Early Detection of Chronic Kidney Disease using Machine Learning

Team ID-PNT2022TMID53897

1.INTRODUCTION

Chronic kidney disease (CKD) is a worldwide under diagnosed public health problem with increasing incidence and prevalence that has high costs and poor outcomes . The number of patients on renal replacement therapy has doubled every decade since 1980, and prevalence of CKD in the early stages has also markedly increased . Unfortunately, CKD in its earliest stages is usually an asymptomatic condition which progresses to its end stage over a period of several years and is diagnosed late in its course. Therefore, strategies to reduce the incidence of end-stage renal disease require effective methods of screening early in the disease process. Intervention in the early stages of CKD seems to be more effective to prevent or delay the progression of CKD. Moreover, reduced kidney function was found to be an independent risk factor for cardiovascular events and/or all-cause mortality . Therefore, early detection of CKD and treatment of its complications seems to be important to improve outcome in cardiovascular diseases. However, the screening methods most suitable to identify for further diagnosis individuals with CKD remain to be settled. Albumin was used as a screening tool but by itself is not sufficient. Garget al found that albumin and renal insufficiency measured on a single occasion identified different segments of the population. More than one third of people with an EGFR below 30 ml/min/1.73 m^2 demonstrated no albumin. Moreover, one third of diabetic patients and almost two thirds of non-diabetic hypertensive patients demonstrated no albumin. There is no simple correlation between the progress of glomerular disease and kidney function . Both glomerular and tubule interstitial damage can mediate impairment of renal function. Furthermore, inulin clearance is better correlated with tubule interstitial damage than glomerular disease . In many forms of renal diseases due to primary glomerular lesions, there is a significant inverse correlation between the extent of tubule interstitial damage and the GFR . Therefore, supplementing screening methods such as albumin and EGFR with other diagnostic tools seems to be necessary for early detection of kidney disease. It is worth mentioning at this point that both albumin and decrease in eGFR were used as 2 basic markers for the new classification of CKD proposed by the Kidney Disease Outcomes Quality

Initiative of the National Kidney Foundation and introduced widely after slight modification by the KDIGO (Kidney Disease: Improving Global Outcomes) international community . Damage to the interstitial compartment, especially to more distal segments of tubules, is accompanied by an inability to maximally concentrate the urine, and therefore may result in nocturia. We believe that asking simple questions about nocturia could help distinguish the missing segment of the population with CKD who demonstrate no albumin (after exclusion of the main reasons for nocturia, such as urinary tract infection, poorly controlled diabetes, severe heart failure and evening dosing of diuretics). The definition of nocturia was accepted as the necessity to urinate more than once during the night, which disturbs sleep and which had lasted at least 3 months.

1.1 Project Overview

Chronic kidney disease (CKD) is one of the most critical health problems due to its increasing prevalence. In this , we aim to test the ability of machine learning algorithms for the prediction of chronic kidney disease using the smallest subset of features. Several statistical tests have been done to remove redundant features such as the ANOVA test, the Pearson's correlation, and the Cramer's V test. Logistic regression, support vector machines, random forest, and gradient boosting algorithms have been trained and tested using 10-fold cross-validation. We achieve an accuracy of 99.1 according to F1- measure from Gradient Boosting classifier and by using Random forest method. Also, we found that hemoglobin has higher importance for both random forest and Gradient boosting in detecting CKD. Finally, our results are among the highest compared to previous studies but with less number of features reached so far. Hence, we can detect CKD at only $26.65 by performing three simple tests.

