe = form.get('pe'),
170 ane = form.get('ane'),
171 result = res
172
173 db.session.add(new_user)
174 db.session.commit()
175 username = session['username']
176 table = Prediction.query.filter_by(username=username).order_by(Prediction.timestamp.desc())
177 print(table)
178 return render_template('result.html', predict=predict, table=table)
179
180 return render_template('dashboard.html')
```

8. TESTING

Introduction:

After finishing the development of any computer based system the next complicated time consuming process is system testing. During the time of testing only the development company can know that, how far the user requirements have been met out, and so on.

Software testing is an important element of the software quality assurance and represents the ultimate review of specification, design and coding. The increasing feasibility of software as a system and the cost associated with the software failures are motivated forces for well planned through testing.

Testing Objectives

These are several rules that can save as testing objectives they are:

- Testing is a process of executing program with the intent of finding an error.
- A good test case is one that has a high probability of finding an undiscovered error.

Following are the some of the testing methods applied to this effective project:

8.1 SOURCE CODE TESTING:

This examines the logic of the system. If we are getting the output that is required by the user, then we can say that the logic is perfect.

8.1.2 SPECIFICATION TESTING:

We can set with, what program should do and how it should perform under various condition. This testing is a comparative study of evolution of system performance and system requirements.

8.1.3 MODULE LEVEL TESTING:

In this the error will be found at each individual module, it encourages the programmer to find and rectify the errors without affecting the other modules.

8.1.4 UNIT TESTING:

Unit testing focuses on verifying the effort on the smallest unit of software-module. The local data structure is examined to ensure that the data stored temporarily maintains its integrity during all steps in the algorithm's execution. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing.

8.1.5 INTEGRATION TESTING:

Data can be tested across an interface. One module can have an inadvertent, adverse effect on the other. **Integration testing** is a systematic technique for constructing a program structure while conducting tests to uncover errors associated with interring.

8.1.6 VALIDATION TESTING:

It begins after the integration testing is successfully assembled. Validation succeeds when the software functions in a manner that can be reasonably accepted by the client. In this the majority of the validation is done during the data entry operation where there is a maximum possibility of entering wrong data. Other validation will be performed in all process where correct details and data should be entered to get the required results.

8.1.7 RECOVERY TESTING:

Recovery Testing is a system that forces the software to fail in variety of ways and verifies that the recovery is properly performed. If recovery is automatic, re-initialization, and data recovery are each evaluated for correctness

8.1.8 SECURITY TESTING:

Security testing attempts to verify that protection mechanism built into system will in fact protect it from improper penetration. The tester may attempt to acquire password through external clerical means, may attack the system with custom software design to break down any defenses to others, and may purposely cause errors.

8.1.9 PERFORMANCE TESTING:

Performance Testing is used to test runtime performance of software within the context of an integrated system. Performance test are often coupled with stress testing and require both software instrumentation.

8.1.10 BLACKBOX TESTING:

Black- box testing focuses on functional requirement of software. It enables to derive sets of input conditions that will fully exercise all functional requirements for a program. Black box testing attempts to find error in the following category:

- ❖ Incorrect or missing function
- ❖ Interface errors
- ❖ Errors in data structures or external database access and performance errors.

8.1.11 OUTPUT TESTING:

After performing the validation testing, the next step is output testing of the proposed system since no system would be termed as useful until it does produce the required output in the specified format. **Output format** is considered in two ways, the **screen format** and the **printer format**.

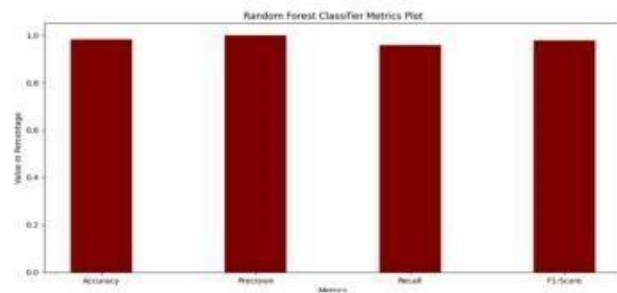
8.2 USER ACCEPTANCE TESTING:

User Acceptance Testing is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system users at the time of developing and making changes whenever required.

