PROJECT REPORT DOCUMENTATION

EARLY DETECTION OF CHRONIC KIDNEY DISEASE
TEAM ID - PNT2022TMID21262

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 using API keys and scoring endpoints.

2. LITERATURE SURVEY

2.1 Existing Problem:-

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 References :-

SNO	LITERATURE PAPER	AUTHOR	PROPOSED METHOD	ACCURACY	YEAR
1	Computer-Aided Diagnosis of Chronic Kidney Disease in Developing Countries: A Comparative Analysis of Machine Learning Techniques	Andressa C. M. Da S. Queiroz, Alvaro Sobrinho, Leandro Dias Da Silva, Evandro De Barros Costa, Maria Eliete Pinheiro, Angelo Perkusich	J48 decision tree is a suitable machine learning technique for such screening in developing countries, due to the easy interpretation of its classification results	95.00%	2020

2	Chronic Kidney Disease Prediction using Machine Learning Models	S.Revathy, B.Bharathi, P.Jeyanthi, M.Ramesh	Decision tree, Random Forest and Support Vector Machine learning models are constructed to carry out the diagnosis of CKD	99.16%	2019
3	Preemptive Diagnosis of Chronic Kidney Disease Using Machine Learning Techniques	Reem A. Alassaf, Khawla A. Alsulaim, Noura Y. Alroomi, Nouf S. Alsharif, Mishael F. Aljubeir, Sunday O. Olatunji, Alaa Y. Alahmadi, Mohammed Imran,	ANN, SVM, Naïve Bayes along with k-NN comparison approach	ANN,SVM ,Naïve Bayes - 98% k-NN - 93.9%	2018

		Rahma A. Alzahrani, Nora S. Alturayeif			
4	Prediction of Chronic Kidney Disease Using Machine Learning Algorithm	Siddheshwar Tekale,Pranjal Shingavi,Sukanya Wandhekar, Ankit Chatorikar	Decision tree algorithms along comparison with SVM	Decision tree – 91.75% SVM-96.75%	2018
5	Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study	Njoud Abdullah Almansour,Hajra FahimSyed, Nuha Radwan Khayat,Rawan KanaanAltheeb,Renad Emad Juri,Jamal Alhiyafi,Sale	Comparative analysis was carried out on the two models-ANN and SVM	ANN - 99.75% SVM - 97.75%	2019

		h Alrashed,Sunday O.Olatunji			

2.3 Problem Statement Definition

Chronic kidney disease (CKD) is one of the most critical health problems due to its increasing prevalence. Chronic kidney disease, also known as chronic renal disease or CKD, is a condition characterized by a gradual loss of kidney function over time. It includes conditions that damage your kidneys and decrease their ability to keep you healthy by filtering wastes from your blood. Diabetes and high blood pressure, or hypertension, are responsible for two-thirds of chronic kidney disease cases. Anyone can get chronic kidney disease at any age. However, some people are more likely than others to develop kidney disease. Most people may not have any severe symptoms until their kidney disease is advanced.

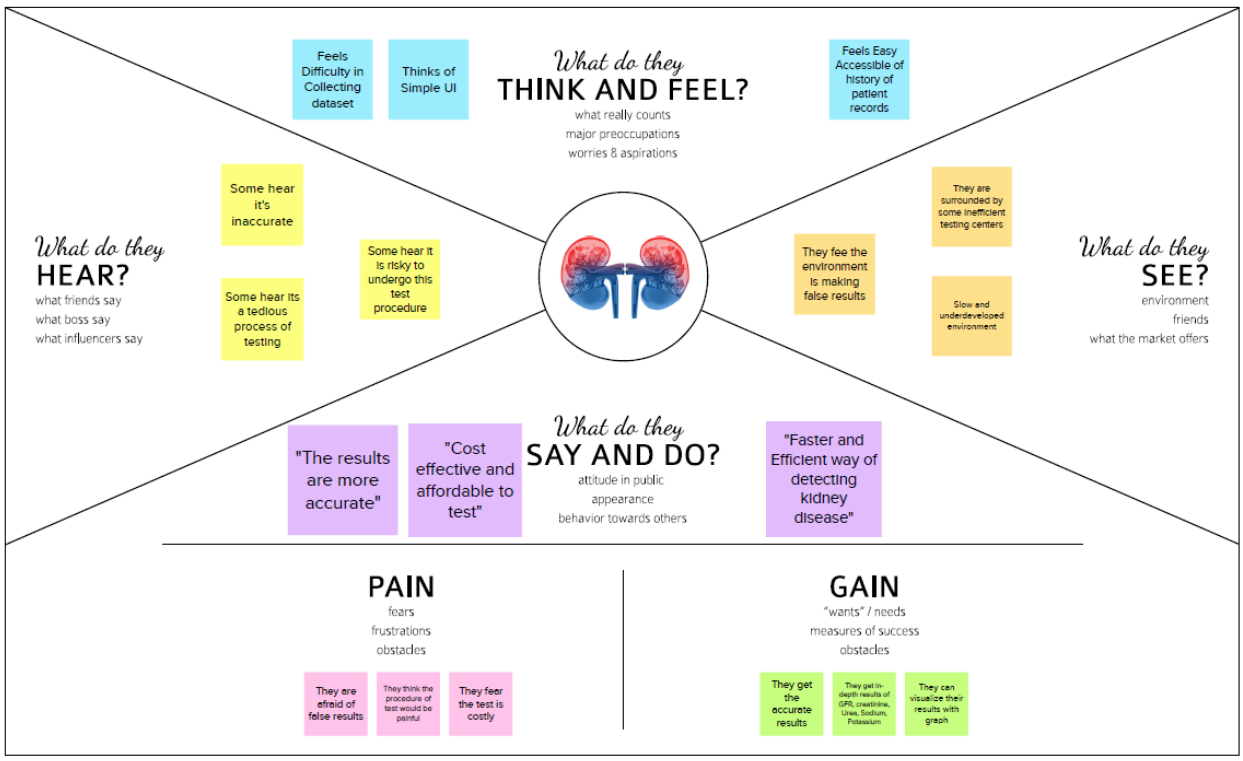
A better testing method which could possibly detect CKD in the early stages would be much more useful. Medical test results taken for other purposes are used to detect CKD at early

stages. 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. Peculiar and contributing attributes from the above-mentioned test results are combined to develop a Machine Learning Model.

This Machine Learning Model will be used to predict CKDs rather early than the presently existing methods.

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas:-



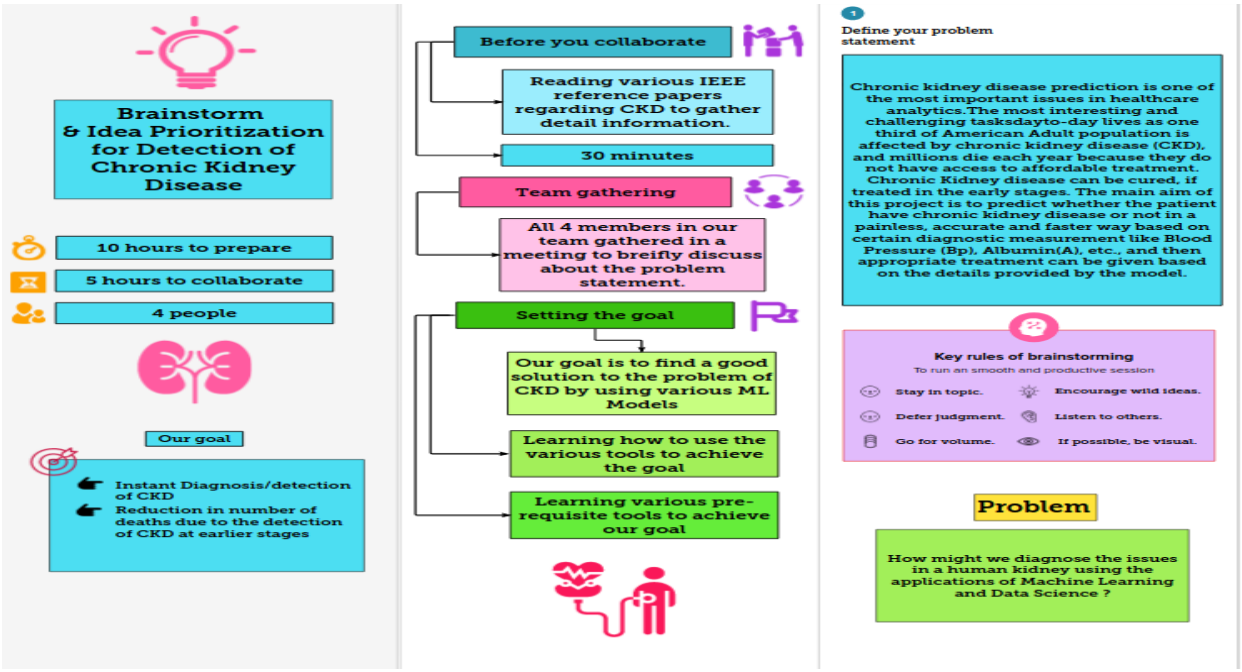
3.2 Ideation & Brainstorming:-

Brainstorming is an activity that will help you generate more innovative ideas. It's one of many methods of ideation—the process of coming up with new ideas—and it's core to the design thinking process.

Brainstorming refers to a problem-solving technique used by teams or individuals. In this process, participants generate various ideas or solutions, then begin discussing and narrowing them down to the best options.

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that **ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity**.

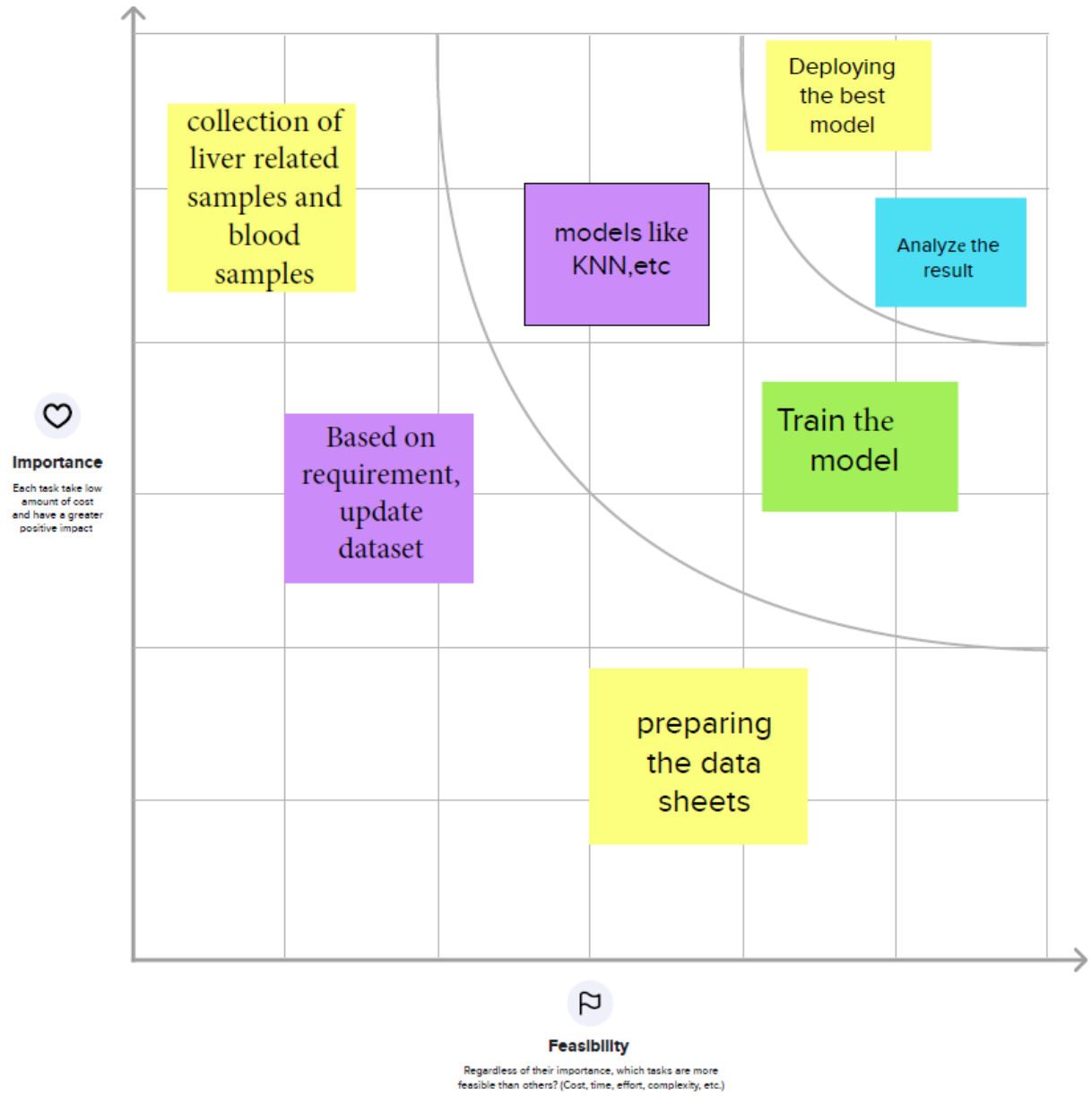
Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brainwriting, Worst Possible Idea, and a wealth of other ideation techniques.


Brainstorm & Idea Prioritization for Detection of Chronic Kidney Disease
10 hours to prepare
5 hours to collaborate
4 people
Our goal
Instant Diagnosis/detection of CKD
Reduction in number of deaths due to the detection of CKD at earlier stages
Before you collaborate
Reading various IEEE reference papers regarding CKD to gather detail information.
30 minutes
Team gathering
All 4 members in our team gathered in a meeting to breifly discuss about the problem statement.
Setting the goal
Our goal is to find a good solution to the problem of CKD by using various ML Models
Learning how to use the various tools to achieve the goal
Learning various pre-requisite tools to achieve our goal
1
Define your problem statement
Chronic kidney disease prediction is one of the most important issues in healthcare analytics.The most interesting and challenging tasksdayto-day lives as one third of American 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 this 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(A), etc., and then appropriate treatment can be given based on the details provided by the model.
Key rules of brainstorming
To run an smooth and productive session
Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.
Problem
How might we diagnose the issues in a human kidney using the applications of Machine Learning and Data Science ?


2
Brainstorm
Write down any ideas that come to mind that address your problem statement.
10 minutes
Aswath Selvam
collection of liver related samples and blood samples
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 and 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

Stage 4 - prioritize



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 customer-specific 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.

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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
2.	Idea / Solution description	The idea of approaching the problem is by creating a suitable machine learning model which involves deep understanding of the data which needs to be collected from real time , handle the missing data and standardizing the data by preprocessing technique which makes it suitable for ml model training and prediction using different approach of model creation depending on the dataset and output
3.	Novelty / Uniqueness	• Easy to use User interface (UI) • accurate accuracy by comparing the performance of different ml model technique
4.	Social Impact / Customer Satisfaction	• Greater cost reduction in hospitals for testing • Helps in early diagnosis of the disease • Chances of recovery is higher
5.	Business Model (Revenue Model)	• subscription based model with initial trial basis • charges/commission for the actual prediction and recovery of a person
6.	Scalability of the Solution	• The server in which the app is deployed containing the ml model must be capable of handling concurrent request and handle multiple request

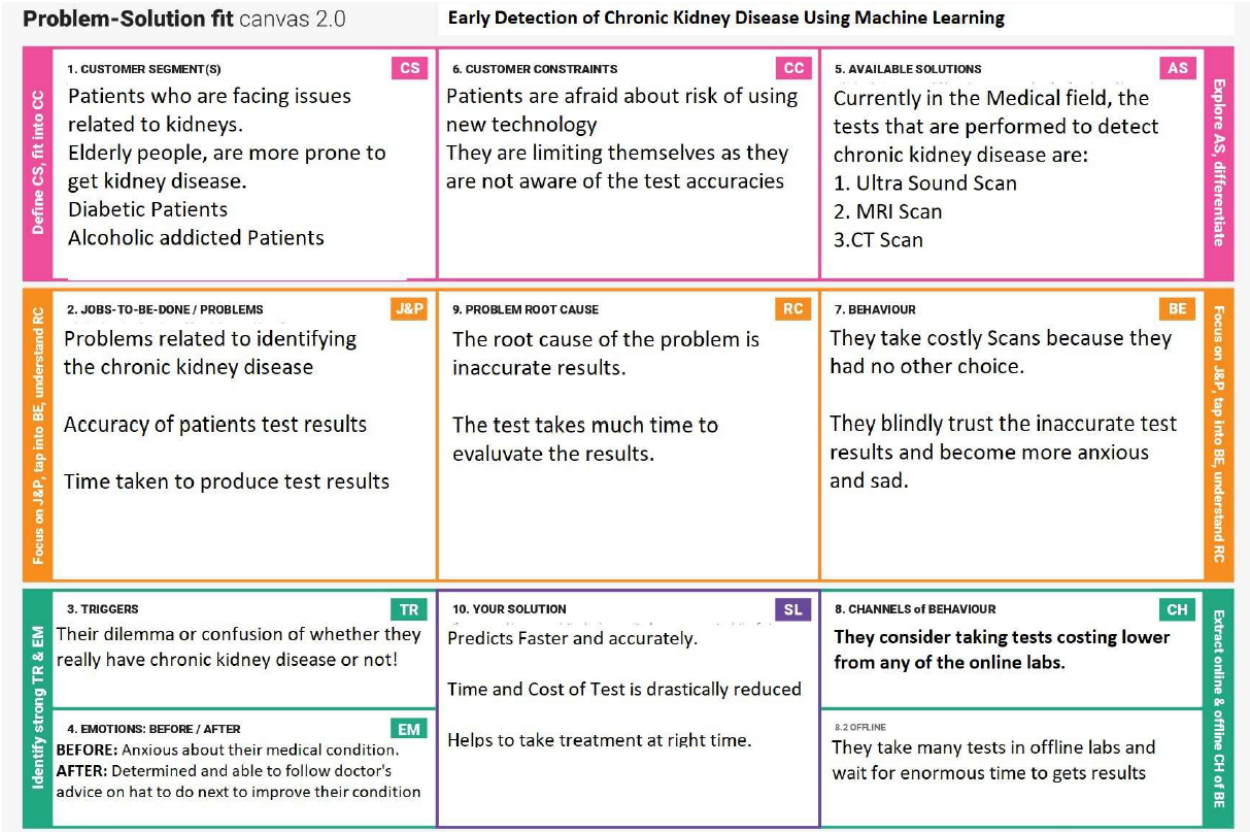
BUSINESS MODEL

Business Model				
Key Partners	**Key Activities**	**Value Propositions**	**Customer Relationships**	**Customer Segments**
Our Key partners have a latest testing laboratory to collect samples of blood and urine to analyse various parameters with a testing capacity of almost 1000 persons a day. Our ML Models can test a large number of samples and give accurate output within a short period of time.	Our key activities are to find out whether persons have chronic kidney disease or not.	We are trying to solve chronic kidney disease at an early stage thereby helping them to recover at a faster rate. We are targeting persons liable to get chronic kidney disease using statistical analysis of the collected data.	We need to have a cordial relationship with the persons coming forward for giving blood test. They expect accurate result from our ML model. The cost of building the model may outfit the early detection of chronic kidney disease.	We are creating value for humans. Our customers are common people who have work culture or wrong lifestyle habits which may lead to chronic kidney disease. Customer base will be a mass market as everyone will give importance to health. We will also be providing a diet regulatory counselling to them.
	Key Resources		**Channels**	
	Our key resources are testing laboratories, ML Models and their data. To create an excel sheet from the data.		Need to reach our patients through doctors. Using ML model to suggest best practices which can help doctors and patients in avoiding and recovering chronic kidney disease.	
Cost Structure		**Revenue Streams**		
Most important cost is mainly for data collection through laboratories. This model is cost driven as it has maximum automation with only a few questions needed to be asked to patients. As compared to older methods, ML Detection technique is cost effective.		Patient's visiting doctors are willing to pay for knowing if they have chronic kidney disease or not. Overall revenue is good and it depends on the accuracy of ML model which will help the doctors to do a correct prediction.		

3.4 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.