1.2 Purpose

This project helps everyone to easily detect the Kidney disease which is present in our body . It is very user friendly

2. LITERATURE SURVEY

S.No	PAPER TITLE	TECHNOLOGIES USED	DESCRIPTION
1	Chronic Kidney Disease Prediction and Recommendation of Suitable Diet plan by using Machine Learning	Machine Learning Algorithms ,MDRD equation	The proposed system which detects chronic kidney disease using machine learning defines 3 zones(Safe zone,Caution zone,Danger zone) on the basis of blood potassium level.
2	Performance Analysis of Machine Learning Classifier for Predicting Chronic Kidney Disease	Regression and classification, decision tree classifier, random forest	This proposed system detects CKD- Chronic Kidney Disease using machine learning; they have attained an accuracy of 100% for decision tree classifier, 95.12% for random forest and 98.82% in logistic regression.
3	Prediction of chronic kidney disease (CKD) using Data Science	Support Vector Machine, Random Forest, XGBoost, Logistic Regression, Neural networks, Naive Bayes Classifier.	This research work is primarily concentrated on finding the best suitable classification algorithm which can be used for the diagnosis of CKD based on the classification report and performance factors.

4	Chronic kidney disease Diagnosis using Multilayer perceptron classifier	Multilayer Perceptron Classifier	The Experimental results show that the proposed model can perform classification with the testing accuracy of 92.5% surpassing the scores achieved by SVM and naive bayes classifier.
5	A Neural Network based Model for Predicting Chronic Kidney Diseases	Artificial Neural Network algorithms	The 14 different properties are analyzed and linked to chronic kidney disorder victims and foretold accuracy for a machine learning algorithm named Artificial Neural Network. After analyzing the outcomes, it is recognized that the algorithm gives correctness of 96.
6	Early Diagnosis of Chronic Kidney Disease Using Machine Learning Algorithms with Least Parameters by RFE and Feature Importance Techniques	Linear, Logistic, Decision tree, CART, and Random forest classifier	The primary goal of this research project is to enhance the diagnostic precision by assessing the optimum feature selection and developing a prediction model using machine learning methods. By using different classifier methods, the model achieved a diagnosis accuracy of 0.925.
7	A Machine Learning Methodology for Diagnosing Chronic Kidney Disease	Logistic regression, Random forest, Support vector machine, knearest neighbor, Naive Bayes classifier, and Feed Forward Neural Network	A machine learning approach for diagnosing CKD was suggested in this study. An integrated model that combines logistic regression and random forest with the aid

			of perceptron was utilized and it was able to attain an average accuracy of 99.83% after ten times of simulation.
8	Detection of Chronic Kidney Disease Using Machine Learning Algorithms with Least Number of Predictors	Logistic regression, SVM, Random forest, and Gradient boosting	The link between variables has been researched in order to decrease the number of features and eliminate redundancy. Tenfold cross-validation has been used to train, test, and validate the classifiers.
9	Intelligent systems on the cloud for the early detection of chronic kidney disease	Back-propagation networks, Generalized Feed Forward Neural Networks, and Modular Neural Networks	Utilizing Google Application Engine, the system created in accordance with the best model is uploaded to the Google cloud platform. The end solution can more effectively give CKD
10	Optimization of Prediction Method of Chronic Kidney Disease Using Machine Learning Algorithm.	Support Vector Machine, AdaBoost, Linear Discriminant Analysis, and Gradient Boosting	These algorithms are used using a dataset from the UCI machine learning repository that is available online. Gradient Boosting (GB) Classifiers produce results with a predictably high accuracy of roughly 99.80%. Based on these benchmarks, the most effective and optimized algorithms for the requested job can be chosen

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	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

		Perkusich			

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 modelsare constructed to carry out the diagnosis of CKD	99.16%	2019
3	Preemptive Diagnosis of Chronic Kidney Disease Using	Reem A. Alassaf, Khawla A. Alsulaim, Noura Y. Alroomi,	ANN, SVM, Naïve Bayes along with k-NN comparison approach	ANN,SVM ,Naïve Bayes - 98% k-NN - 93.9%	2018

	Machine Learning Techniques	Nouf S. Alsharif, Mishael F. Aljubeir, Sunday O. Olatunji, Alaa Y. Alahmadi, Mohammed Imran, Rahma A. Alzahrani, Nora S. Alturayeif			