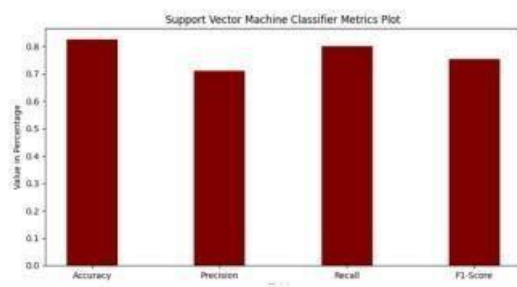
TEST CASES

S. No	Test Case Name	Test Procedure	Pre-Condition	Expected Result	Passed/ failed
1	Data Input	Enter no details and click submit button	Enter no details input	Alert “Select Dataset, Enter Latitude, Longitude”	Passed
2	Data Input	Select dataset and click submit button	Select dataset and click submit button	Alert “Select Dataset, Enter Latitude, Longitude”	Passed
3	Data Input	Select dataset, enter latitude and click submit button	Select dataset, enter latitude and click submit button	Alert “Select Dataset, Enter Latitude, Longitude”	Passed

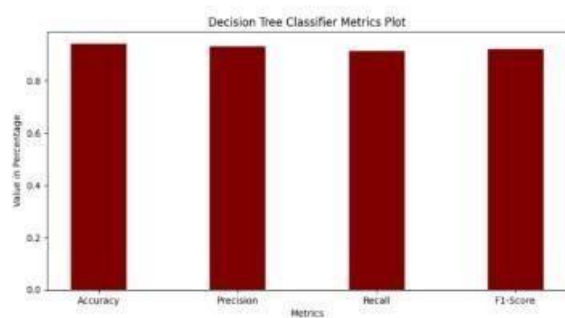
9.1 RESULTS



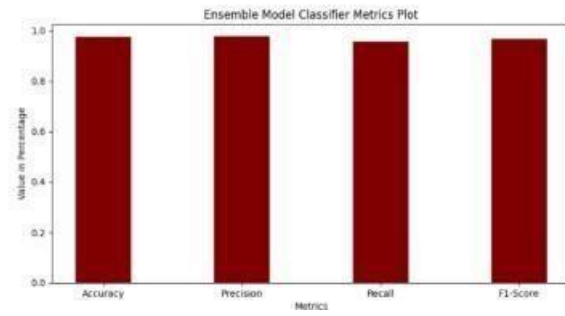
Graph- 1: Plotting of Random Forest Classifier Metrics



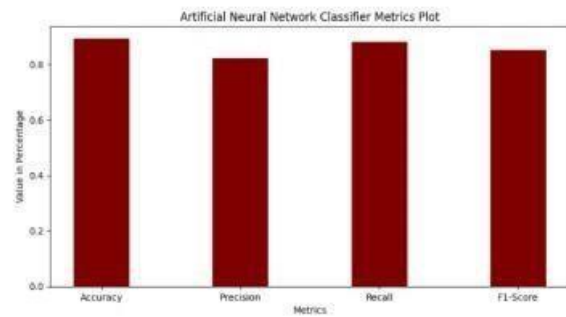
Graph- 2: Plotting of Support Vector Machine Classifier Metrics



Graph- 3: Plotting of Decision Tree Classifier Metrics



Graph- 4: Plotting of Ensemble Model Classifier Metrics



Graph- 5: Plotting of Artificial Neural Network Classifier Metrics

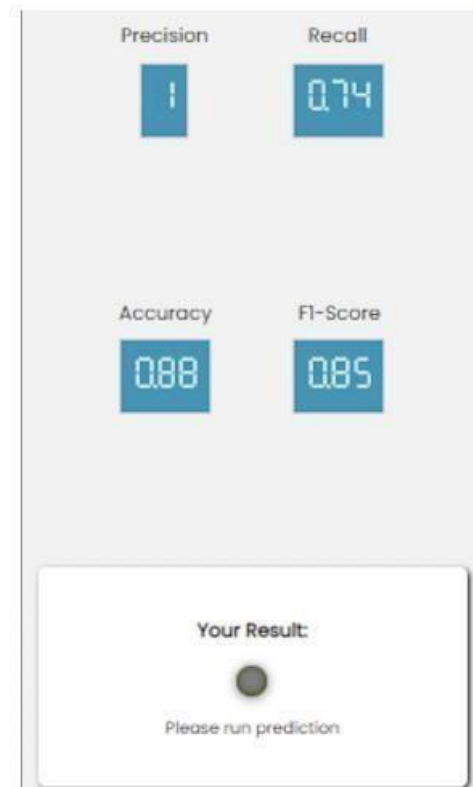
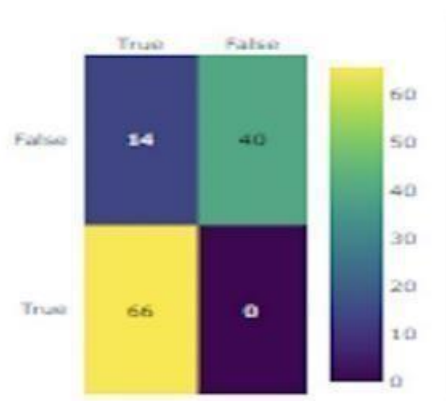


Fig - 5 : Representation of Precision & Accuracy

10. ADVANTAGE & DISADVANTAGE

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 its a web application it requires scaling of web application to handle concurrent requests after certain threshold

11CONCLUSION

This study developed an algorithm for predicting CKD at an early stage. The dataset contains input parameters obtained from CKD patients, and the models are trained and validated using the valid parameters. To diagnose CKD, decision tree, random forest, and support vector machine learning models are built. The accuracy of prediction is used to assess the performance of the models. The study's findings revealed that the Random Forest Classifier model outperforms Decision Trees and Support Vector Machines in predicting CKD. As an extension of this research, the comparison may also be done depending on the duration of execution and feature set selection.

12.FUTURE SCOPE

This work will be considered as basement for the healthcare system for CKD patients. Also extension to this work is that implementation of deep learning since deep learning provides high-quality performance than machine learning algorithm.

13. Appendix:

<https://ieeexplore.ieee.org/abstract/document/8029917>

<https://iopscience.iop.org/article/10.1088/1742-6596/1255/1/012024/meta>

<https://start.atlassian.com/> <https://ieeexplore.ieee.org/abstract/document/9333572>

GITHUB LINK :

<https://github.com/IBM-EPBL/IBM-Project-49613-1660829809>