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements:-

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Verification	Confirmation via Email
FR-3	User Login	Login through Email
FR-4	User Help	Report issues through Email
FR-5	Disease prediction	User can predict the disease based on the valid inputs given by him/her

4.2 Non- Functional Requirements:-

Non-functional Requirements:

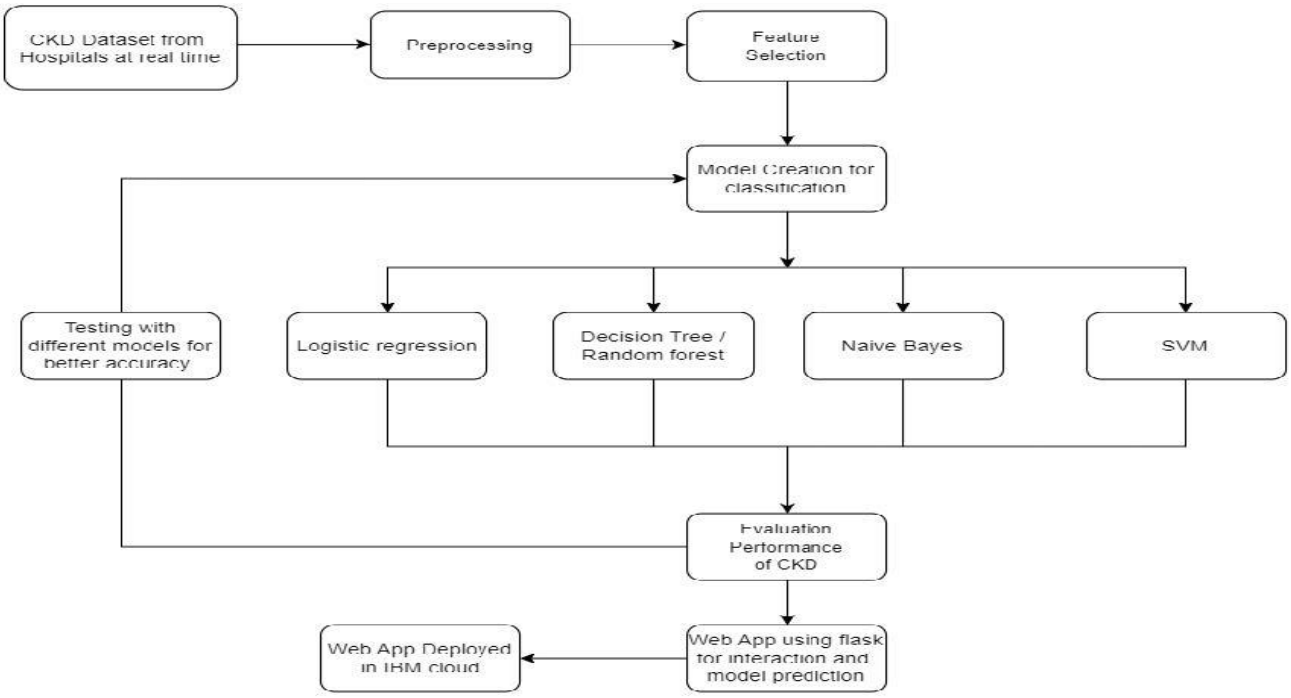
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	In terms of accuracy and efficiency, this project is the best for predicting kidney disease
NFR-2	**Security**	It is highly secured as there is separate user id and password to login
NFR-3	**Reliability**	It is reliable as it is user friendly and cost-efficient
NFR-4	**Performance**	In terms of time, this project is highly efficient as we have used very efficient algorithms like regression
NFR-5	**Availability**	Since it is web based one, it is available to those who have access to web
NFR-6	**Scalability**	If user demand increases, it can be scaled up easily as we made everything dynamic

5. PROJECT DESIGN

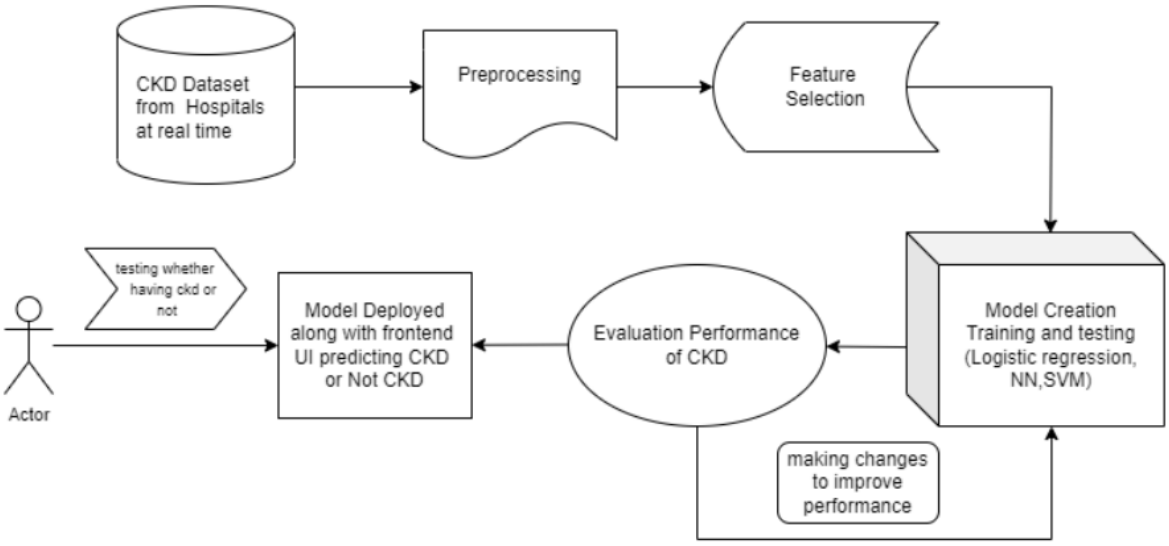
5.1 Data Flow Diagrams:-

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.



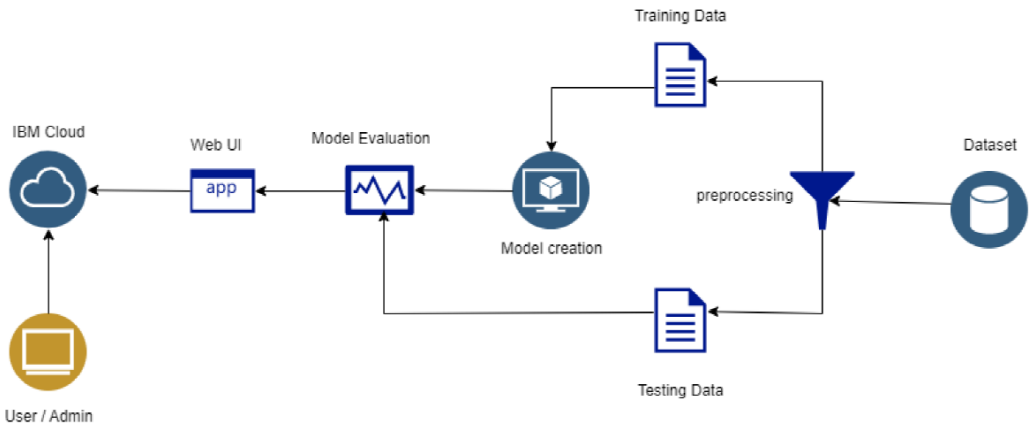
4.2 Solution & Technical Architecture: -

Solution Architecture:



Technical Architecture:

Technical Architecture:



5.3 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 application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Verification	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	As a user, I can log into the application by entering email & password	I am authorized user to avail the web service	High	Sprint-1
	Dashboard	USN-4	As a user, I can navigate and interact with the web app to provide inputs for prediction and testing	I am entitled to enter only valid input for prediction	High	Sprint-1
Customer Care Executive	Assist	USN-5	Collecting the issues and reports from the user through various method of communication	The report or issue must be valid and fully verified	High	Sprint-2
Administrator	Manage	USN-6	Management head controlling all the web services as well as assigning task to improve the service	Complete proper working of web service including security aspect	High	Sprint-3

6. PROJECT PLANNING AND 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.

Technical Architecture:

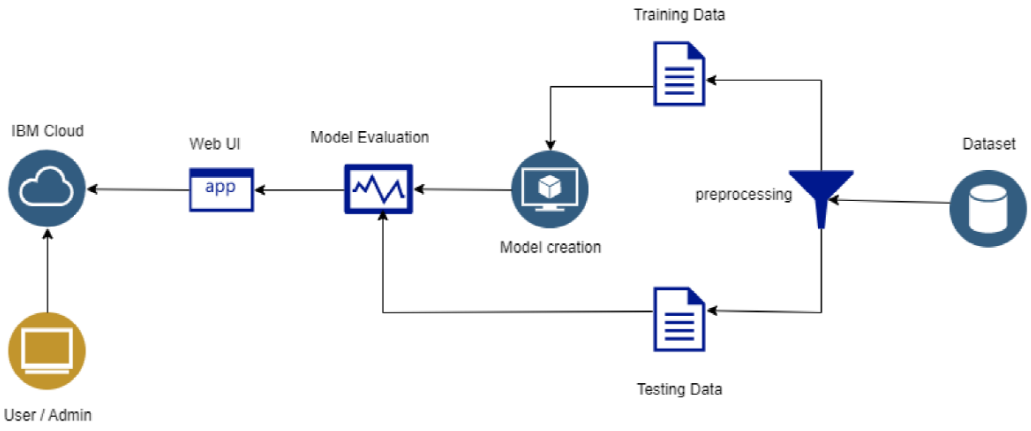


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	User Interact with our application through web user interface	HTML, CSS , Flask , *React (Subsidiary)*
2.	Application Logic-1 (Registration)	User is redirected to register page for registering themselves by providing valid details	HTML, CSS , Flask , *React (Subsidiary)*
3.	Application Logic-2 (Login)	Once the user is registered he is now able to login to access the web service . There is an external login button to redirect to login page	HTML, CSS , Flask , *React (Subsidiary)*
4.	Application Logic-3 (Test / Prediction)	The test or prediction page is present for the user who has logged in and can predict the disease by providing input in the form.	HTML, CSS , Flask , *React (Subsidiary)*
5.	Database	Data Type - String , Numbers	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Role Based Access is provided for using the API	Backend API
9.	External API-2	Purpose of External API used in the application	NIL
10.	Machine Learning Model	To predict the output based on the training and testing of the data from dataset	Data Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	NIL

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Frameworks are used in both in making web app and for model creation	Flask (micro web framework) , python (web framework) , Scikit-learn (ML framework)
2.	Security Implementations	passwords are hashed for the user , as well roles are provided for access based control system	SHA 256
3.	Scalable Architecture	The Scalability can be improvised by using three-tier architecture	Three tier architecture
4.	Availability	Scalability includes availability,the service must be available even if there are more user request ,load balancer is needed to do the above task	Load Balancer
5.	Performance	Performance is key for increased revenue , handling multiple requests and expanding it can be done using Load Balancer.	Load Balancer

6.2 Sprint Estimation and Delivery Schedule:

A sprint estimation shows how much effort a series of tasks require. It's based on assumptions, requirements, and dependencies of a project.

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	Collecting the dataset	USN-1	Collected dataset from Kaggle related to our problem statement.	2	High	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-1	Cleaning the dataset	USN-2	Cleaning dataset by handling and replacing missing values and making effective for prediction.	3	High	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-1	Model Building	USN-3	Building the model and selecting best ML model based on accuracy for accurate prediction.	4	High	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Model Evaluation	USN-4	Evaluating the model and finding accuracy for the model.	4	High	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-2	User Registration	USN-5	As a user, I can register for the application by entering my email, password, and confirming my password.	2	Medium	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-2	User Login	USN-6	As a user, I can log into the application by entering email & password	2	Medium	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-2	User Verification	USN-7	Verifying the user through email.	1	Low	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-2	Dashboard	USN-8	Designing HTML Dashboard page to navigate for the logged in users.	3	Medium	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-3	Integration	USN-9	Using Flask to integrate user dashboard with trained model to predict and request result.	4	High	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Train ML Model in IBM	USN-10	The ML model will be trained in IBM by hosting ipynb file in IBM Cloud.	3	High	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-3	Integrating with IBM Cloud	USN-11	Integrating trained IBM Cloud Model with scoring endpoints using flask	4	High	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-4	Deployment	USN-12	Deploying the IBM model backend flask and frontend application in Cloud	3	Medium	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S
Sprint-4	Further Classification	USN-13	Getting user experience feedback and improving the model and application through customer feedback	2	Medium	Periyakaruppan N Yashwant C M Venkatesh N Ashwath S

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	13	6 Days	24 Oct 2022	29 Oct 2022	13	29 Oct 2022
Sprint-2	8	6 Days	31 Oct 2022	05 Nov 2022	8	05 Nov 2022
Sprint-3	11	6 Days	07 Nov 2022	12 Nov 2022	11	07 Nov 2022
Sprint-4	5	6 Days	14 Nov 2022	19 Nov 2022	5	14 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)

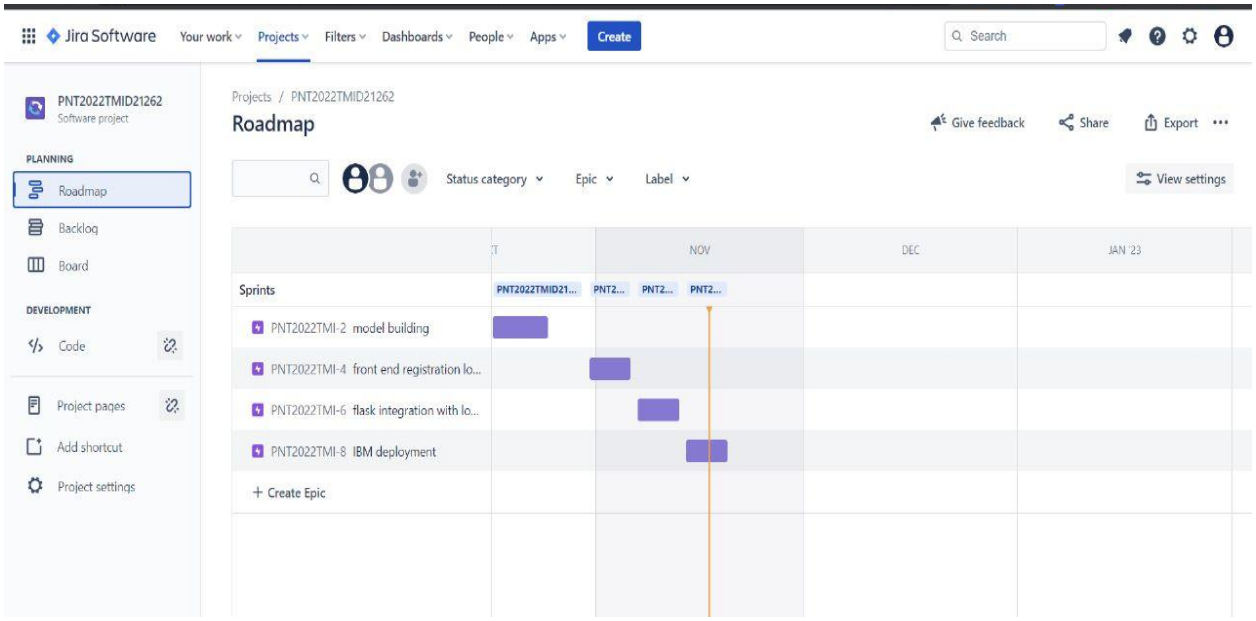
Sprint 1 AV = Sprint duration/velocity = 13/6 = 2.16

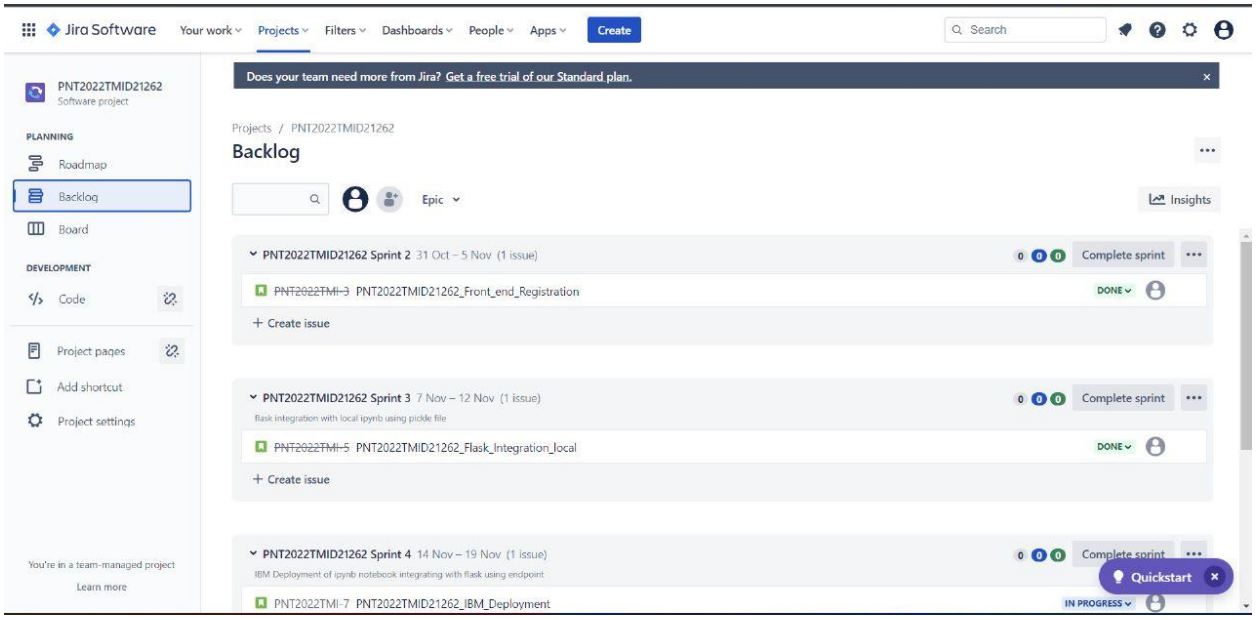
Sprint 2 AV = Sprint duration/velocity = 8/6 = 1.33

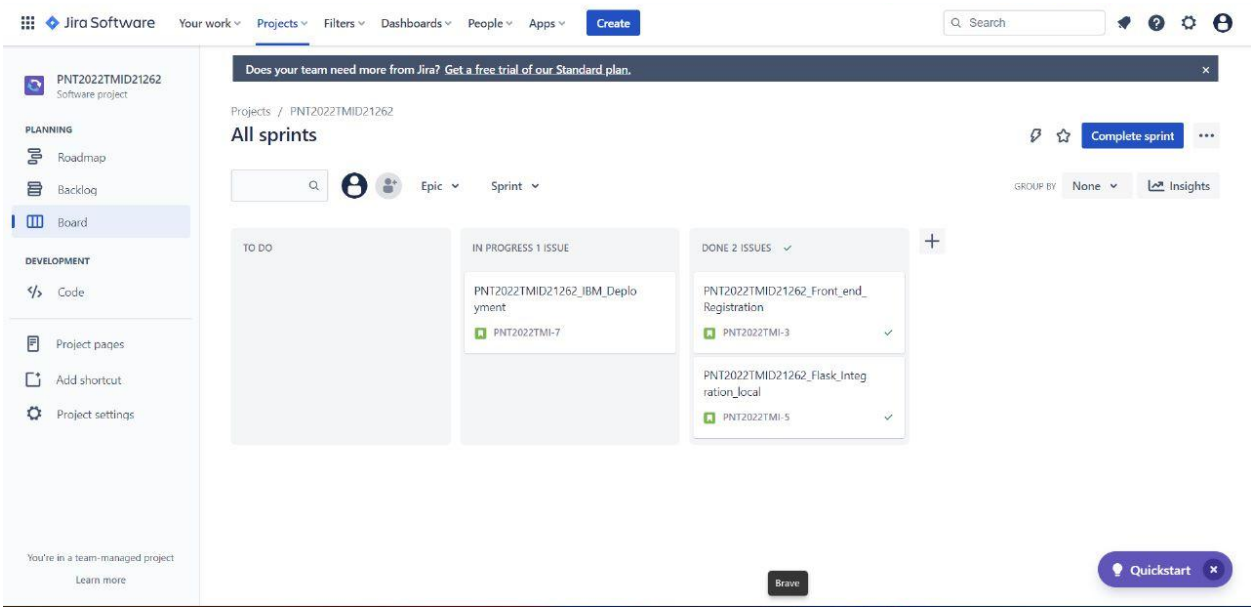
Sprint 3 AV = Sprint duration/velocity = 11/6 = 1.83

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

6.3 Reports from JIRA:



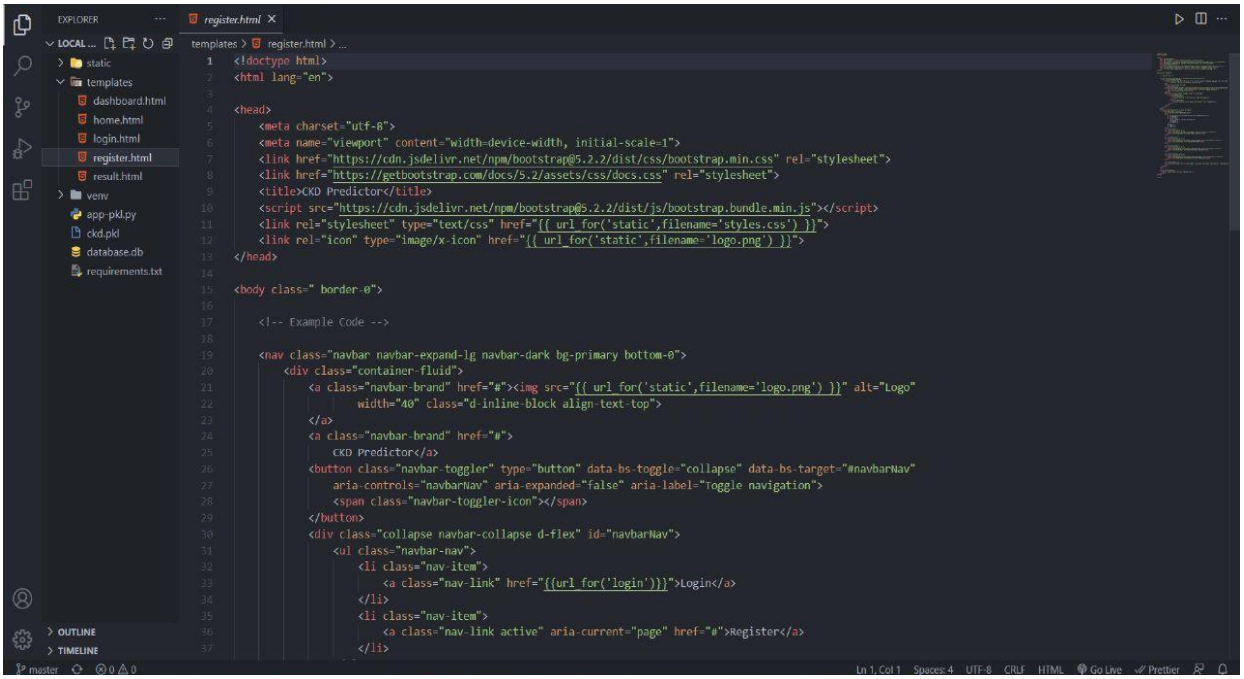


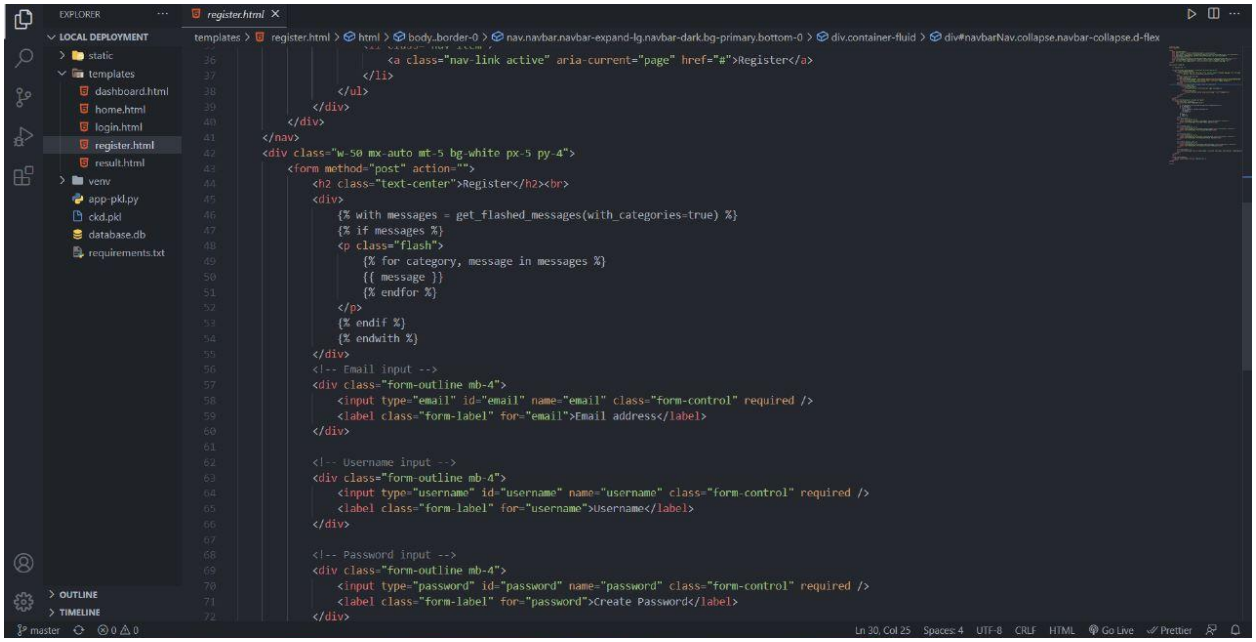


7. CODING & SOLUTIONING

7.1 User Registration and login :

Register.html:




EXPLORER
register.html
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
templates > register.html > html > body_border-0 > nav.navbar.navbar-expand-lg.navbar-dark.bg-primary.bottom-0 > div.container-fluid > div#navbarNav.collapse.navbar-collapse.d-flex
Register

</nav>

<form method="post" action="">
<h2 class="text-center">Register</h2>

{% with messages = get_flashed_messages(with_categories=true) %}
{% if messages %}

{% for category, message in messages %}
{{ message }}
{% endfor %}

{% endif %}
{% endwith %}

<!-- Email input -->

<input type="email" id="email" name="email" class="form-control" required />
<label class="form-label" for="email">Email address</label>

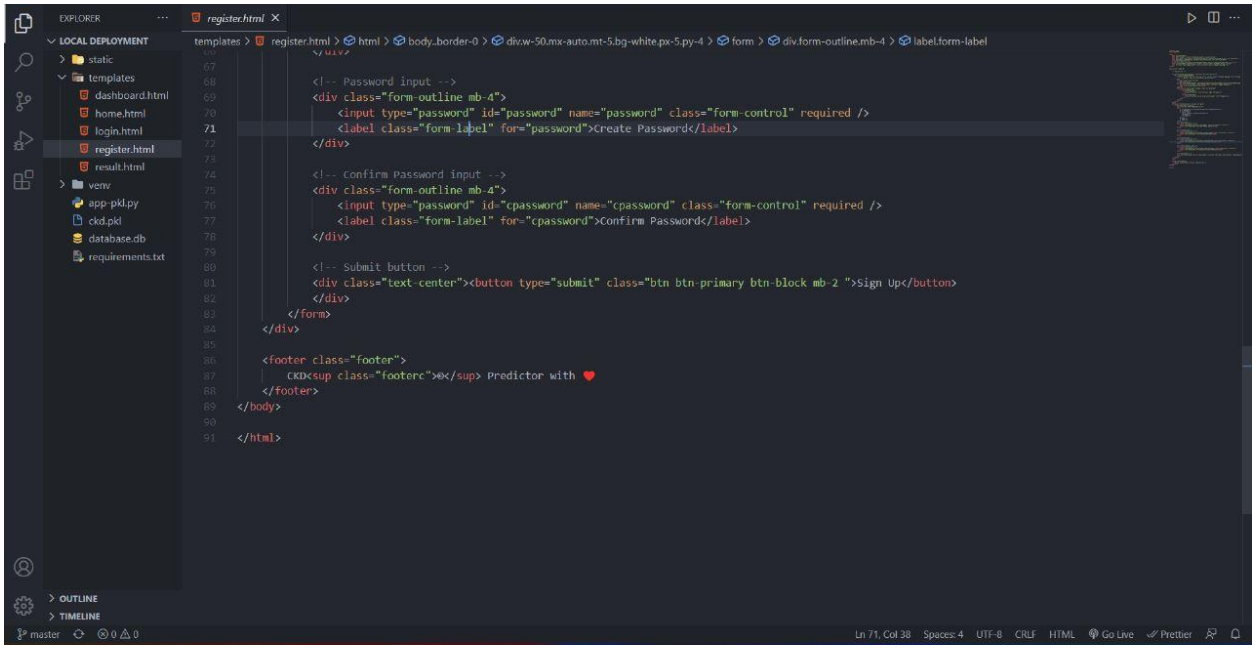
<!-- Username input -->

<input type="username" id="username" name="username" class="form-control" required />
<label class="form-label" for="username">Username</label>

<!-- Password input -->

<input type="password" id="password" name="password" class="form-control" required />
<label class="form-label" for="password">Create Password</label>

master
Ln 30, Col 25
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier


EXPLORER
register.html
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
templates > register.html > html > body_border-0 > div.w-50.mx-auto.mt-5.bg-white.px-5.py-4 > form > div.form-outline.mb-4 > label.form-label
<!-- Password input -->

<input type="password" id="password" name="password" class="form-control" required />
<label class="form-label" for="password">Create Password</label>

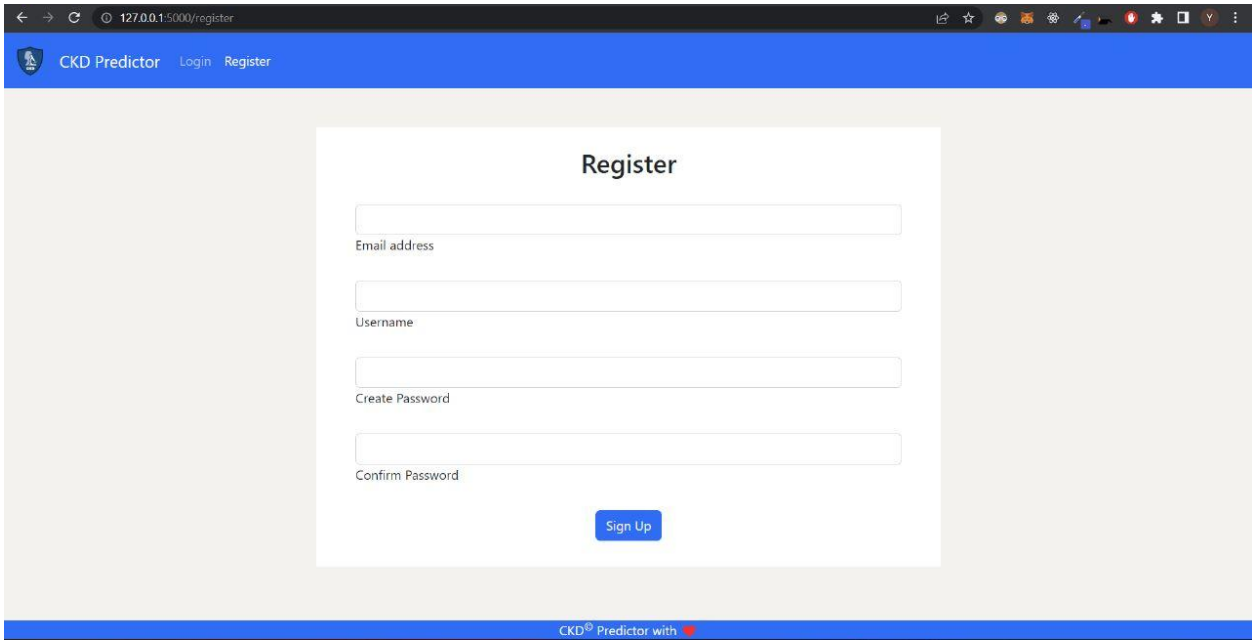
<!-- Confirm Password input -->

<input type="password" id="cpassword" name="cpassword" class="form-control" required />
<label class="form-label" for="cpassword">Confirm Password</label>

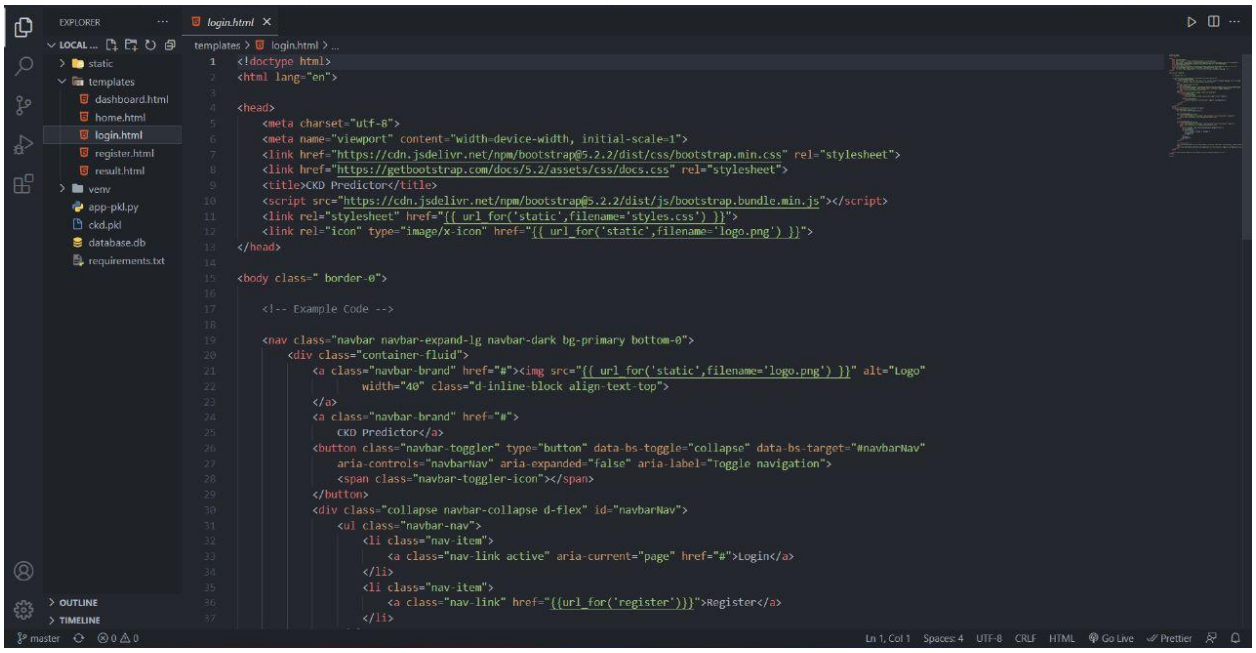
<!-- Submit button -->
<button type="submit" class="btn btn-primary btn-block mb-2 ">Sign Up</button>

</form>

<footer class="footer">
CKD® Predictor with ❤️
</footer>
</body>
</html>
master
Ln 71, Col 38
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier


127.0.0.1:5000/register
CKD Predictor
Login
Register
Register
Email address
Username
Create Password
Confirm Password
Sign Up
CKD© Predictor with

Login.html


EXPLORER
login.html
templates > login.html > ...
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
<link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">
<title>CKD Predictor</title>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>
<link rel="stylesheet" href="{{ url_for('static',filename='styles.css') }}">
<link rel="icon" type="image/x-icon" href="{{ url_for('static',filename='logo.png') }}">
</head>
<body class=" border-0">
<!-- Example Code -->
<nav class="navbar navbar-expand-lg navbar-dark bg-primary bottom-0">

<img src="{{ url_for('static',filename='logo.png') }}" alt="Logo"
width="40" class="d-inline-block align-text-top">

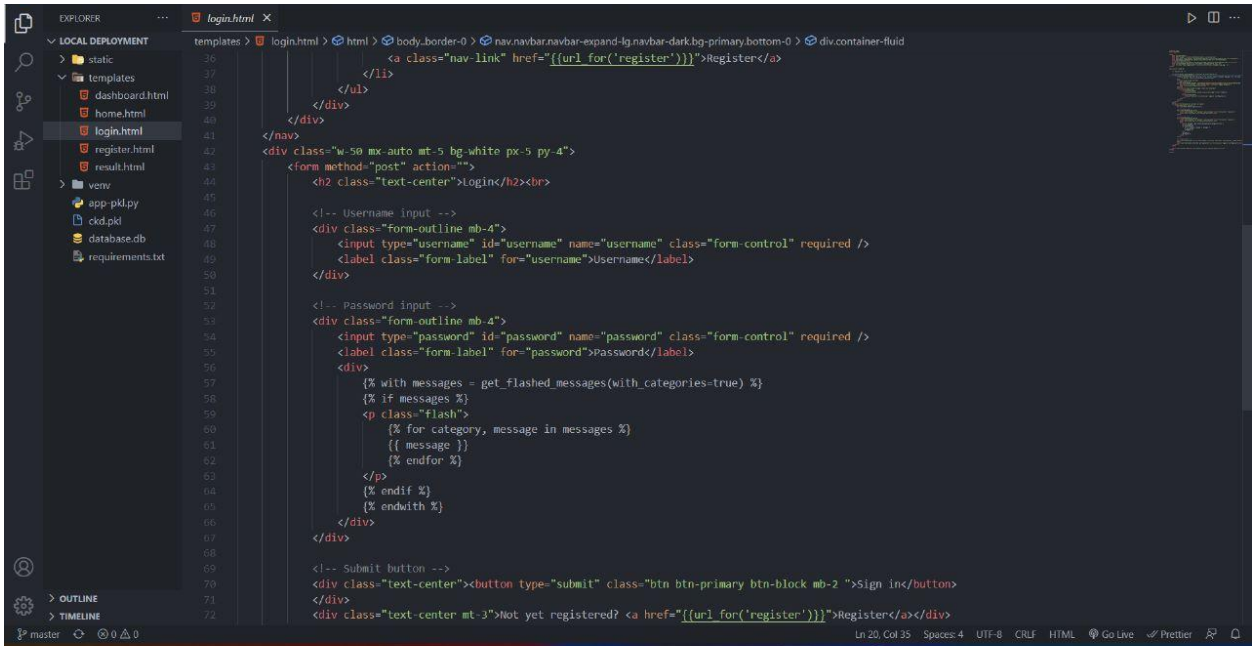
CKD Predictor
<button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

</button>

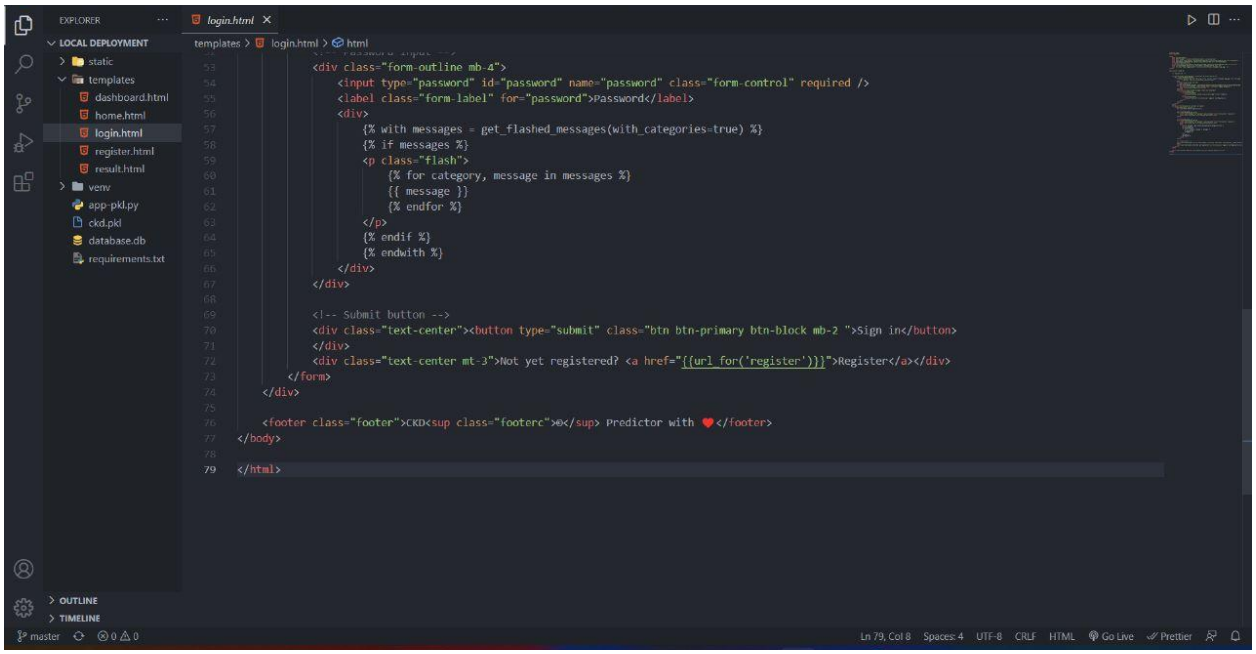
<ul class="navbar-nav">
<li class="nav-item">
Login

<li class="nav-item">
Register

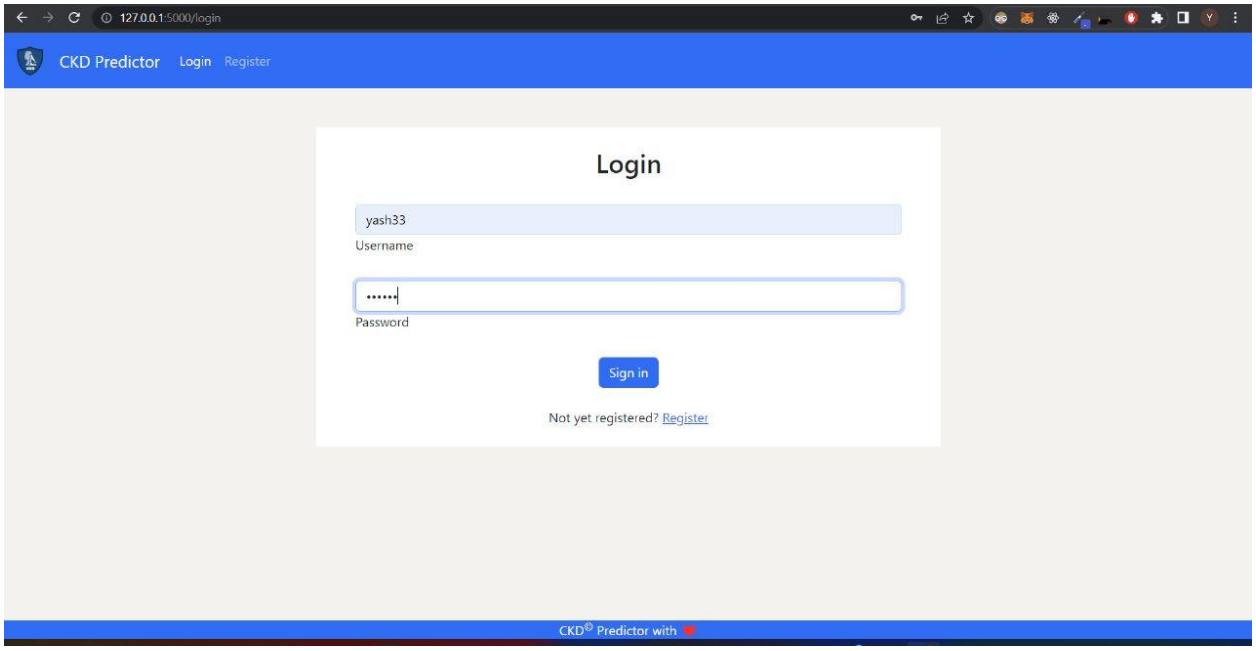
master
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier



```
<a class="nav-link" href="{{url_for('register')}}">Register</a>
</li>
</ul>
</div>
</div>
</nav>
<div class="w-50 mx-auto mt-5 bg-white px-5 py-4">
<form method="post" action="">
<h2 class="text-center">Login</h2><br>

<!-- Username input -->
<div class="form-outline mb-4">
<input type="username" id="username" name="username" class="form-control" required />
<label class="form-label" for="username">Username</label>
</div>

<!-- Password input -->
<div class="form-outline mb-4">
<input type="password" id="password" name="password" class="form-control" required />
<label class="form-label" for="password">Password</label>
<div>
{% with messages = get_flashed_messages(with_categories=true) %}
{% if messages %}
<p class="flash">
{% for category, message in messages %}
{{ message }}
{% endfor %}
</p>
{% endif %}
{% endwith %}
</div>
</div>

<!-- Submit button -->
<div class="text-center"><button type="submit" class="btn btn-primary btn-block mb-2 ">Sign in</button>
</div>
<div class="text-center mt-3">Not yet registered? <a href="{{url_for('register')}}">Register</a></div>
```