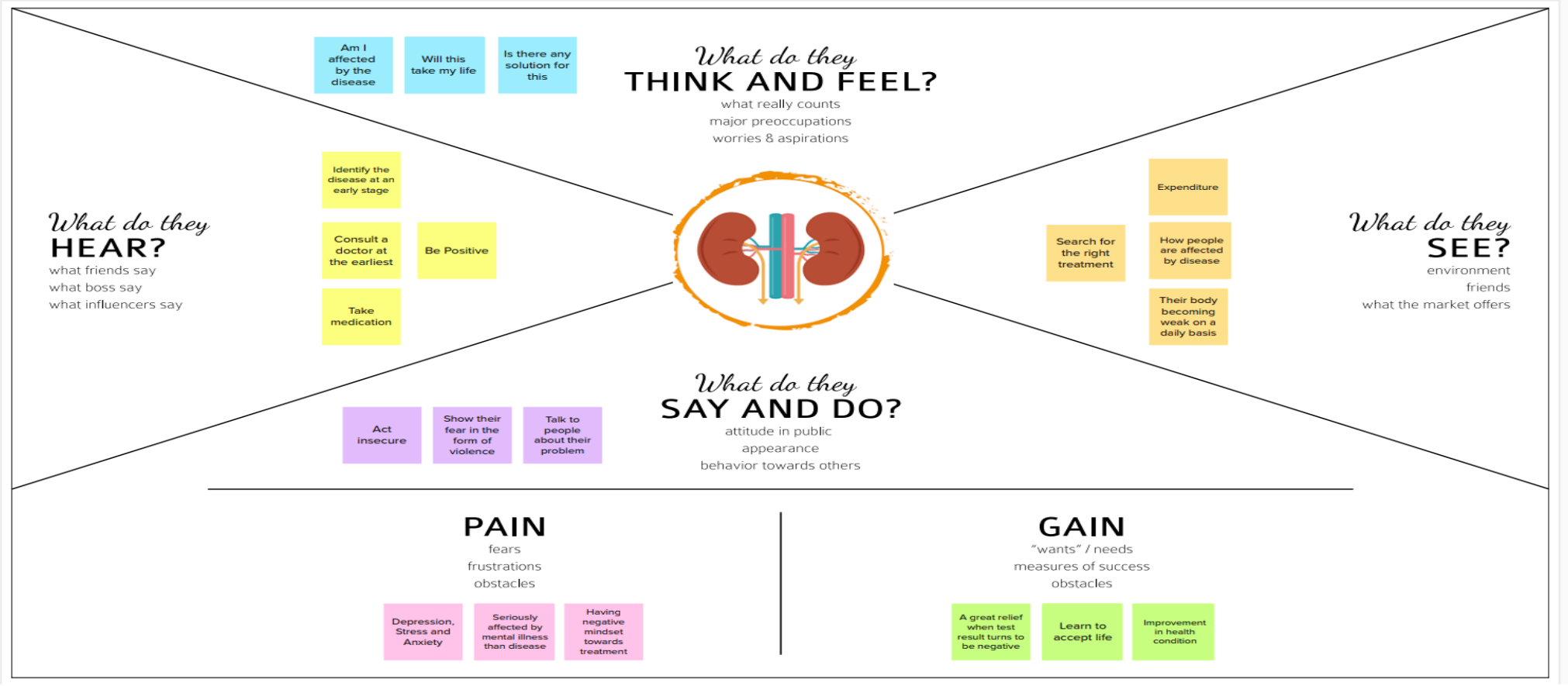
4	Prediction ofChronic Kidney Disease Using Machine Learning Algorithm	Siddheshwar Tekale,Pranjal Shingavi,Su kanya Wandhekar,Ankit Chatorikar	Decision tree algorithms along comparison withSVM	Decision tree – 91.75% SVM-96.75%	2018
5	Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study	Njoud Abdullah Almansour,H ajra FahimSyed, Nuha Radwan Khayat,Raw an KanaanAlthe eb,Renad Emad Juri,Jamal Alhiyafi,Sale h Alrashed,Su nday O.Olatunji	Comparative analysis was carried out on the two models-ANN and SVM	ANN - 99.75% SVM - 97.75%	2019

2.3 Problem Statement Definition

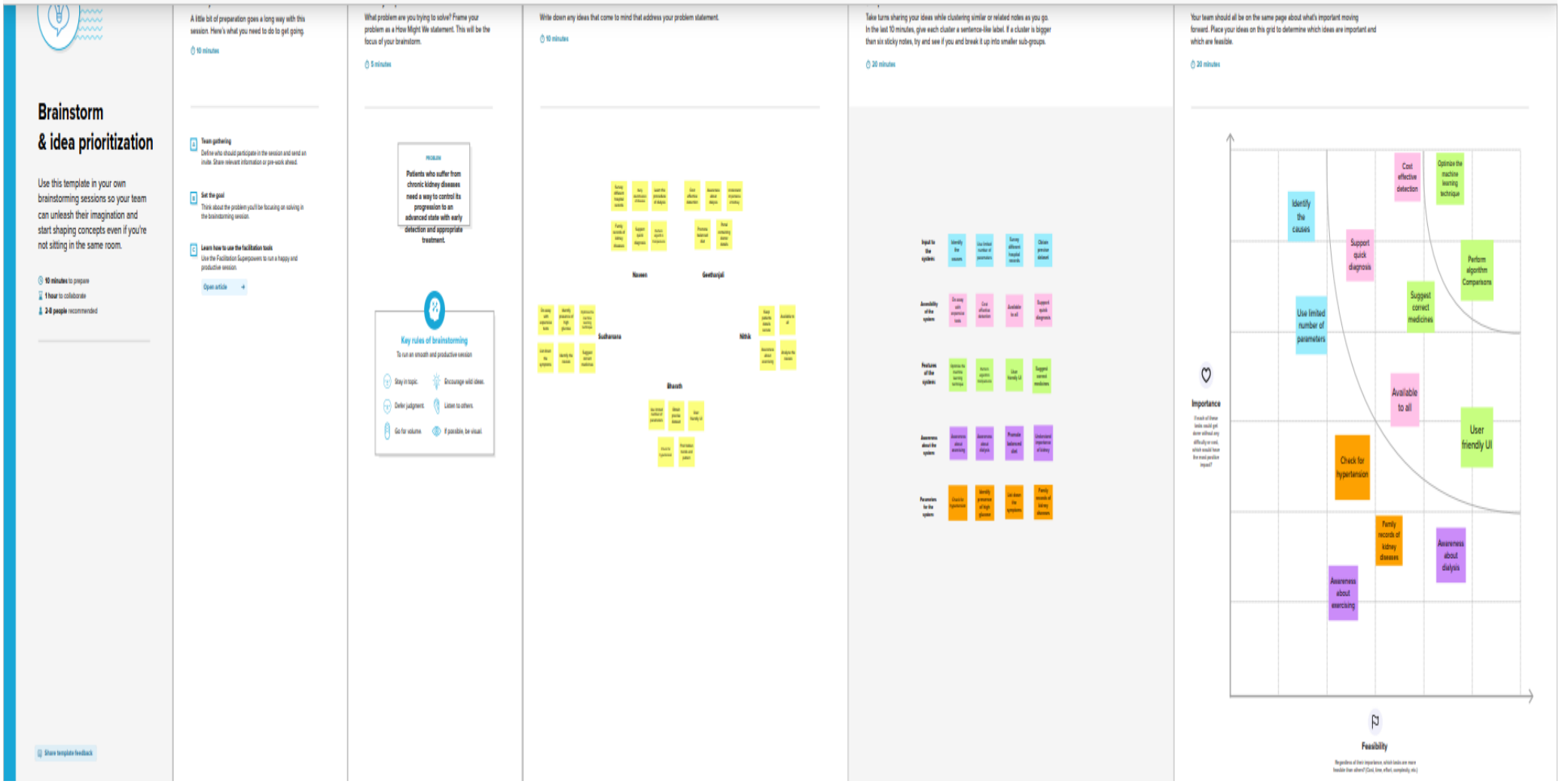
Non-communicable illnesses are the leading cause of early death, and CKD is the leading non-communicable disease. Chronic Kidney Disease is a major concern for the global health care system. People with CKD must focus on implementing proven, cost-effective therapies to as many people as possible while 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.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