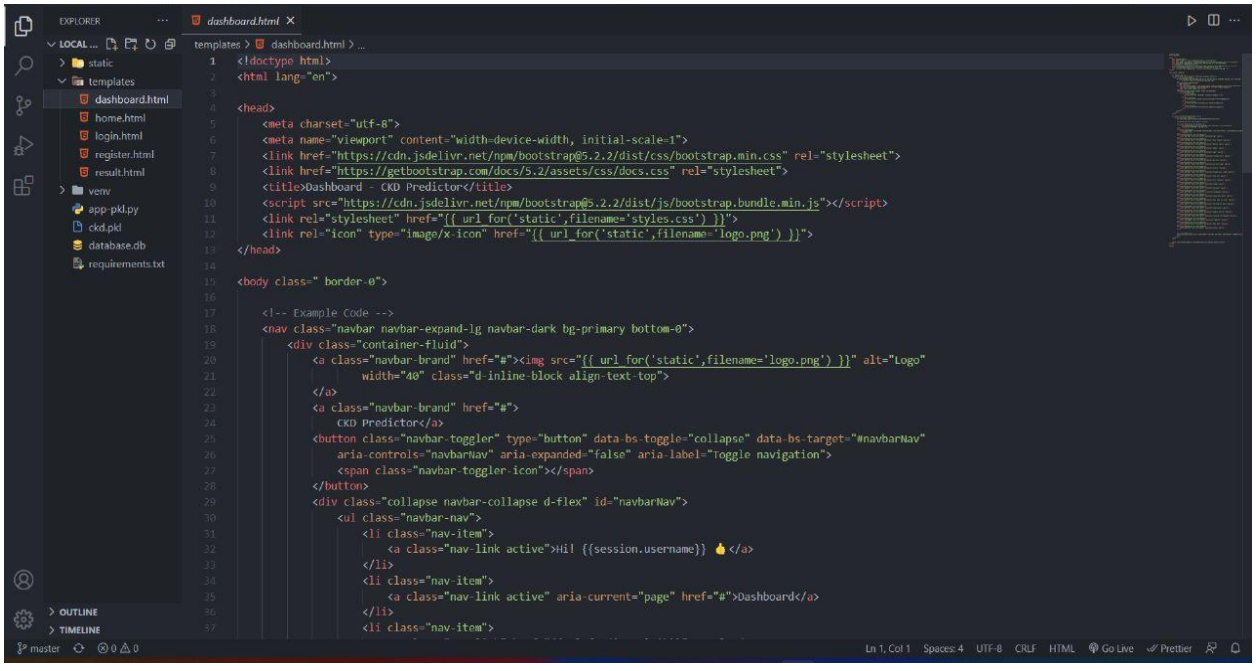


```
<div class="form-outline mb-4">
<input type="password" id="password" name="password" class="form-control" required />
<label class="form-label" for="password">Password</label>
<div>
{% with messages = get_flashed_messages(with_categories=true) %}
{% if messages %}
<p class="flash">
{% for category, message in messages %}
{{ message }}
{% endfor %}
</p>
{% endif %}
{% endwith %}
</div>
</div>

<!-- Submit button -->
<div class="text-center"><button type="submit" class="btn btn-primary btn-block mb-2 ">Sign in</button>
</div>
<div class="text-center mt-3">Not yet registered? <a href="{{url_for('register')}}">Register</a></div>
</form>
</div>

<footer class="footer">CKD<sup class="footerc">©</sup> Predictor with ❤️</footer>
</body>

</html>
```

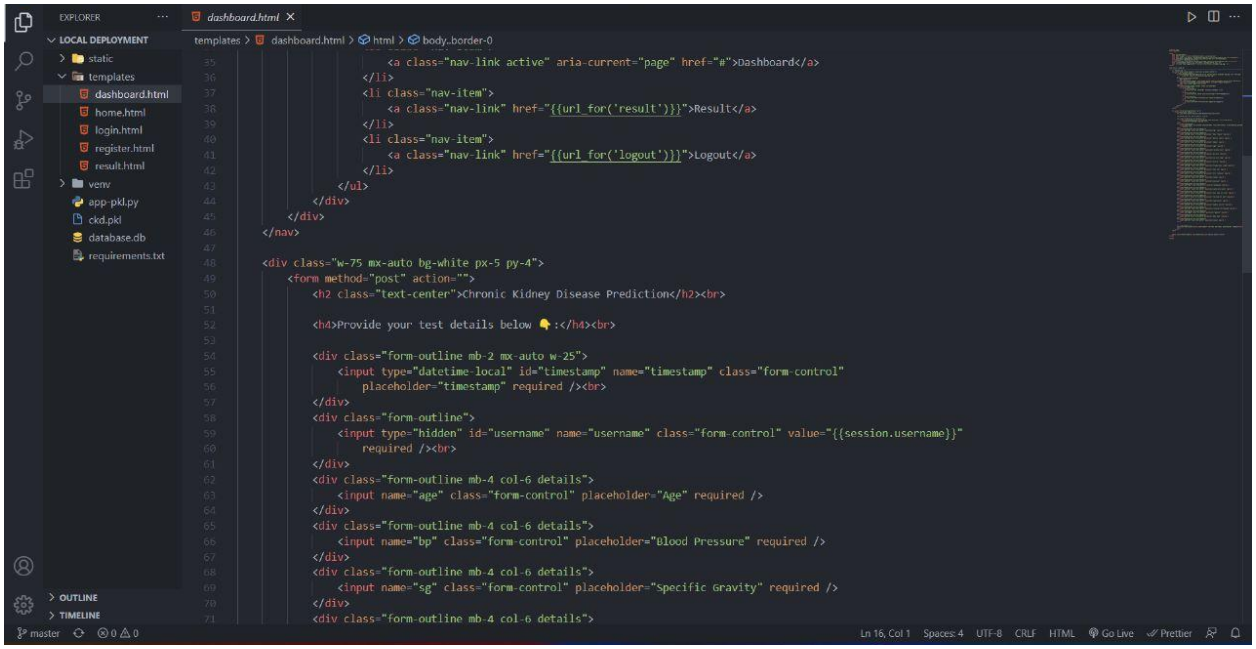


7.2 Dashboard and Result

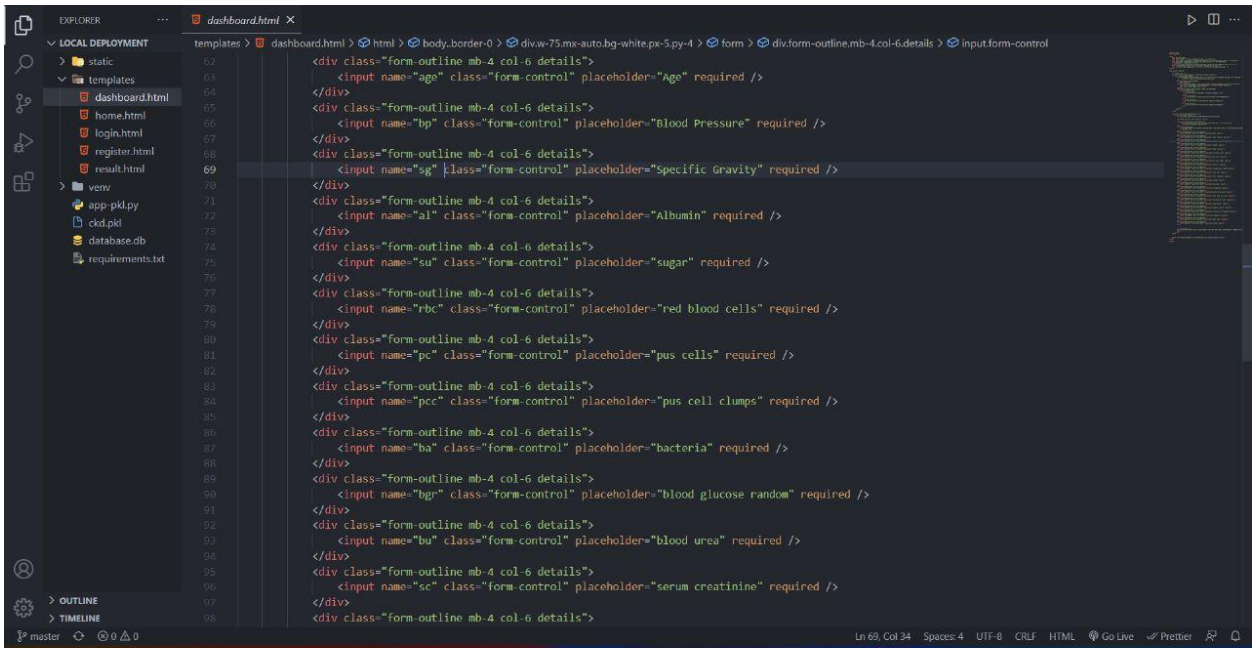
Dashboard.html



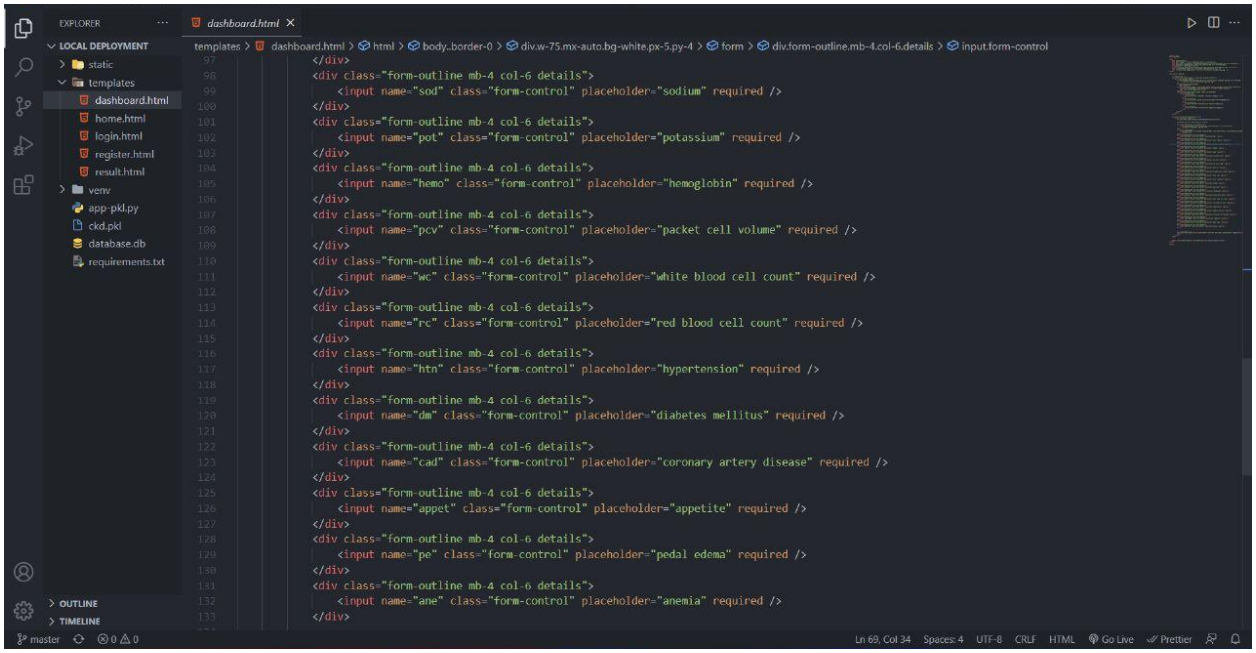

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet">
    <link href="https://getbootstrap.com/docs/5.2/assets/css/docs.css" rel="stylesheet">
    <title>Dashboard - CKD Predictor</title>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>
    <link rel="stylesheet" href="{{ url_for('static',filename='styles.css') }}">
    <link rel="icon" type="image/x-icon" href="{{ url_for('static',filename='logo.png') }}">
</head>

<body class=" border-0">

    <!-- Example Code -->
    <nav class="navbar navbar-expand-lg navbar-dark bg-primary bottom-0">
        <div class="container-fluid">
            <a class="navbar-brand" href="#"><img src="{{ url_for('static',filename='logo.png') }}" alt="Logo"
                    width="40" class="d-inline-block align-text-top">
            </a>
            <a class="navbar-brand" href="#">
                CKD Predictor</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
                aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse d-flex" id="navbarNav">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link active">Hi! {{session.username}} 👍</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="#">Dashboard</a>
                    </li>
                    <li class="nav-item">
```

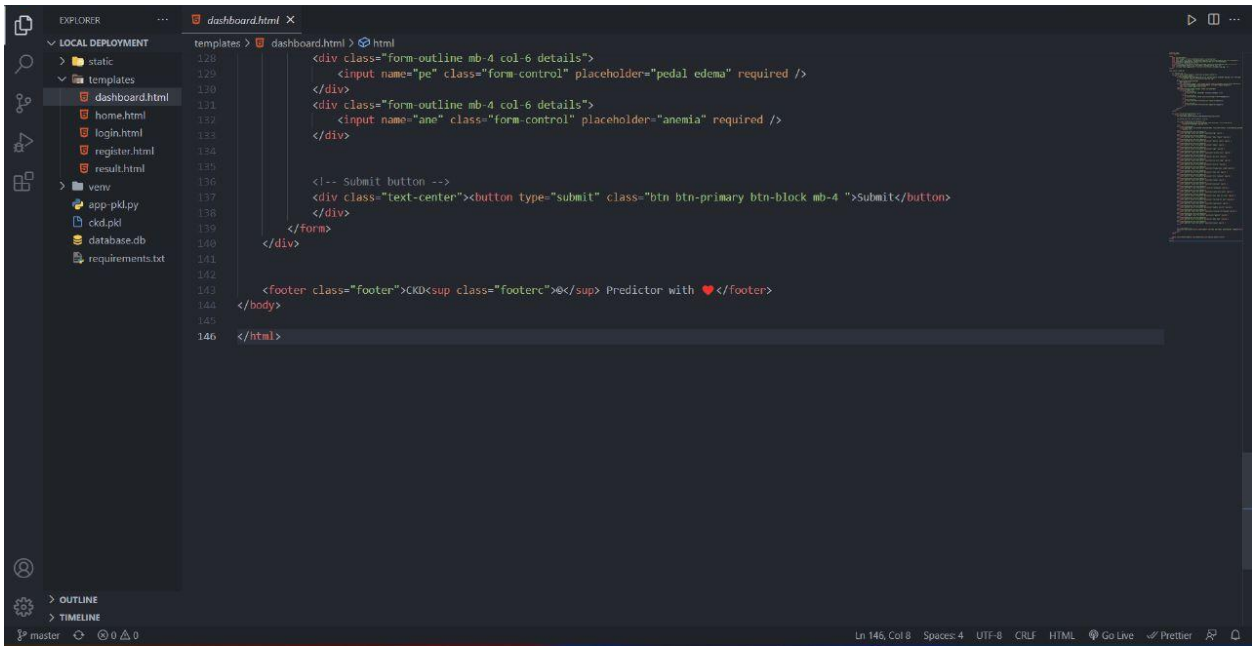


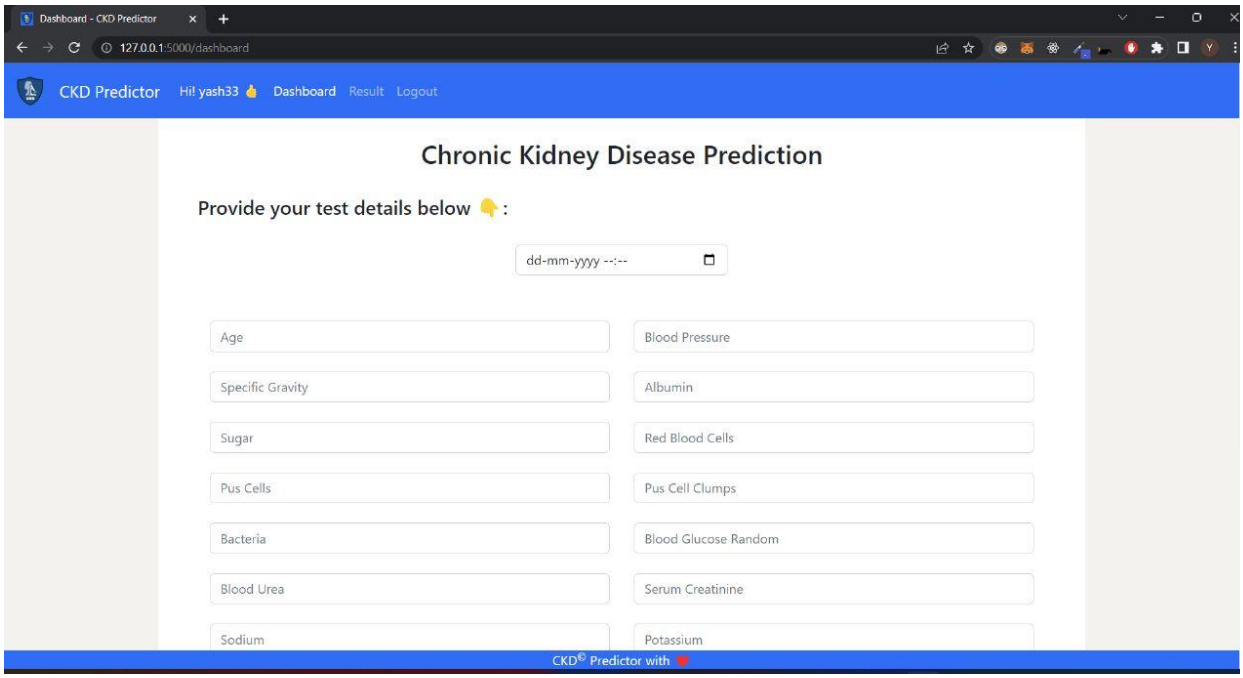

```
                        <a class="nav-link active" aria-current="page" href="#">Dashboard</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{url_for('result')}}">Result</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{url_for('logout')}}">Logout</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <div class="w-75 mx-auto bg-white px-5 py-4">
        <form method="post" action="">
            <h2 class="text-center">Chronic Kidney Disease Prediction</h2><br>

            <h4>Provide your test details below 👇:</h4><br>

            <div class="form-outline mb-2 mx-auto w-25">
                <input type="datetime-local" id="timestamp" name="timestamp" class="form-control"
                    placeholder="timestamp" required /><br>
            </div>
            <div class="form-outline">
                <input type="hidden" id="username" name="username" class="form-control" value="{{session.username}}"
                    required /><br>
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="age" class="form-control" placeholder="Age" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="bp" class="form-control" placeholder="Blood Pressure" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="sg" class="form-control" placeholder="Specific Gravity" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
```



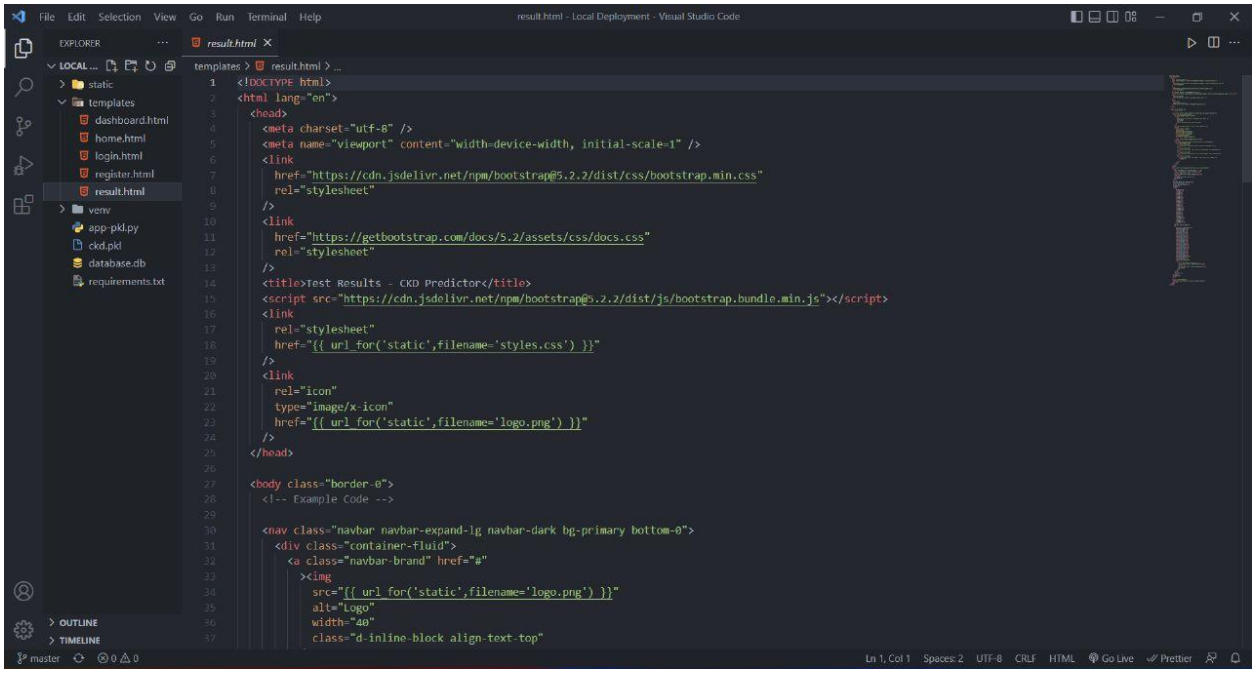