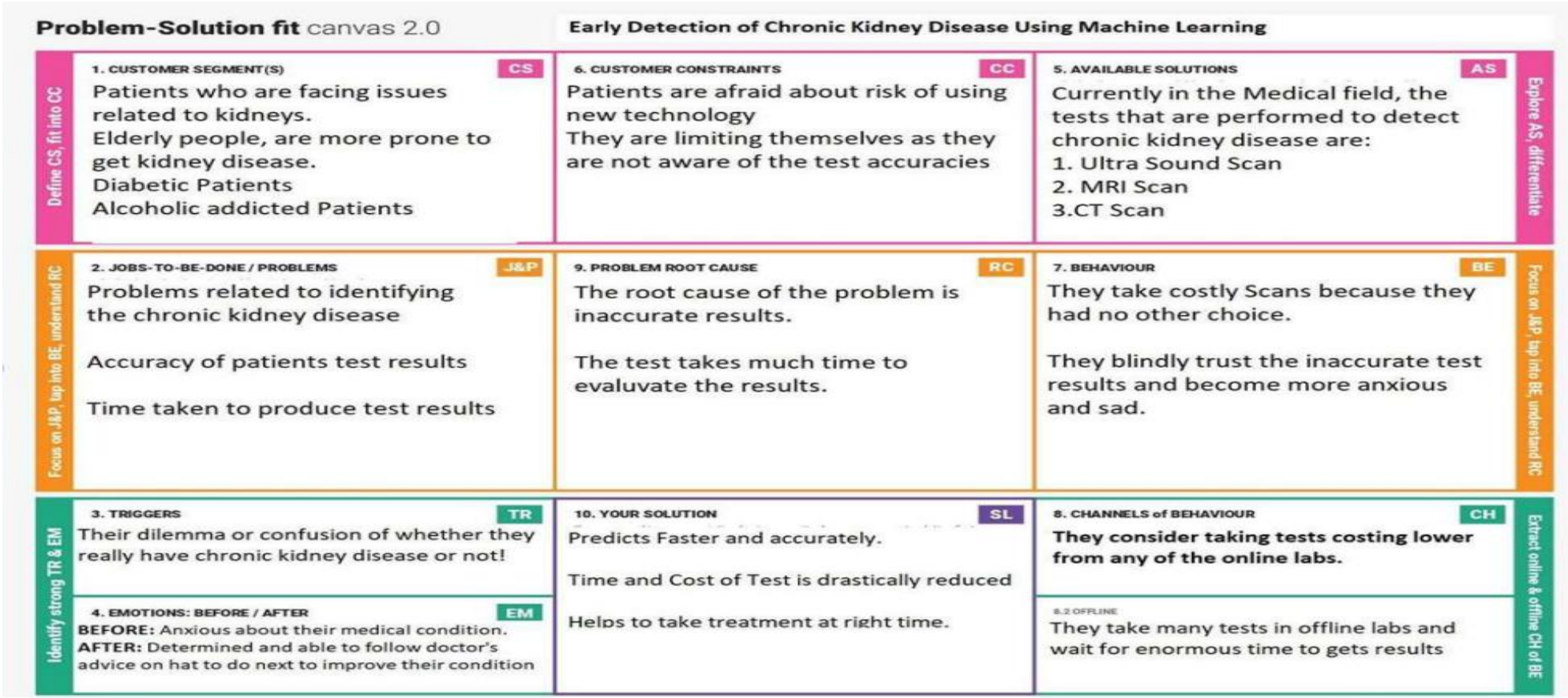


3.3 Proposed Solution

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 • maintaining the ml model by tweaking the parameter which doesn't play vital role in prediction by seeing the next set of dataset. • regular maintenance and changes in model with new features included in it