```
            <div class="form-outline mb-4 col-6 details">
                <input name="age" class="form-control" placeholder="Age" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="bp" class="form-control" placeholder="Blood Pressure" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="sg" class="form-control" placeholder="Specific Gravity" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="al" class="form-control" placeholder="Albumin" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="su" class="form-control" placeholder="sugar" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="rbc" class="form-control" placeholder="red blood cells" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="pc" class="form-control" placeholder="pus cells" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="pcc" class="form-control" placeholder="pus cell clumps" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="ba" class="form-control" placeholder="bacteria" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="bgr" class="form-control" placeholder="blood glucose random" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="bu" class="form-control" placeholder="blood urea" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="sc" class="form-control" placeholder="serum creatinine" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
```


EXPLORER
dashboard.html
templates > dashboard.html > html > body..border-0 > div.w-75.mx-auto.bg-white.px-5.py-4 > form > div.form-outline.mb-4.col-6.details > input.form-control
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
master
Ln 69, Col 34
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier

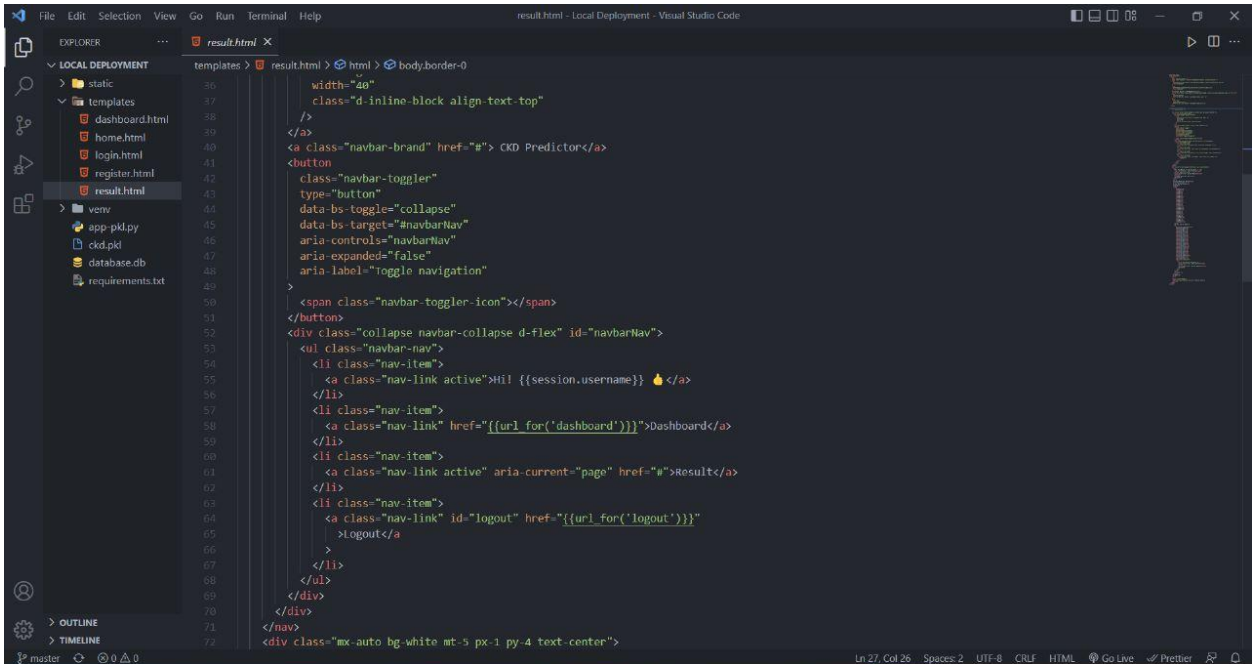

EXPLORER
dashboard.html
templates > dashboard.html > html
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
master
Ln 146, Col 8
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier


Dashboard - CKD Predictor
127.0.0.1:5000/dashboard
CKD Predictor
Hi! yash33
Dashboard
Result
Logout
Chronic Kidney Disease Prediction
Provide your test details below 👇:
dd-mm-yyyy --:--
Age
Blood Pressure
Specific Gravity
Albumin
Sugar
Red Blood Cells
Pus Cells
Pus Cell Clumps
Bacteria
Blood Glucose Random
Blood Urea
Serum Creatinine
Sodium
Potassium
CKD© Predictor with

Result.html


result.html - Local Deployment - Visual Studio Code
EXPLORER
result.html
templates
static
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1" />
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"
/>
<link
href="https://getbootstrap.com/docs/5.2/assets/css/docs.css"
rel="stylesheet"
/>
<title>Test Results - CKD Predictor</title>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>
<link
rel="stylesheet"
href="{{ url_for('static',filename='styles.css') }}"
/>
<link
rel="icon"
type="image/x-icon"
href="{{ url_for('static',filename='logo.png') }}"
/>
</head>
<body class="border-0">
<!-- Example Code -->
<nav class="navbar navbar-expand-lg navbar-dark bg-primary bottom-0">

<a class="navbar-brand" href="#"
><img
src="{{ url_for('static',filename='logo.png') }}"
alt="Logo"
width="40"
class="d-inline-block align-text-top"
OUTLINE
TIMELINE
master
Ln 1, Col 1
Spaces: 2
UTF-8
CRLF
HTML
Go Live
Prettier


File Edit Selection View Go Run Terminal Help
result.html - Local Deployment - Visual Studio Code
EXPLORER
result.html
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
templates > result.html > html > body.border-0
width="40"
class="d-inline-block align-text-top"
/>

 CKD Predictor
<button
class="navbar-toggler"
type="button"
data-bs-toggle="collapse"
data-bs-target="#navbarNav"
aria-controls="navbarNav"
aria-expanded="false"
aria-label="Toggle navigation"
>

</button>

<ul class="navbar-nav">
<li class="nav-item">
Hi! {{session.username}}

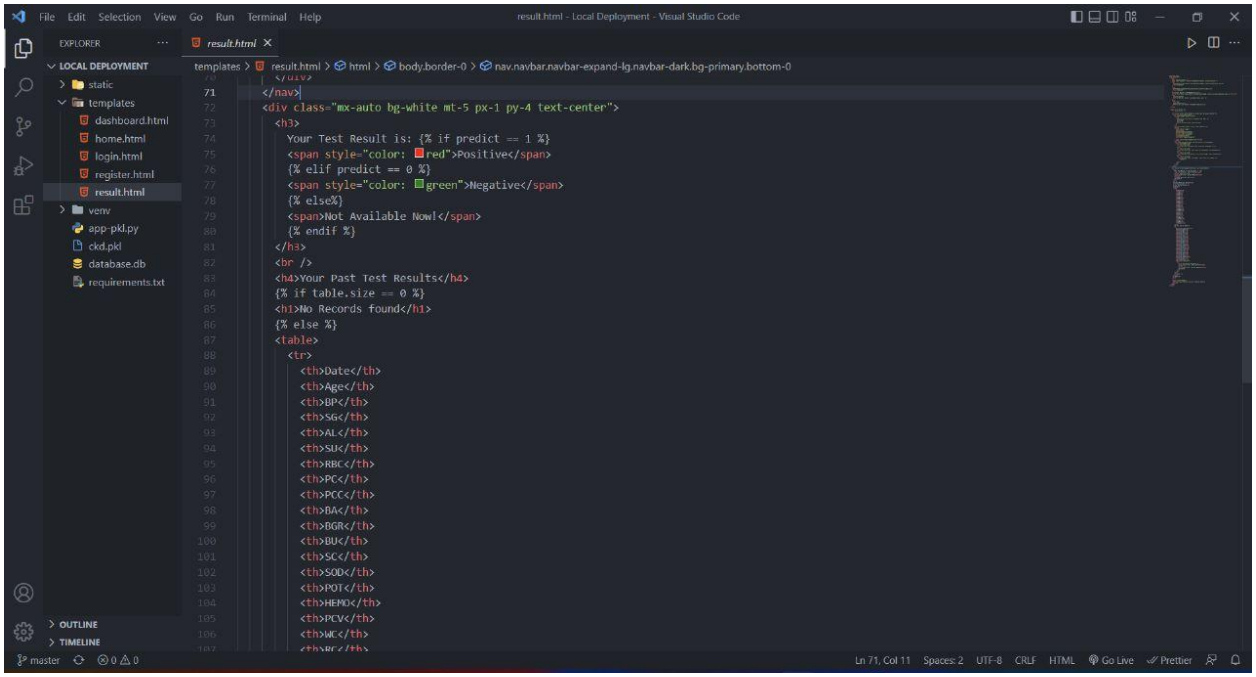
<li class="nav-item">
Dashboard

<li class="nav-item">
Result

<li class="nav-item">
<a class="nav-link" id="logout" href="{{url_for('logout')}}"
>Logout

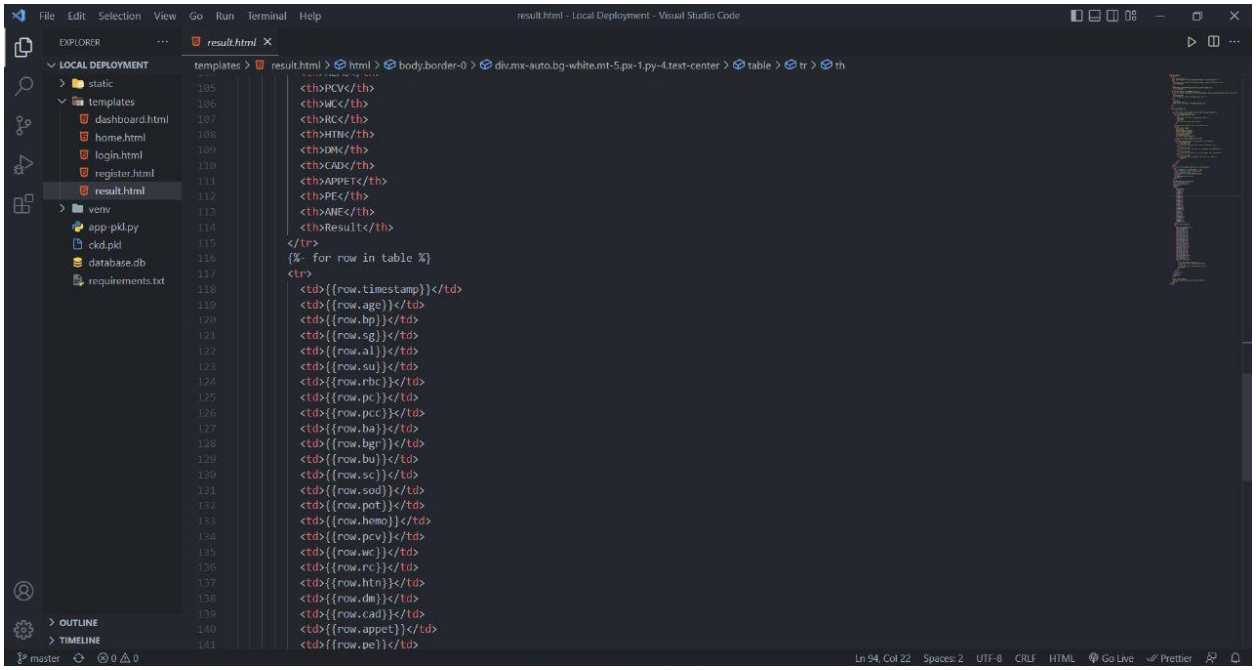
</nav>

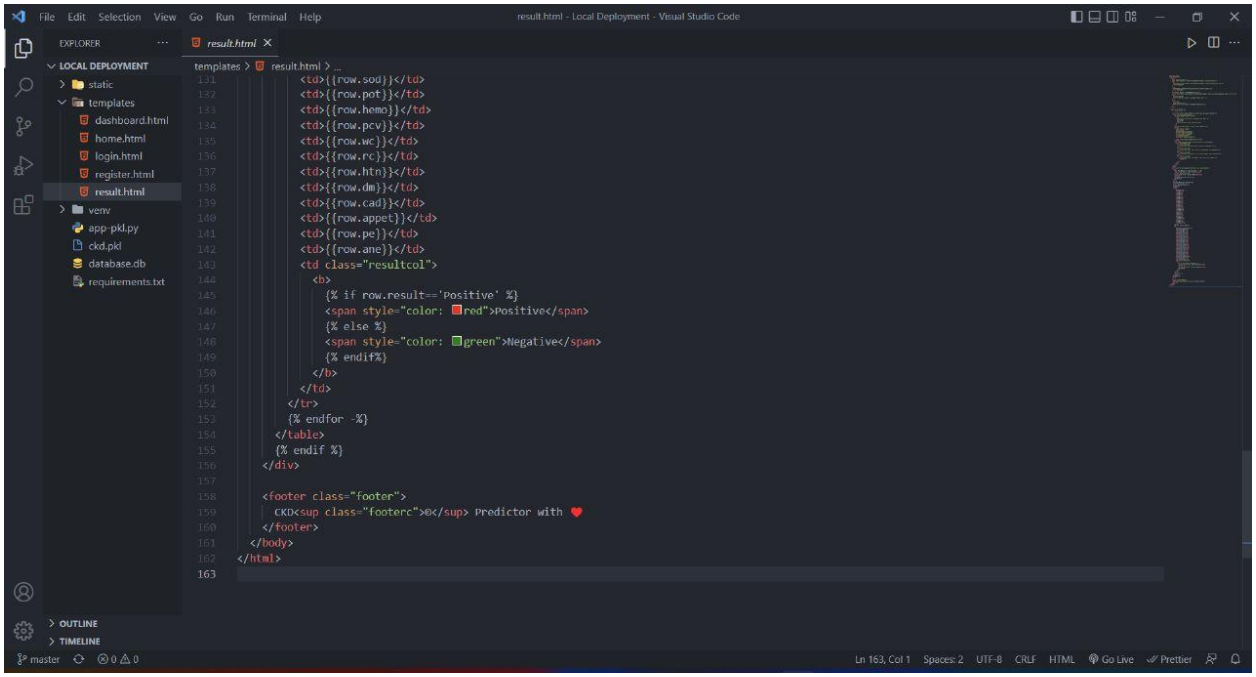
master
Ln 27, Col 26 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
result.html - Local Deployment - Visual Studio Code
EXPLORER
result.html
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
templates > result.html > html > body.border-0 > nav.navbar.navbar-expand-lg.navbar-dark.bg-primary.bottom-0
</nav>

<h3>
Your Test Result is: {% if predict == 1 %}
Positive
{% elif predict == 0 %}
Negative
{% else%}
Not Available Now!
{% endif %}
</h3>

<h4>Your Past Test Results</h4>
{% if table.size == 0 %}
<h1>No Records found</h1>
{% else %}
<table>
<tr>
<th>Date</th>
<th>Age</th>
<th>BP</th>
<th>SG</th>
<th>AL</th>
<th>SU</th>
<th>RBC</th>
<th>PC</th>
<th>PCC</th>
<th>BA</th>
<th>BGR</th>
<th>BU</th>
<th>SC</th>
<th>SOD</th>
<th>POT</th>
<th>HEMO</th>
<th>PCV</th>
<th>WC</th>
master
Ln 71, Col 11 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier

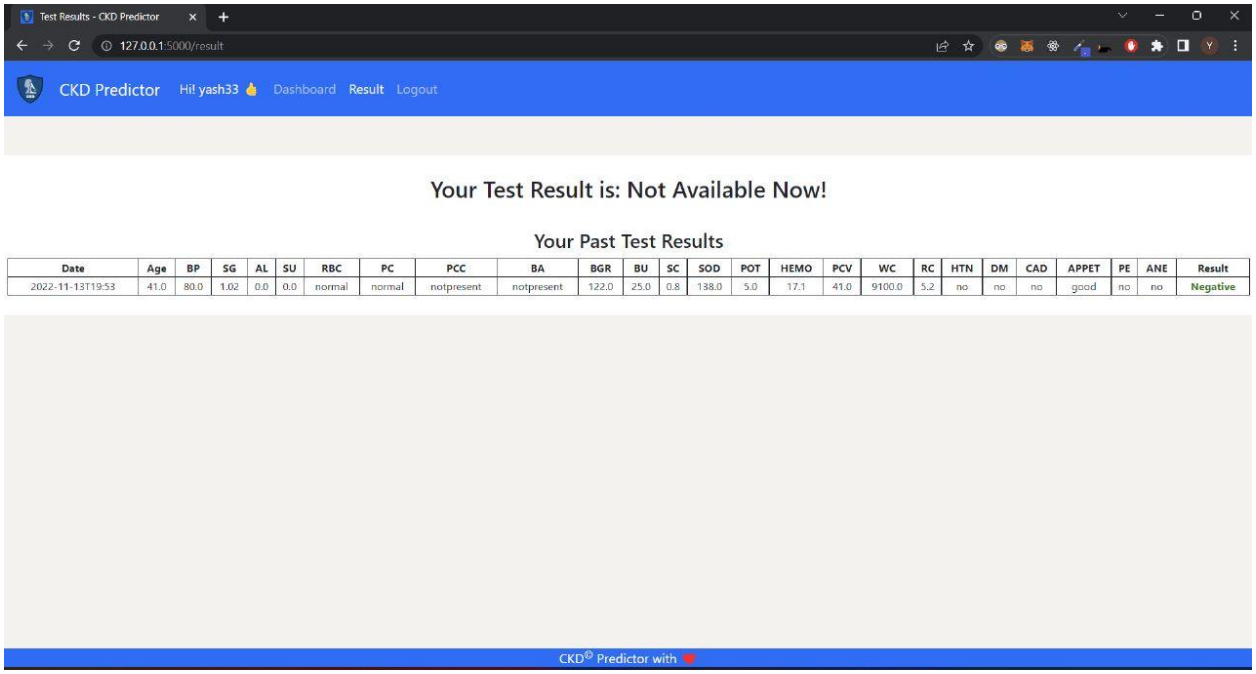

File Edit Selection View Go Run Terminal Help
result.html - Local Deployment - Visual Studio Code
EXPLORER
result.html
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
templates > result.html > html > body.border-0 > div.mx-auto.bg-white.mt-5.px-1.py-4.text-center > table > tr > th
<th>PCV</th>
<th>WC</th>
<th>RC</th>
<th>HTN</th>
<th>DM</th>
<th>CAD</th>
<th>APPET</th>
<th>PE</th>
<th>ANE</th>
<th>Result</th>
</tr>
{%- for row in table %}
<tr>
<td>{{row.timestamp}}</td>
<td>{{row.age}}</td>
<td>{{row.bp}}</td>
<td>{{row.sg}}</td>
<td>{{row.al}}</td>
<td>{{row.su}}</td>
<td>{{row.rbc}}</td>
<td>{{row.pc}}</td>
<td>{{row.pcc}}</td>
<td>{{row.ba}}</td>
<td>{{row.bgr}}</td>
<td>{{row.bu}}</td>
<td>{{row.sc}}</td>
<td>{{row.sod}}</td>
<td>{{row.pot}}</td>
<td>{{row.hemo}}</td>
<td>{{row.pcv}}</td>
<td>{{row.wc}}</td>
<td>{{row.rc}}</td>
<td>{{row.htn}}</td>
<td>{{row.dm}}</td>
<td>{{row.cad}}</td>
<td>{{row.appet}}</td>
<td>{{row.pe}}</td>
OUTLINE
TIMELINE
master
Ln 94, Col 22 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier


File Edit Selection View Go Run Terminal Help
result.html - Local Deployment - Visual Studio Code
EXPLORER
result.html
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
templates > result.html > ...
<td>{{row.sod}}</td>
<td>{{row.pot}}</td>
<td>{{row.hemo}}</td>
<td>{{row.pcv}}</td>
<td>{{row.wc}}</td>
<td>{{row.rc}}</td>
<td>{{row.htn}}</td>
<td>{{row.dm}}</td>
<td>{{row.cad}}</td>
<td>{{row.appet}}</td>
<td>{{row.pe}}</td>
<td>{{row.ane}}</td>
<td class="resultcol">

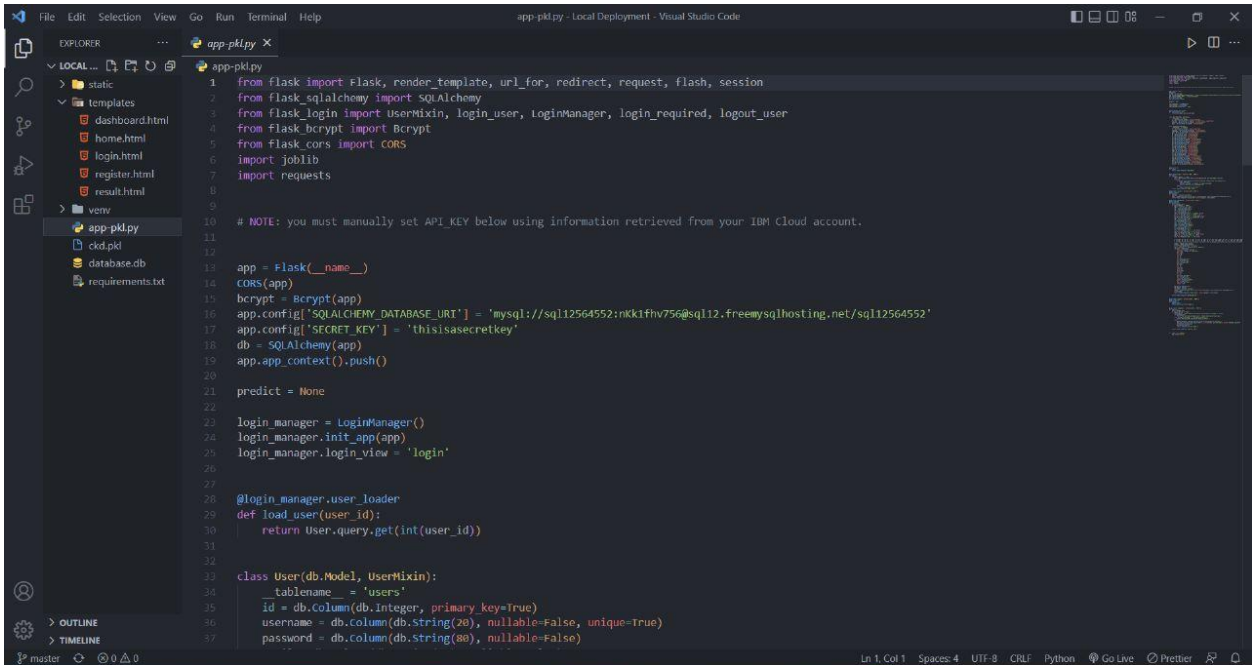
{% if row.result=='Positive' %}
Positive
{% else %}
Negative
{% endif%}

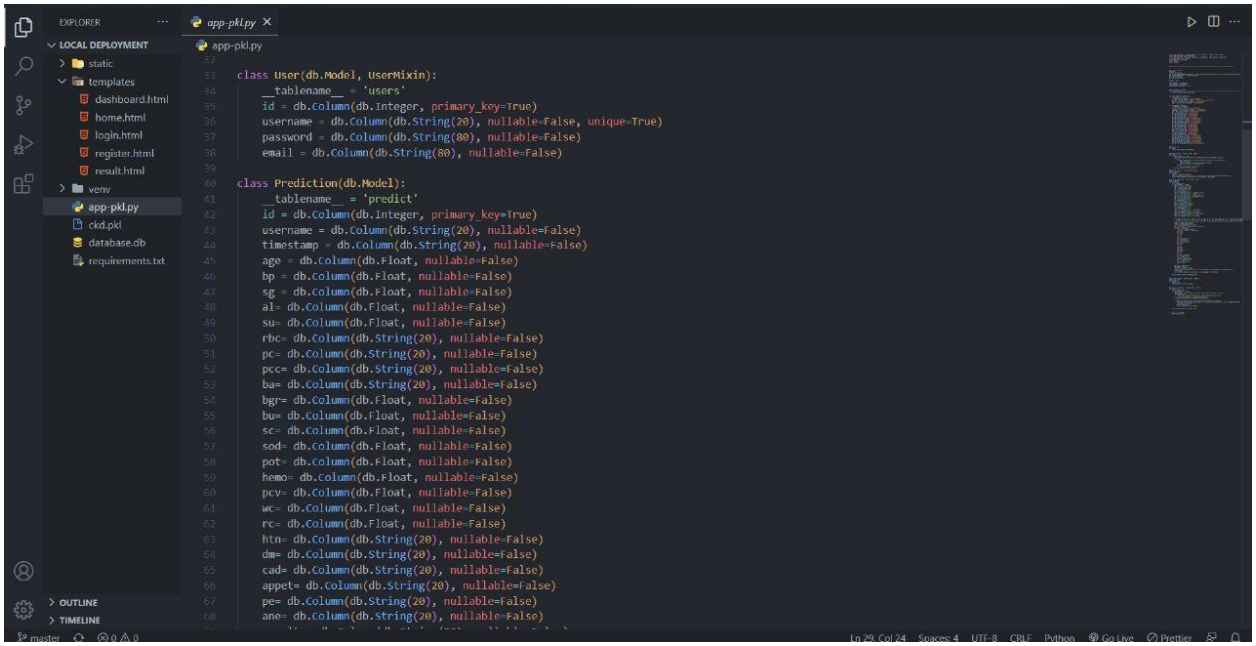
</td>
</tr>
{% endfor -%}
</table>
{% endif %}

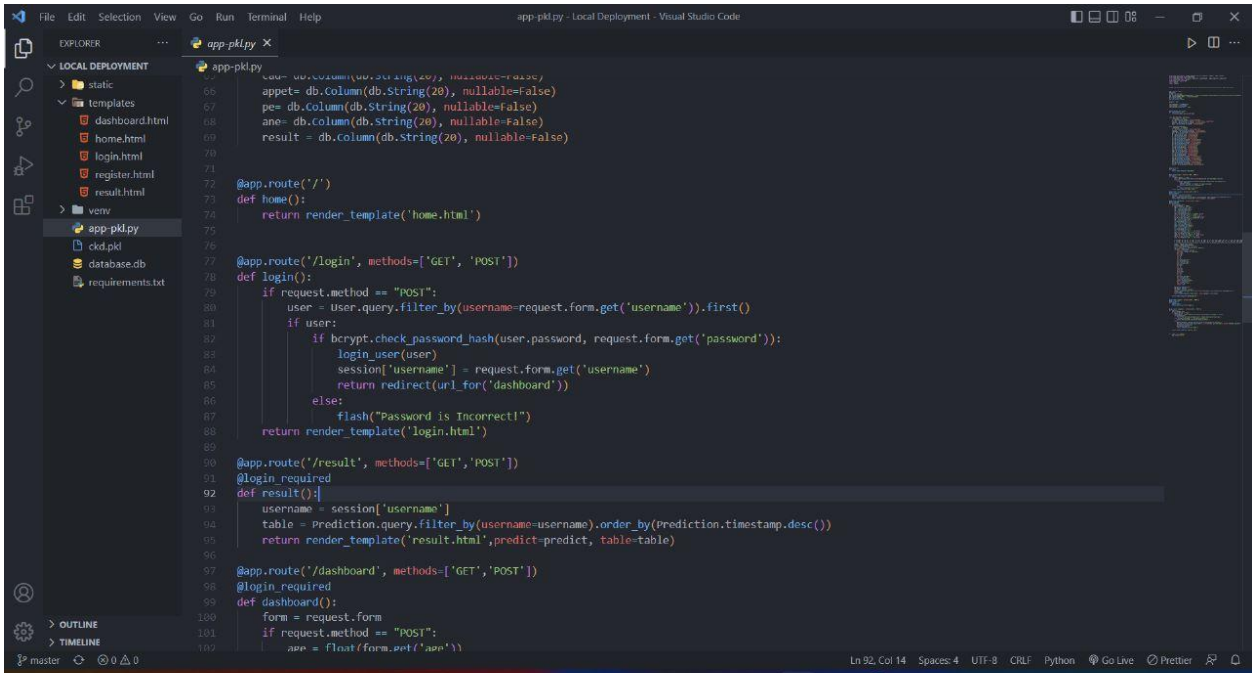
<footer class="footer">
CKD© Predictor with
</footer>
</body>
</html>
OUTLINE
TIMELINE
master
Ln 163, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier

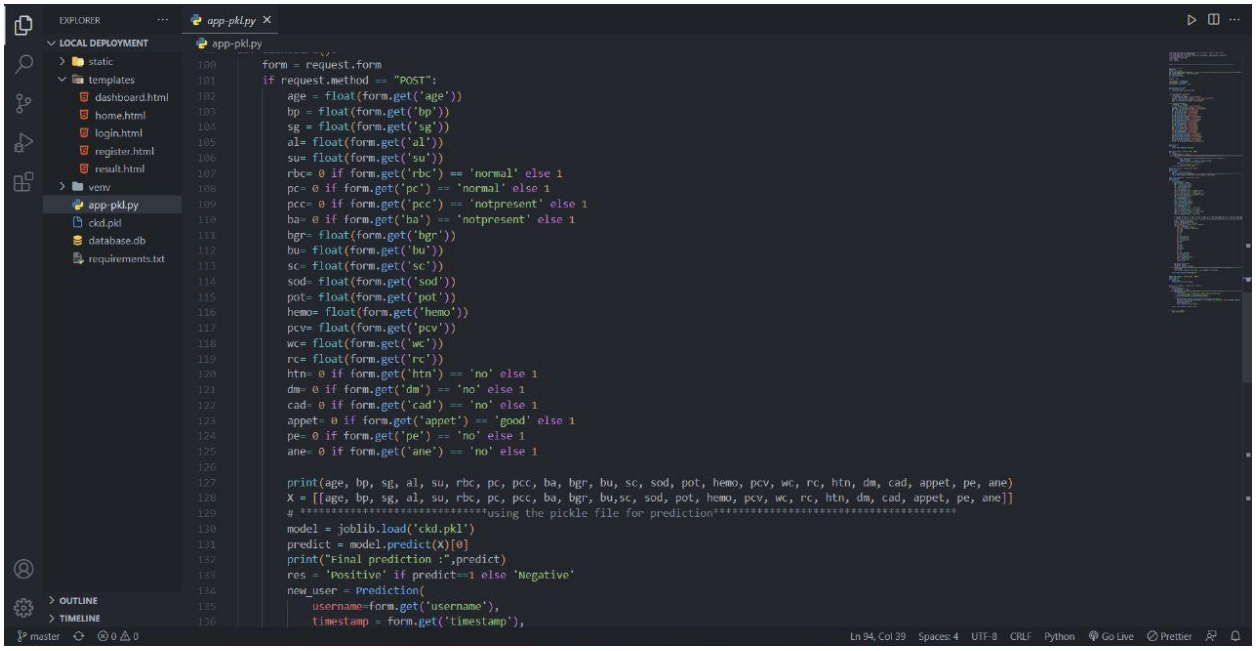


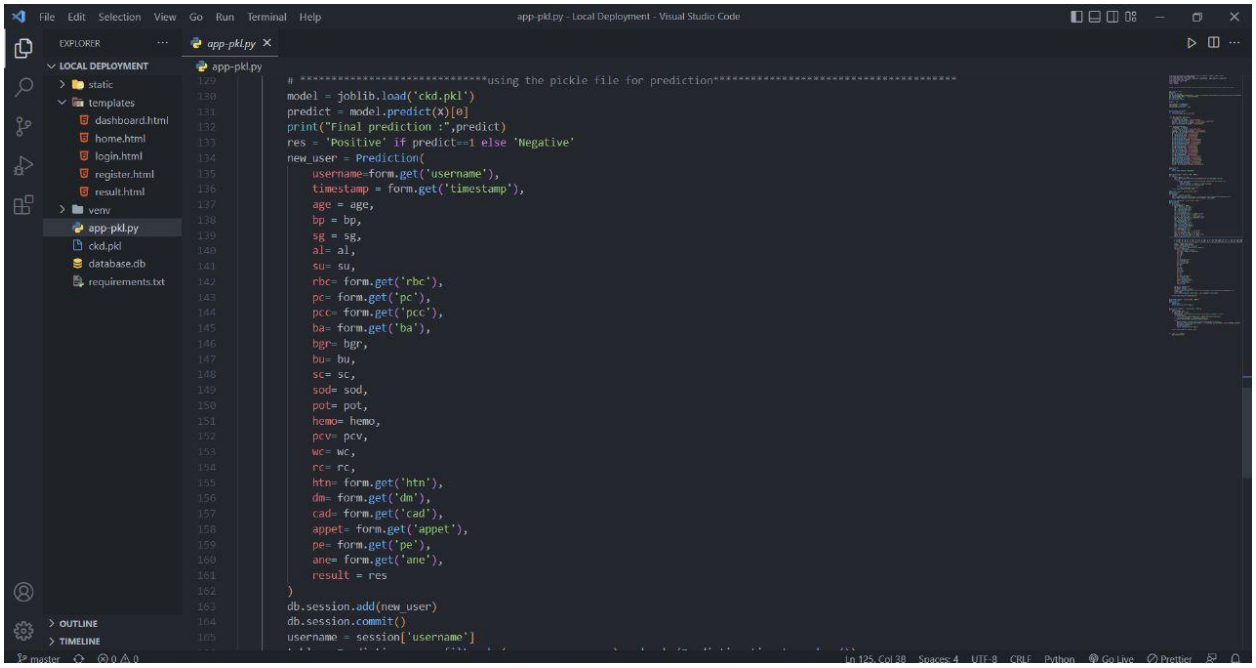
7.3 Flask Integration and Deployment

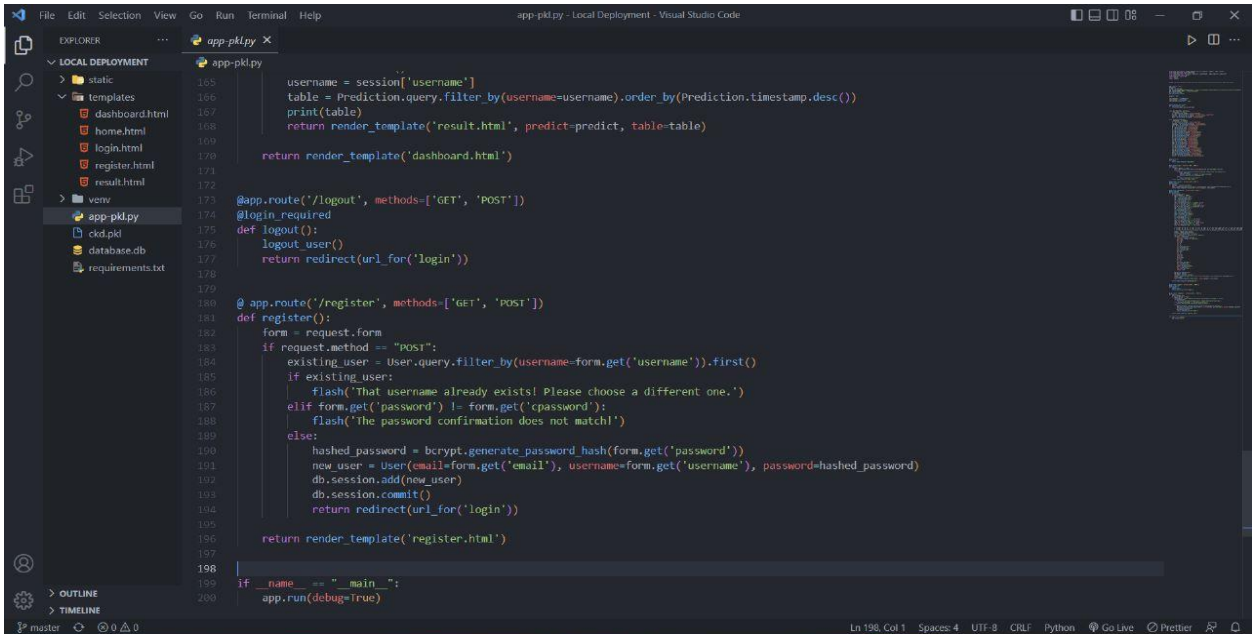



EXPLORER
app-pkl.py
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
class User(db.Model, UserMixin):
__tablename__ = 'users'
id = db.Column(db.Integer, primary_key=True)
username = db.Column(db.String(20), nullable=False, unique=True)
password = db.Column(db.String(80), nullable=False)
email = db.Column(db.String(80), nullable=False)
class Prediction(db.Model):
__tablename__ = 'predict'
id = db.Column(db.Integer, primary_key=True)
username = db.Column(db.String(20), nullable=False)
timestamp = db.Column(db.String(20), nullable=False)
age = db.Column(db.Float, nullable=False)
bp = db.Column(db.Float, nullable=False)
sg = db.Column(db.Float, nullable=False)
al= db.Column(db.Float, nullable=False)
su= db.Column(db.Float, nullable=False)
rbc= db.Column(db.String(20), nullable=False)
pc= db.Column(db.String(20), nullable=False)
pcc= db.Column(db.String(20), nullable=False)
ba= db.Column(db.String(20), nullable=False)
bgr= db.Column(db.Float, nullable=False)
bu= db.Column(db.Float, nullable=False)
sc= db.Column(db.Float, nullable=False)
sod= db.Column(db.Float, nullable=False)
pot= db.Column(db.Float, nullable=False)
hemo= db.Column(db.Float, nullable=False)
pcv= db.Column(db.Float, nullable=False)
wc= db.Column(db.Float, nullable=False)
rc= db.Column(db.Float, nullable=False)
htn= db.Column(db.String(20), nullable=False)
dm= db.Column(db.String(20), nullable=False)
cad= db.Column(db.String(20), nullable=False)
appet= db.Column(db.String(20), nullable=False)
pe= db.Column(db.String(20), nullable=False)
ane= db.Column(db.String(20), nullable=False)
master
Ln 29, Col 24
Spaces: 4
UTF-8
CRLF
Python
Go Live
Prettier


File Edit Selection View Go Run Terminal Help
app-pkl.py - Local Deployment - Visual Studio Code
EXPLORER
app-pkl.py
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
appet= db.Column(db.String(20), nullable=False)
pe= db.Column(db.String(20), nullable=False)
ane= db.Column(db.String(20), nullable=False)
result = db.Column(db.String(20), nullable=False)
@app.route('/')
def home():
return render_template('home.html')
@app.route('/login', methods=['GET', 'POST'])
def login():
if request.method == "POST":
user = User.query.filter_by(username=request.form.get('username')).first()
if user:
if bcrypt.check_password_hash(user.password, request.form.get('password')):
login_user(user)
session['username'] = request.form.get('username')
return redirect(url_for('dashboard'))
else:
flash("Password is Incorrect!")
return render_template('login.html')
@app.route('/result', methods=['GET','POST'])
@login_required
def result():
username = session['username']
table = Prediction.query.filter_by(username=username).order_by(Prediction.timestamp.desc())
return render_template('result.html',predict=predict, table=table)
@app.route('/dashboard', methods=['GET','POST'])
@login_required
def dashboard():
form = request.form
if request.method == "POST":
master
Ln 92, Col 14
Spaces: 4
UTF-8
CRLF
Python
Go Live
Prettier

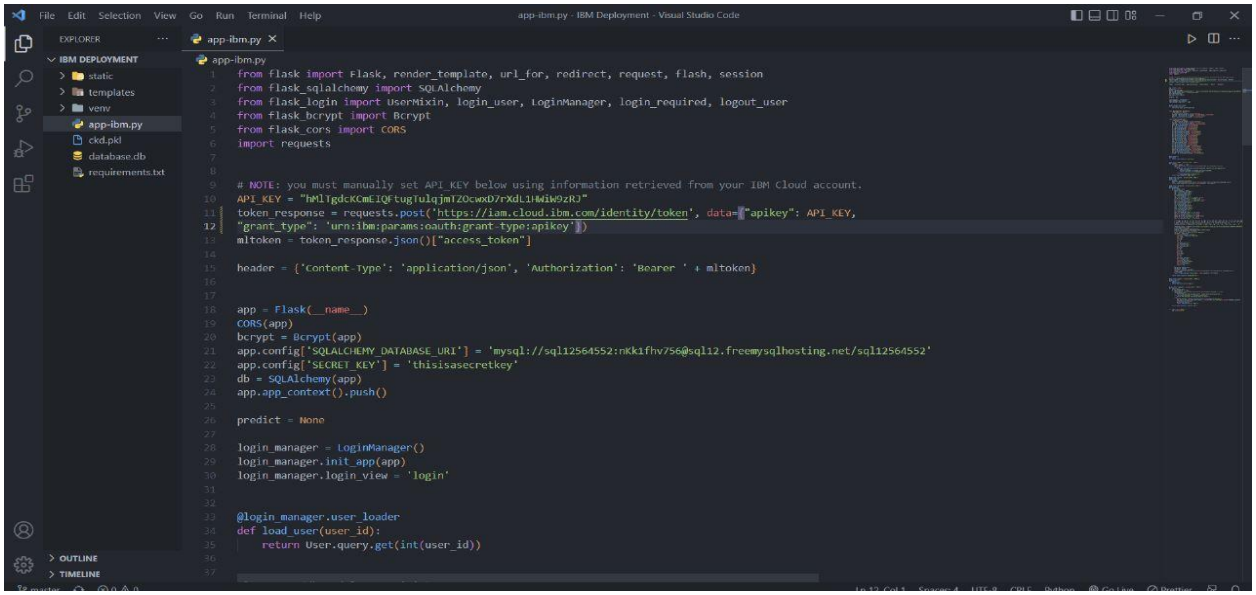

EXPLORER
app-pkl.py
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
form = request.form
if request.method == "POST":
age = float(form.get('age'))
bp = float(form.get('bp'))
sg = float(form.get('sg'))
al= float(form.get('al'))
su= float(form.get('su'))
rbc= 0 if form.get('rbc') == 'normal' else 1
pc= 0 if form.get('pc') == 'normal' else 1
pcc= 0 if form.get('pcc') == 'notpresent' else 1
ba= 0 if form.get('ba') == 'notpresent' else 1
bgr= float(form.get('bgr'))
bu= float(form.get('bu'))
sc= float(form.get('sc'))
sod= float(form.get('sod'))
pot= float(form.get('pot'))
hemo= float(form.get('hemo'))
pcv= float(form.get('pcv'))
wc= float(form.get('wc'))
rc= float(form.get('rc'))
htn= 0 if form.get('htn') == 'no' else 1
dm= 0 if form.get('dm') == 'no' else 1
cad= 0 if form.get('cad') == 'no' else 1
appet= 0 if form.get('appet') == 'good' else 1
pe= 0 if form.get('pe') == 'no' else 1
ane= 0 if form.get('ane') == 'no' else 1
print(age, bp, sg, al, su, rbc, pc, pcc, ba, bgr, bu, sc, sod, pot, hemo, pcv, wc, rc, htn, dm, cad, appet, pe, ane)
X = [[age, bp, sg, al, su, rbc, pc, pcc, ba, bgr, bu,sc, sod, pot, hemo, pcv, wc, rc, htn, dm, cad, appet, pe, ane]]
# ******************************using the pickle file for prediction**************************************
model = joblib.load('ckd.pkl')
predict = model.predict(X)[0]
print("Final prediction :",predict)
res = 'Positive' if predict==1 else 'Negative'
new_user = Prediction(
username=form.get('username'),
timestamp = form.get('timestamp'),
master
Ln 94, Col 39
Spaces: 4
UTF-8
CRLF
Python
Go Live
Prettier

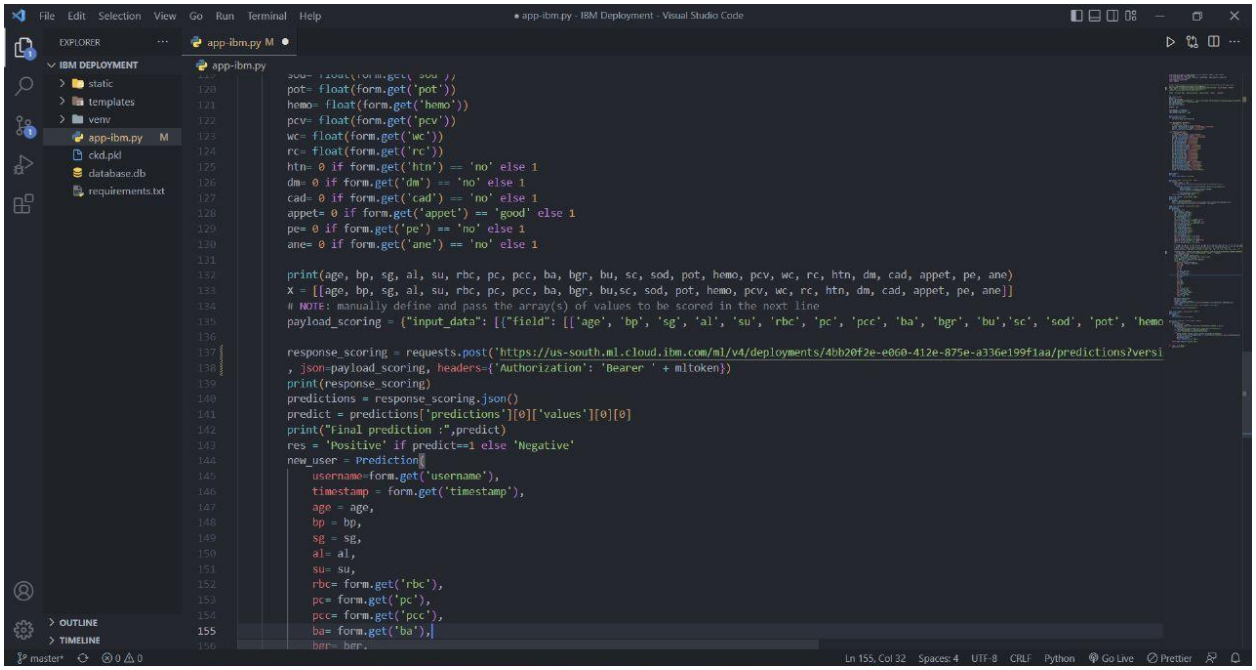

File Edit Selection View Go Run Terminal Help
app-pkl.py - Local Deployment - Visual Studio Code
EXPLORER
app-pkl.py
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
# ******************************using the pickle file for prediction**************************************
model = joblib.load('ckd.pkl')
predict = model.predict(X)[0]
print("Final prediction :",predict)
res = 'Positive' if predict==1 else 'Negative'
new_user = Prediction(
username=form.get('username'),
timestamp = form.get('timestamp'),
age = age,
bp = bp,
sg = sg,
al= al,
su= su,
rbc= form.get('rbc'),
pc= form.get('pc'),
pcc= form.get('pcc'),
ba= form.get('ba'),
bgr= bgr,
bu= bu,
sc= sc,
sod= sod,
pot= pot,
hemo= hemo,
pcv= pcv,
wc= wc,
rc= rc,
htn= form.get('htn'),
dm= form.get('dm'),
cad= form.get('cad'),
appet= form.get('appet'),
pe= form.get('pe'),
ane= form.get('ane'),
result = res
)
db.session.add(new_user)
db.session.commit()
username = session['username']
master
Ln 125, Col 28
Spaces: 4
UTF-8
CRLF
Python
Go Live
Prettier

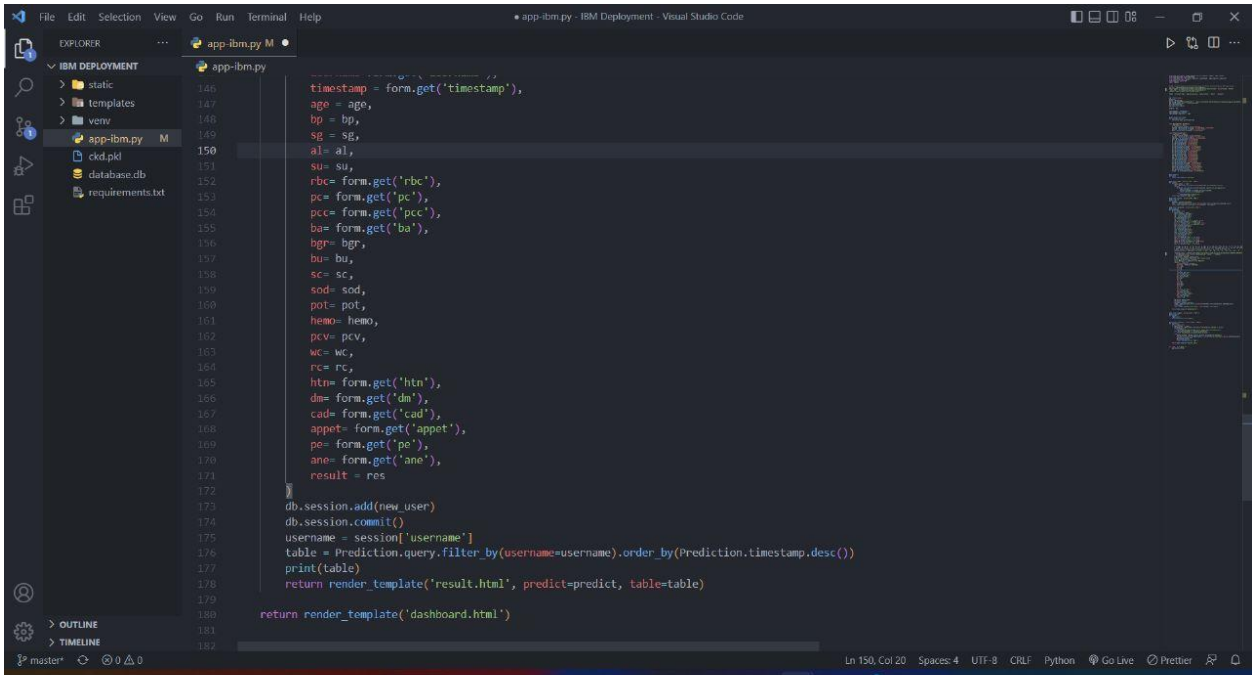


Using pickle to integrate with flask

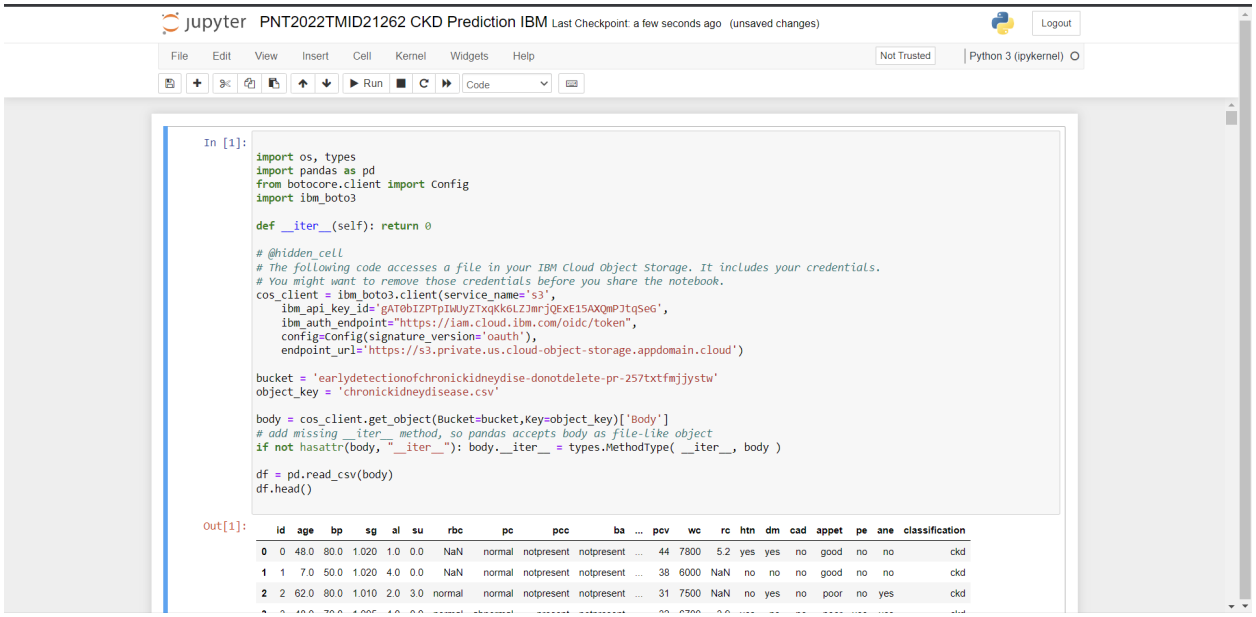
Flask changes for ibm deployment

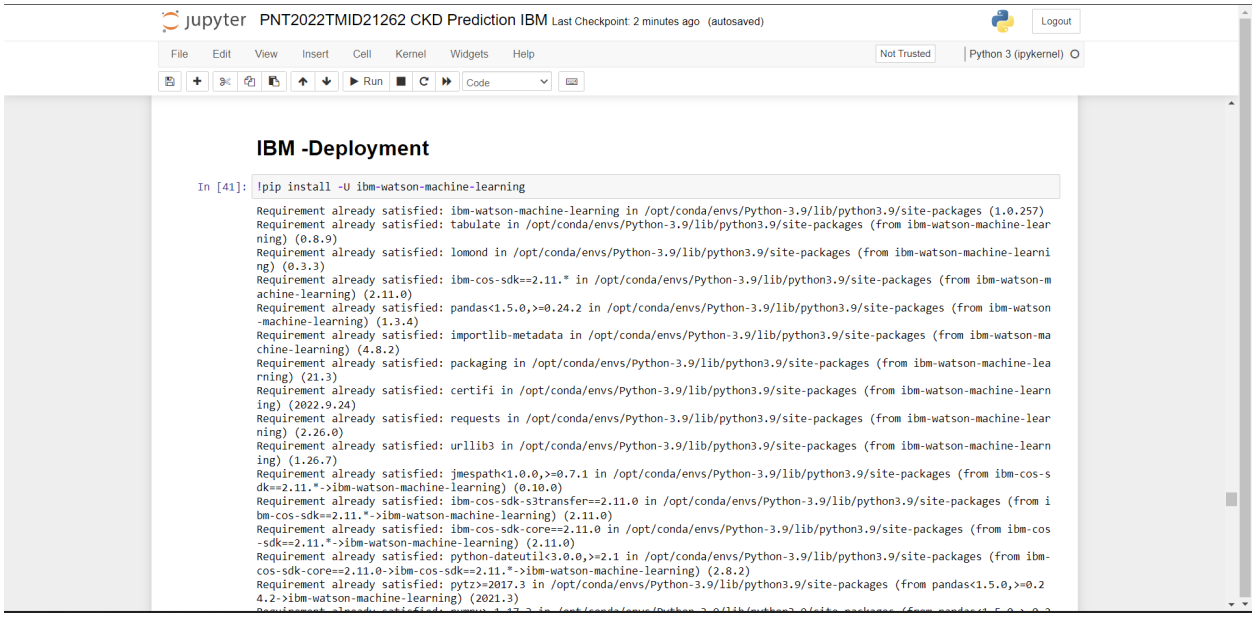


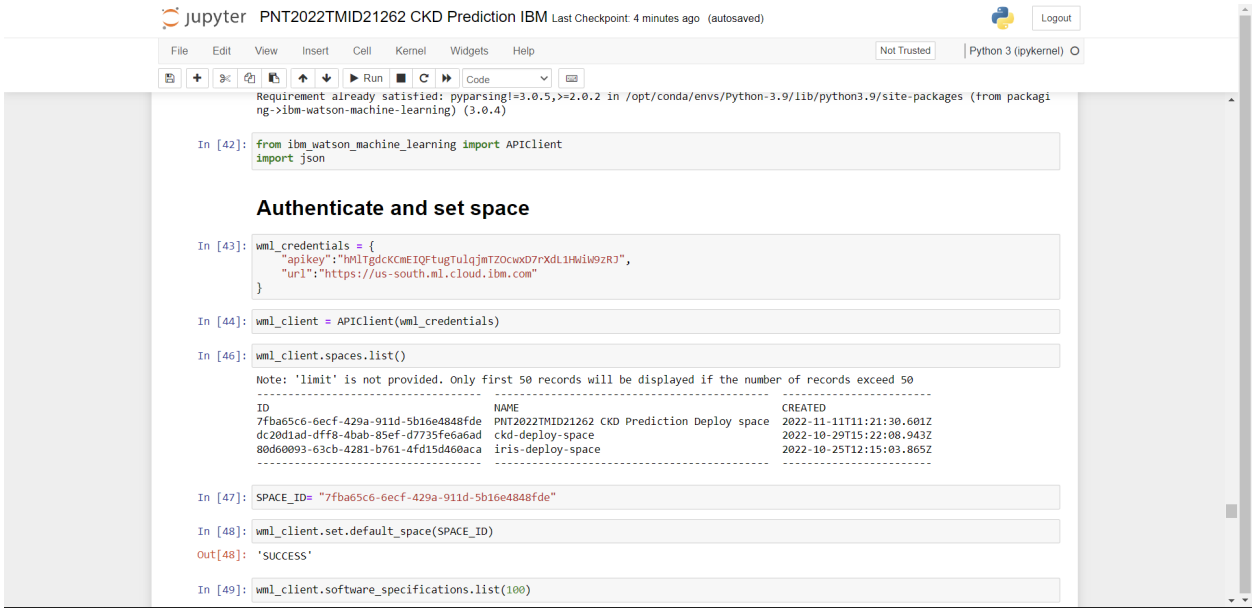

File Edit Selection View Go Run Terminal Help
• app-ibm.py - IBM Deployment - Visual Studio Code
EXPLORER
IBM DEPLOYMENT
static
templates
venv
app-ibm.py M
ckd.pkl
database.db
requirements.txt
app-ibm.py M
app-ibm.py
pot= float(form.get('pot'))
hemo= float(form.get('hemo'))
pcv= float(form.get('pcv'))
wc= float(form.get('wc'))
rc= float(form.get('rc'))
htn= 0 if form.get('htn') == 'no' else 1
dm= 0 if form.get('dm') == 'no' else 1
cad= 0 if form.get('cad') == 'no' else 1
appet= 0 if form.get('appet') == 'good' else 1
pe= 0 if form.get('pe') == 'no' else 1
ane= 0 if form.get('ane') == 'no' else 1
print(age, bp, sg, al, su, rbc, pc, pcc, ba, bgr, bu, sc, sod, pot, hemo, pcv, wc, rc, htn, dm, cad, appet, pe, ane)
x = [[age, bp, sg, al, su, rbc, pc, pcc, ba, bgr, bu,sc, sod, pot, hemo, pcv, wc, rc, htn, dm, cad, appet, pe, ane]]
# NOTE: manually define and pass the array(s) of values to be scored in the next line
payload_scoring = {"input_data": [{"field": [['age', 'bp', 'sg', 'al', 'su', 'rbc', 'pc', 'pcc', 'ba', 'bgr', 'bu','sc', 'sod', 'pot', 'hemo
response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/4bb20f2e-e060-412e-875e-a336e199f1aa/predictions?versi
, json=payload_scoring, headers={'Authorization': 'Bearer ' + mltoken})
print(response_scoring)
predictions = response_scoring.json()
predict = predictions['predictions'][0]['values'][0][0]
print("Final prediction :",predict)
res = 'Positive' if predict==1 else 'Negative'
new_user = Prediction(
username=form.get('username'),
timestamp = form.get('timestamp'),
age = age,
bp = bp,
sg = sg,
al= al,
su= su,
rbc= form.get('rbc'),
pc= form.get('pc'),
pcc= form.get('pcc'),
ba= form.get('ba'),
OUTLINE
TIMELINE
master*
Ln 155, Col 32 Spaces: 4 UTF-8 CRLF Python Go Live Prettier

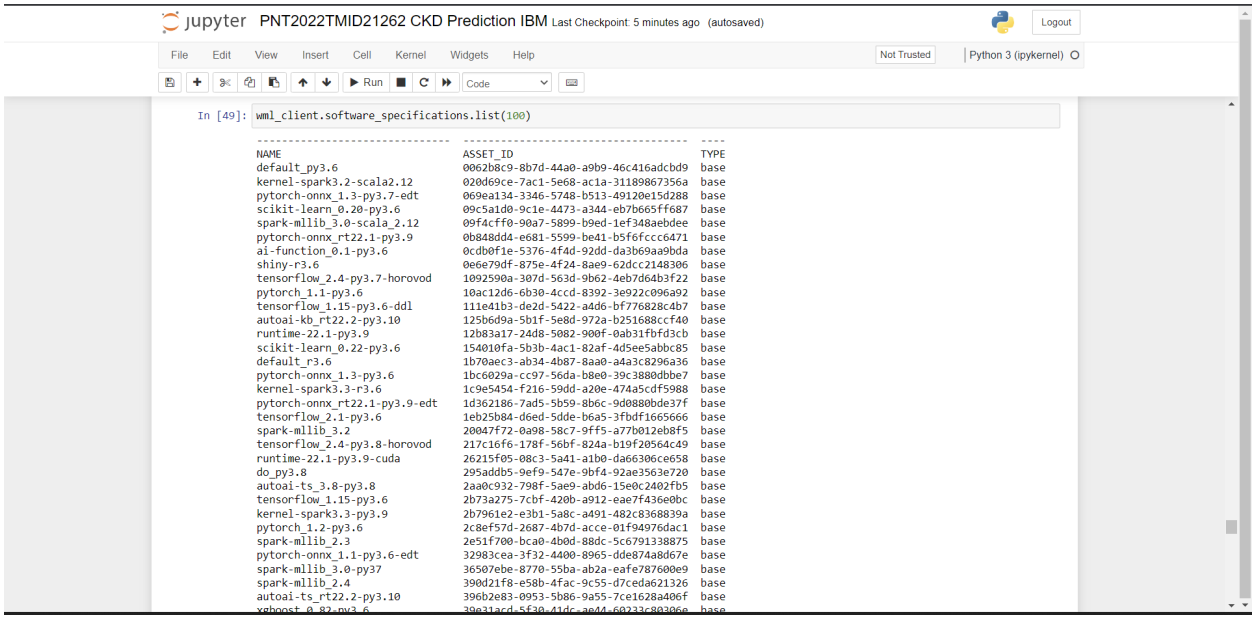

File Edit Selection View Go Run Terminal Help
• app-ibm.py - IBM Deployment - Visual Studio Code
EXPLORER
IBM DEPLOYMENT
static
templates
venv
app-ibm.py M
ckd.pkl
database.db
requirements.txt
app-ibm.py M
app-ibm.py
timestamp = form.get('timestamp'),
age = age,
bp = bp,
sg = sg,
al= al,
su= su,
rbc= form.get('rbc'),
pc= form.get('pc'),
pcc= form.get('pcc'),
ba= form.get('ba'),
bgr= bgr,
bu= bu,
sc= sc,
sod= sod,
pot= pot,
hemo= hemo,
pcv= pcv,
wc= wc,
rc= rc,
htn= form.get('htn'),
dm= form.get('dm'),
cad= form.get('cad'),
appet= form.get('appet'),
pe= form.get('pe'),
ane= form.get('ane'),
result = res
)
db.session.add(new_user)
db.session.commit()
username = session['username']
table = Prediction.query.filter_by(username=username).order_by(Prediction.timestamp.desc())
print(table)
return render_template('result.html', predict=predict, table=table)
return render_template('dashboard.html')
OUTLINE
TIMELINE
master*
Ln 150, Col 20 Spaces: 4 UTF-8 CRLF Python Go Live Prettier

Ibm deployed ipynb file






jupyter PNT2022TMID21262 CKD Prediction IBM Last Checkpoint: 4 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm-watson-machine-learning) (3.0.4)
In [42]:
from ibm_watson_machine_learning import APIClient
import json
Authenticate and set space
In [43]:
wml_credentials = {
"apikey":"hMlTgdcKCmEIQFtugTulqjmTZOcwxD7rXdL1HWiW9zRJ",
"url":"https://us-south.ml.cloud.ibm.com"
}
In [44]:
wml_client = APIClient(wml_credentials)
In [46]:
wml_client.spaces.list()
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
ID NAME CREATED
7fba65c6-6ecf-429a-911d-5b16e4848fde PNT2022TMID21262 CKD Prediction Deploy space 2022-11-11T11:21:30.601Z
dc20d1ad-dff8-4bab-85ef-d7735fe6a6ad ckd-deploy-space 2022-10-29T15:22:08.943Z
80d60093-63cb-4281-b761-4fd15d460aca iris-deploy-space 2022-10-25T12:15:03.865Z
In [47]:
SPACE_ID= "7fba65c6-6ecf-429a-911d-5b16e4848fde"
In [48]:
wml_client.set.default_space(SPACE_ID)
Out[48]:
'SUCCESS'
In [49]:
wml_client.software_specifications.list(100)


jupyter PNT2022TMID21262 CKD Prediction IBM Last Checkpoint: 5 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [49]:
wml_client.software_specifications.list(100)
NAME ASSET_ID TYPE
default_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base
kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base
pytorch-onnx_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base
scikit-learn_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base
spark-mllib_3.0-scala_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base
pytorch-onnx_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base
ai-function_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base
shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base
tensorflow_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base
pytorch_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base
tensorflow_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base
autoai-kb_rt22.2-py3.10 125b6d9a-5b1f-5e8d-972a-b251688ccf40 base
runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base
scikit-learn_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base
default_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base
pytorch-onnx_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base
kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base
pytorch-onnx_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base
tensorflow_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base
spark-mllib_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base
tensorflow_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base
runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base
do_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base
autoai-ts_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base
tensorflow_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base
kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base
pytorch_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base
spark-mllib_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base
pytorch-onnx_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base
spark-mllib_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base
spark-mllib_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base
autoai-ts_rt22.2-py3.10 396b2e83-0953-5b86-9a55-7ce1628a406f base

jupyter PNT2022TMID21262 CKD Prediction IBM Last Checkpoint: 6 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel)

Run Code

Saving and Deploying the Model

In [50]:
```
import sklearn
sklearn.__version__
```

Out[50]: '1.0.2'

In [51]:
```
# Set Python Version
MODEL_NAME = 'PNT2022TMID21262 CKD Prediction'
DEPLOYMENT_NAME = 'PNT2022TMID21262 CKD Prediction'
DEMO_MODEL = model
```

In [53]:
```
# Set Python Version
software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

In [54]:
```
# Setup model meta
model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
```

In [56]:
```
#Save model

model_details = wml_client.repository.store_model(
    model=DEMO_MODEL,
    meta_props=model_props,
    training_data=X_train,
    training_target=y_train
)
```

In [57]:
```
model_details
```

jupyter PNT2022TMID21262 CKD Prediction IBM Last Checkpoint: 7 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

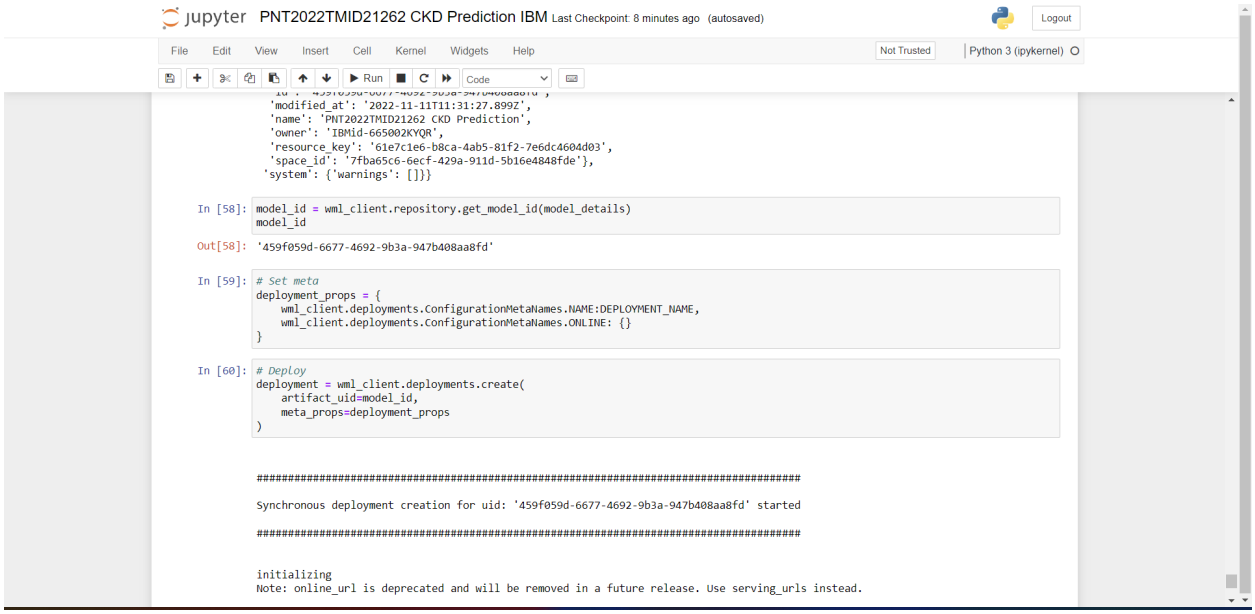
Not Trusted | Python 3 (ipykernel)

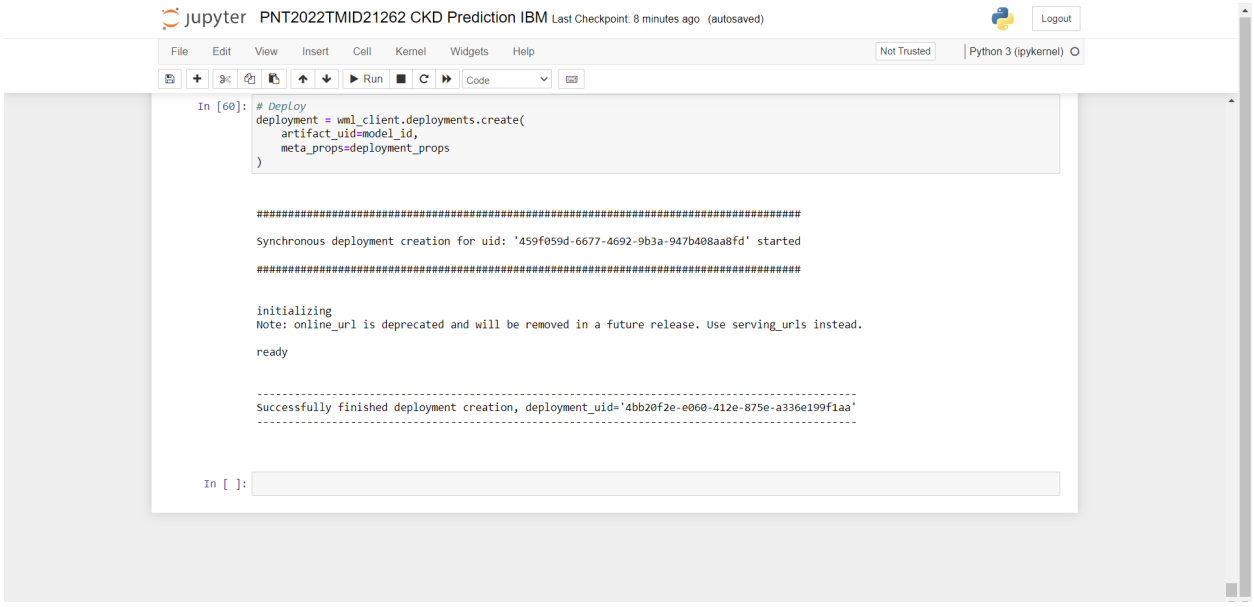
Run Code

In [57]:
```
model_details
```

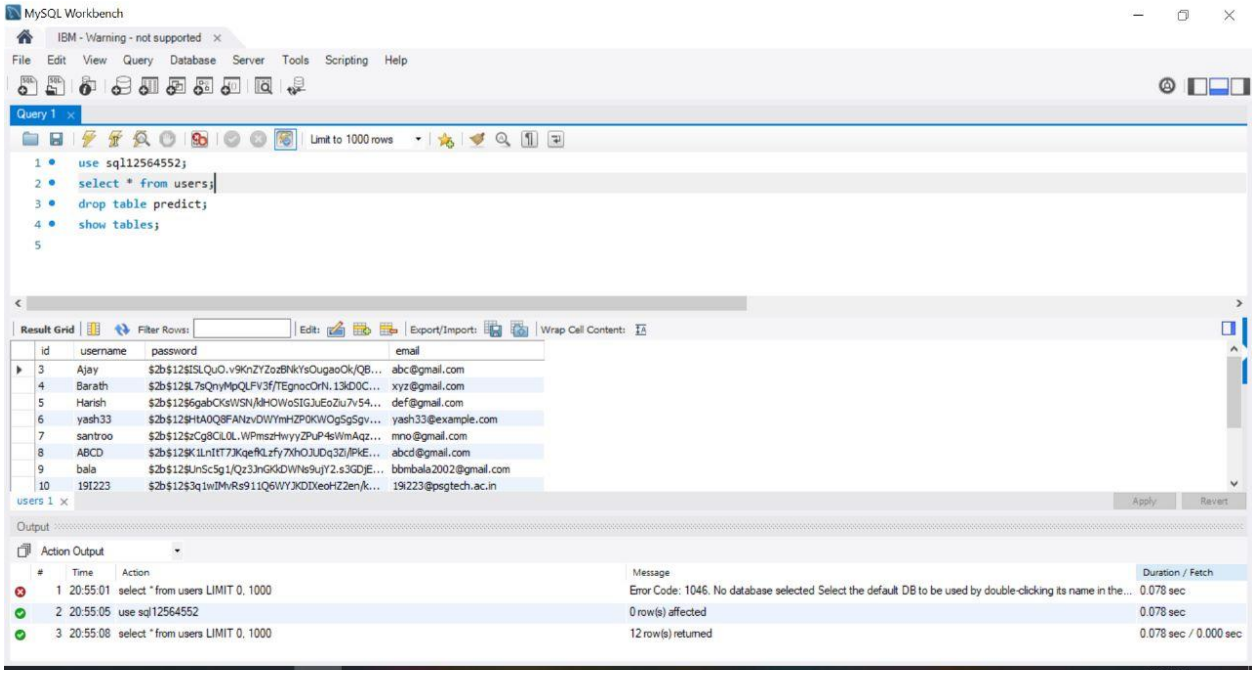
Out[57]:

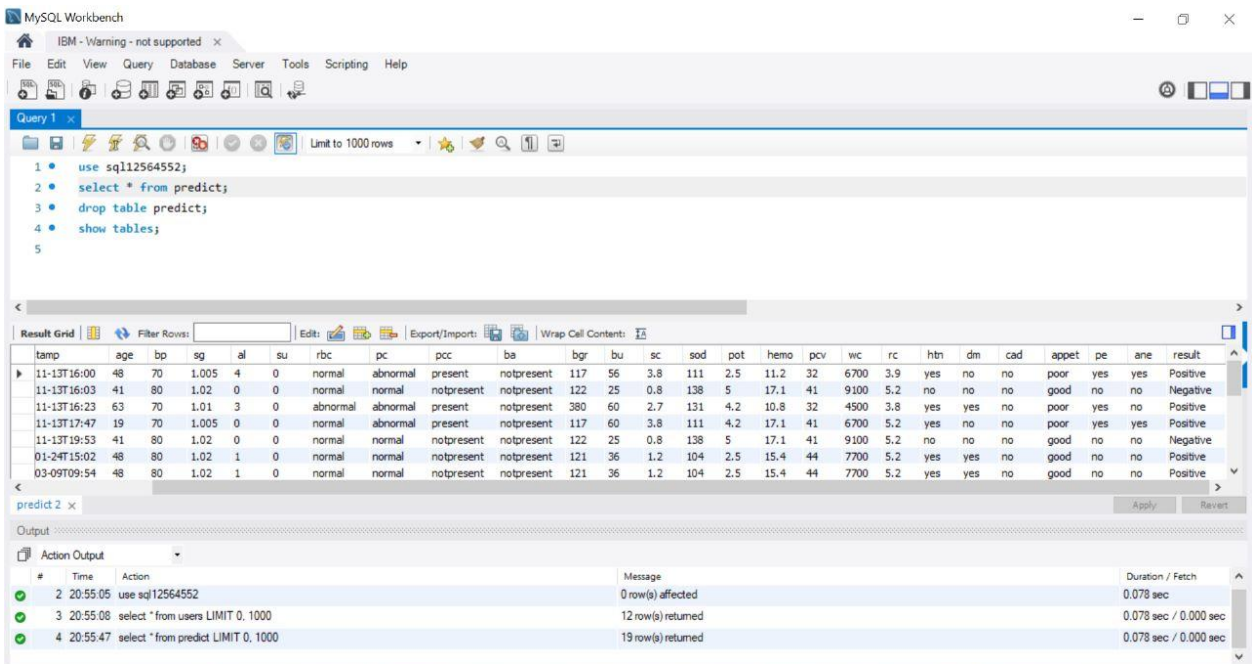
```
{'entity': {'hybrid_pipeline_software_specs': [],
  'label_column': 'classification',
  'schemas': {'input': [{'fields': [{'name': 'age', 'type': 'float64'},
     {'name': 'bp', 'type': 'float64'},
     {'name': 'sg', 'type': 'float64'},
     {'name': 'al', 'type': 'float64'},
     {'name': 'su', 'type': 'float64'},
     {'name': 'rbc', 'type': 'int64'},
     {'name': 'pc', 'type': 'int64'},
     {'name': 'pcc', 'type': 'int64'},
     {'name': 'ba', 'type': 'int64'},
     {'name': 'bgr', 'type': 'float64'},
     {'name': 'bu', 'type': 'float64'},
     {'name': 'sc', 'type': 'float64'},
     {'name': 'sod', 'type': 'float64'},
     {'name': 'pot', 'type': 'float64'},
     {'name': 'hemo', 'type': 'float64'},
     {'name': 'pcv', 'type': 'float64'},
     {'name': 'wc', 'type': 'float64'},
     {'name': 'rc', 'type': 'float64'},
     {'name': 'htn', 'type': 'int64'},
     {'name': 'dm', 'type': 'int64'},
     {'name': 'cad', 'type': 'int64'},
     {'name': 'appet', 'type': 'int64'},
     {'name': 'pe', 'type': 'int64'},
     {'name': 'ane', 'type': 'int64'}],
    'id': '1',
    'type': 'struct'}],
   'output': []},
  'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
   'name': 'runtime-22.1-py3.9'},
  'type': 'scikit-learn_1.0'},
 'metadata': {'created_at': '2022-11-11T11:31:24.426Z',
  'id': '459f059d-6677-4692-9b3a-947b408aa8fd',
  'modified_at': '2022-11-11T11:31:27.899Z',
```


jupyter PNT2022TMID21262 CKD Prediction IBM Last Checkpoint: 8 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
'modified_at': '2022-11-11T11:31:27.899Z',
'name': 'PNT2022TMID21262 CKD Prediction',
'owner': 'IBMid-665002KYQR',
'resource_key': '61e7c1e6-b8ca-4ab5-81f2-7e6dc4604d03',
'space_id': '7fba65c6-6ecf-429a-911d-5b16e4848fde'},
'system': {'warnings': []}}
In [58]:
model_id = wml_client.repository.get_model_id(model_details)
model_id
Out[58]:
'459f059d-6677-4692-9b3a-947b408aa8fd'
In [59]:
# Set meta
deployment_props = {
wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
}
In [60]:
# Deploy
deployment = wml_client.deployments.create(
artifact_uid=model_id,
meta_props=deployment_props
)
Synchronous deployment creation for uid: '459f059d-6677-4692-9b3a-947b408aa8fd' started
initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.


jupyter PNT2022TMID21262 CKD Prediction IBM Last Checkpoint: 8 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [60]:
# Deploy
deployment = wml_client.deployments.create(
artifact_uid=model_id,
meta_props=deployment_props
)
Synchronous deployment creation for uid: '459f059d-6677-4692-9b3a-947b408aa8fd' started
initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.
ready
Successfully finished deployment creation, deployment_uid='4bb20f2e-e060-412e-875e-a336e199f1aa'
In [ ]:

7.4 Database (mysql)



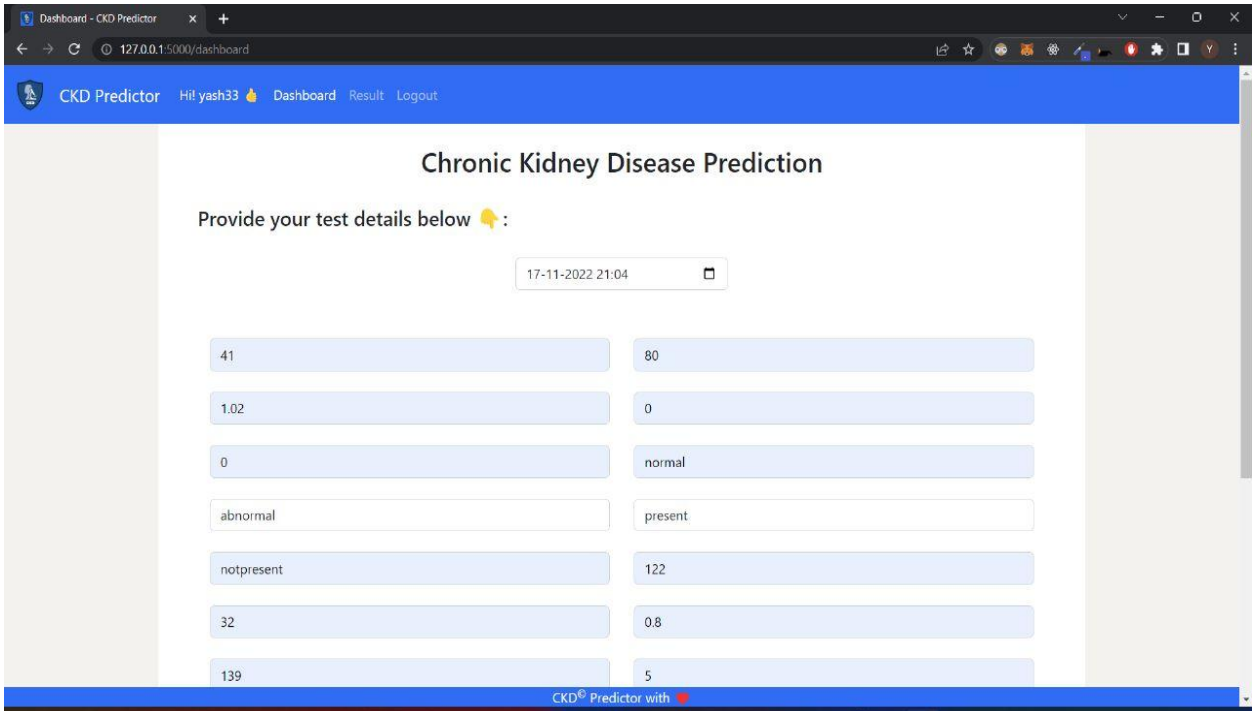


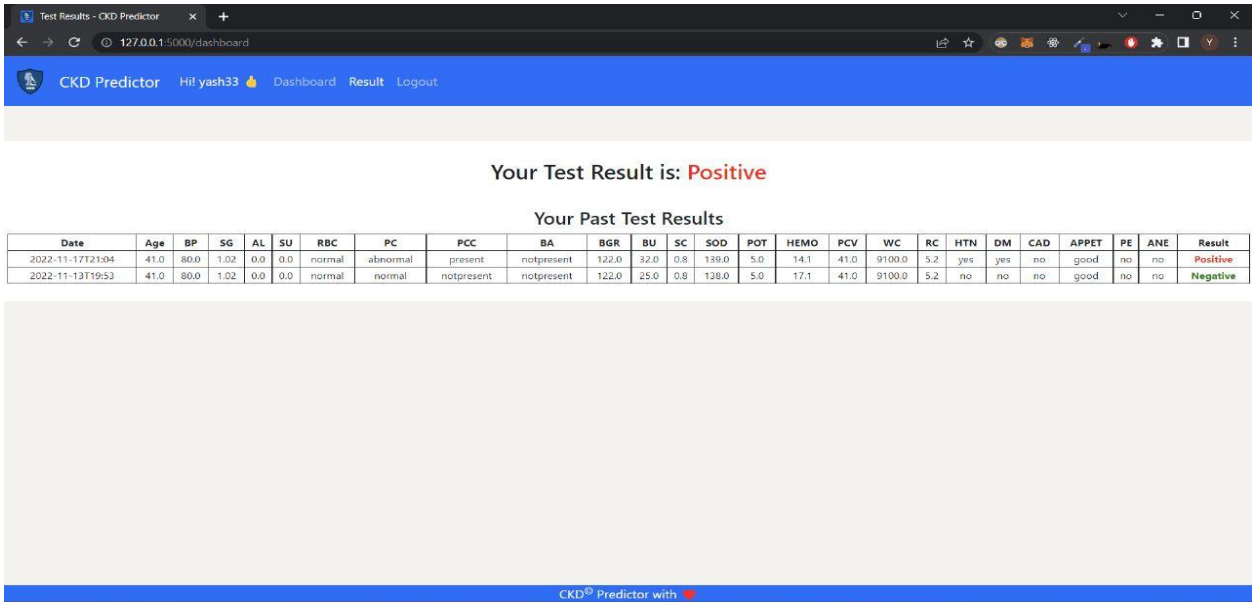
8. TESTING

8.1 Test cases

Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)
Kaggle	1.Enter into kaggle website 2.Download the dataset	https://www.kaggle.com/	Download the Dataset	Working as expected	Pass		NO
Anaconda prompt , Jupyter Notebook	1.Enter Anaconda prompt 2.Enter Jupyter Notebook & do Data pre-processing		Pre-processing the dataset using machine learning Algorithm	Working as expected	pass		NO
Anaconda prompt , Jupyter Notebook	1.Enter Anaconda prompt 2.Enter Jupyter Notebook & do Model Building	Model building using logistic regression	Build a Machine Learning Model	Working as expected	pass		NO
Visual Studio Code	1.Click on VS code ,create html pages . Run html pages on app.py by using live server .	Run a website in localhost server http://127.0.0.1:5000/	Appears a Prediction page on local host server	Working as expected	pass		NO
Visual Studio Code	Click on the http link Enter the values as in the dataset Click on submit	Gives prediction result as patient have CKD or NOT http://127.0.0.1:5000/predict	Predict the Result	Working as expected	Pass		NO
	1.Enter IBM Cloud using login credentials 2.Use jupyter notebook in IBM	Deploy the project in IBM CLOUD	Application should show same resut as vs code flask integration				

Sample tests:





8.2 User Acceptance Testing (UAT)

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

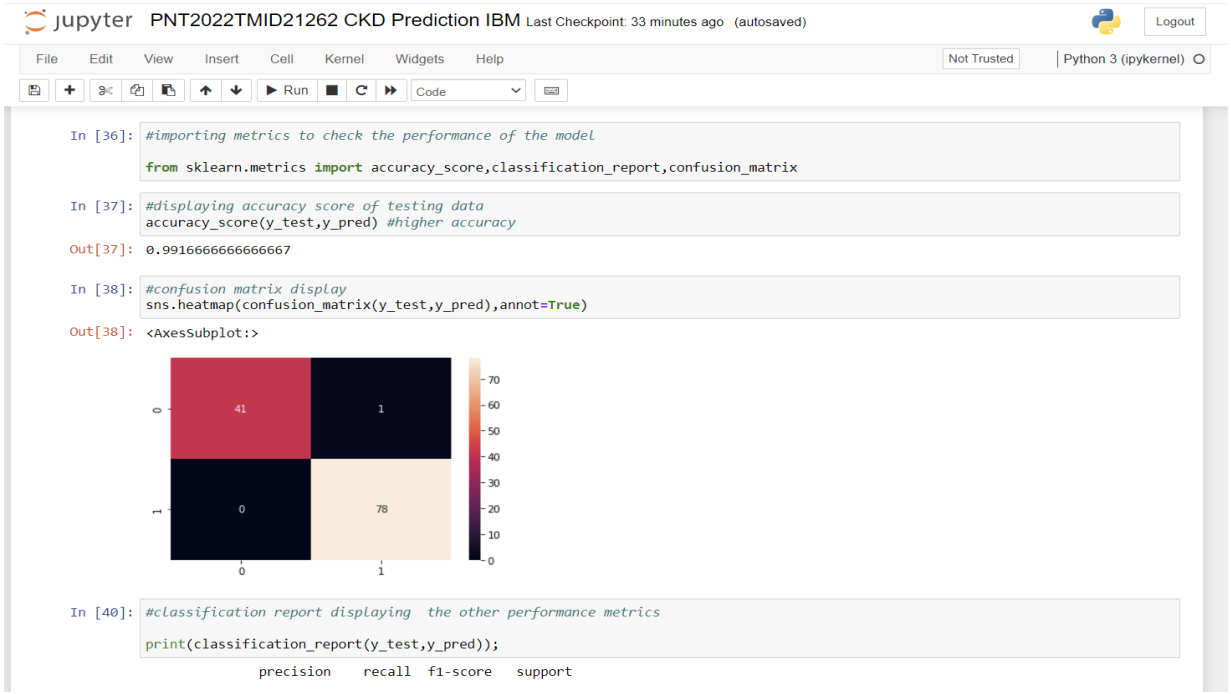
3. Test Case Analysis

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 (Random Forest Classifier)



```
In [40]: #classification report displaying  the other performance metrics

         print(classification_report(y_test,y_pred));
```

```
              precision    recall  f1-score   support

           0       1.00      0.98      0.99        42
           1       0.99      1.00      0.99        78

    accuracy                           0.99       120
   macro avg       0.99      0.99      0.99       120
weighted avg       0.99      0.99      0.99       120
```

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

11.CONCLUSION

Chronic Kidney Disease as the name suggests it's a chronic disease,any chronic disease would make the person miserable and last longer till their livelihood . If in such cases the disease gets unnoticed in early stages which can be cured by medical facilities it's a huge carelessness and risking a person's life . In such cases finding an optimal solution is important ,thus there comes the use of a machine learning model for early detection and prediction of the chronic kidney disease which can greatly reduce the potential risk of getting the disease and get cured immediately if it is detected in early stages of the disease. Think of the traditional way of diagnosing kidney disease,it is through blood test,and blood test reports take longer than expected ,but blood test is not the only step for diagnosing there are still many more tests taken , which can be time consuming . In those cases the model prediction plays an important role in predicting the disease sooner and faster for the medical team to treat the person if he/she is vulnerable.

Thus early detection of chronic kidney disease is very much necessary in current hospital functioning to diagnose the patient in no time and do necessary treatment to cure if found.

12. FUTURE WORK :

The current work remains the base for the prediction model primarily used by everyone extending from hospitals to normal users . The future aspects can be as follows:

- subscription based model can be created with initial trial basis
- Scaling the existing application for simultaneous user to request
- Modifying the model based on adding new feature in the existing dataset based on the hospitals input and standards

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-26228-1660021841

DEMO LINK :

https://drive.google.com/file/d/1lEgB8TNz4rScn0TqT4uNr3NX53lo1vdQ/view?usp=sharing