3.4 Problem Solution fit



4.REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	In order to become a new user, you will need to register through a form
FR-2	User Login	Users who already have credentials can log in with those credentials
FR-3	User Requirements	Past records can be stored in a database Create a report to indicate whether or not there is chronic kidney disease present A diagnostic remedy for the symptoms that you are experiencing
FR-4	User Entry	A form to be filled out in order to enter the results of pre-diagnostic tests
FR-5	Business Requirements	Diagnose CKD quickly with a quick blood test
FR-6	User Feedback	The user can submit feedback through a form on the website

4.2 Non-Functional requirements

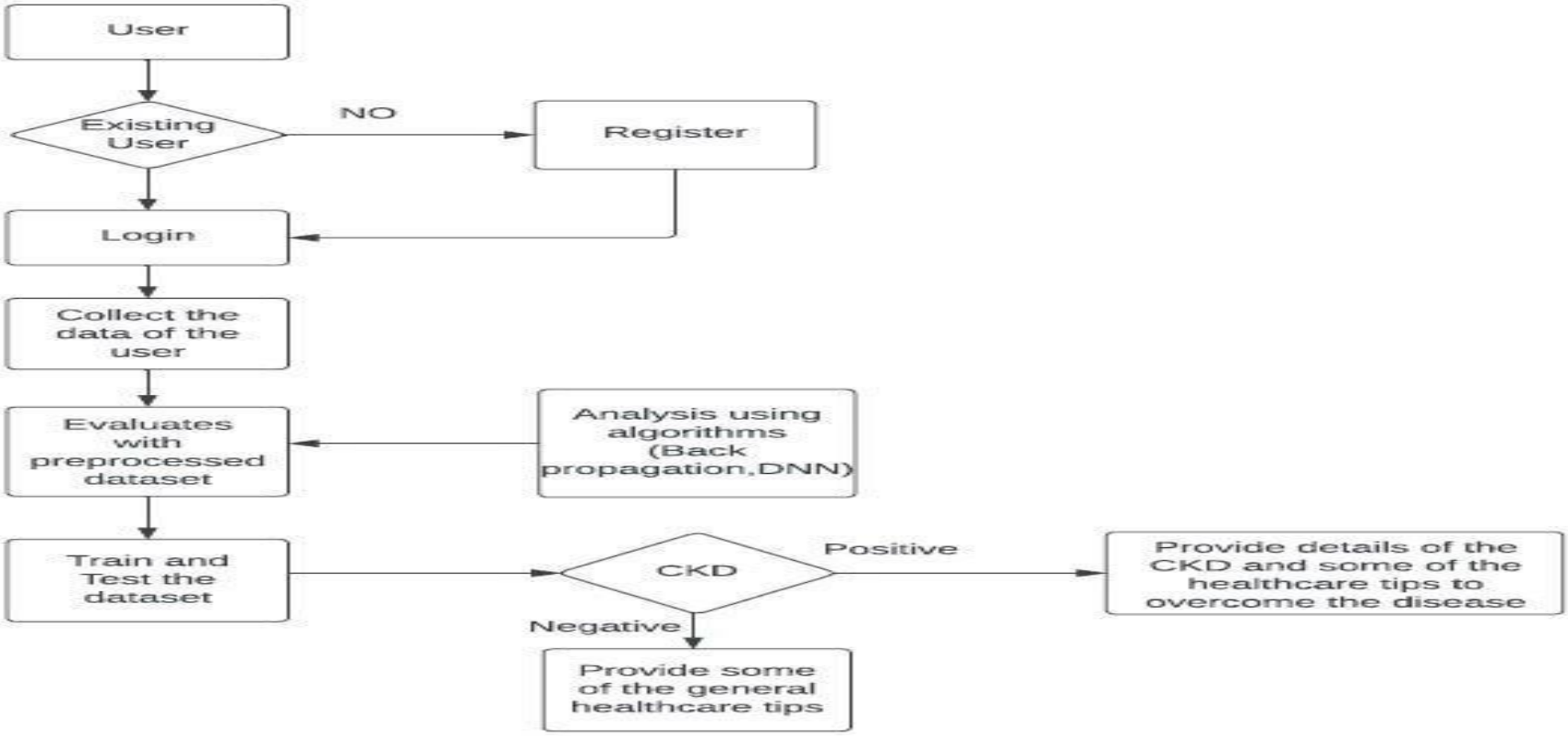
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Easy-to-use interface for communication that isuser-friendly
NFR-2	**Security**	Maintain the confidentiality of the details that users share with you
NFR-3	**Reliability**	A ML model must be able to predict probabilities with sufficient accuracy to provide a reliable diagnosis
NFR-4	**Performance**	A reduction in the overall time it takes for a diagnosisto be completed
NFR-5	**Availability**	The service is available to users from various locations at any time
NFR-6	**Scalability**	A large number of users need to be supported at the same time

5. PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flowswithin 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 store



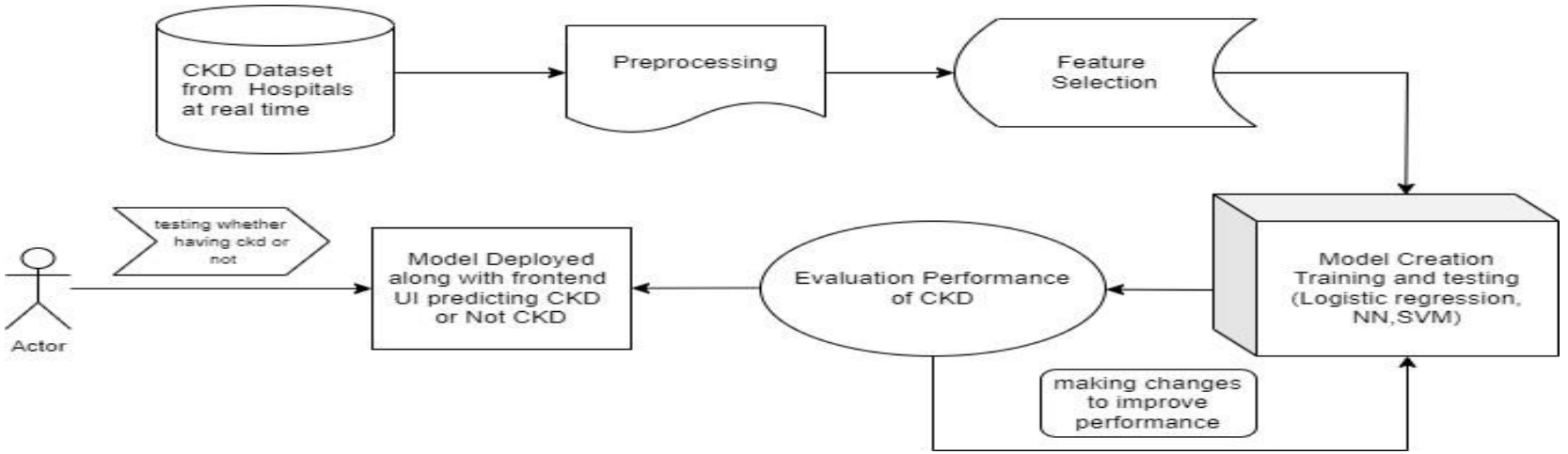
5.2 Solution & Technical Architecture

Solution Architecture

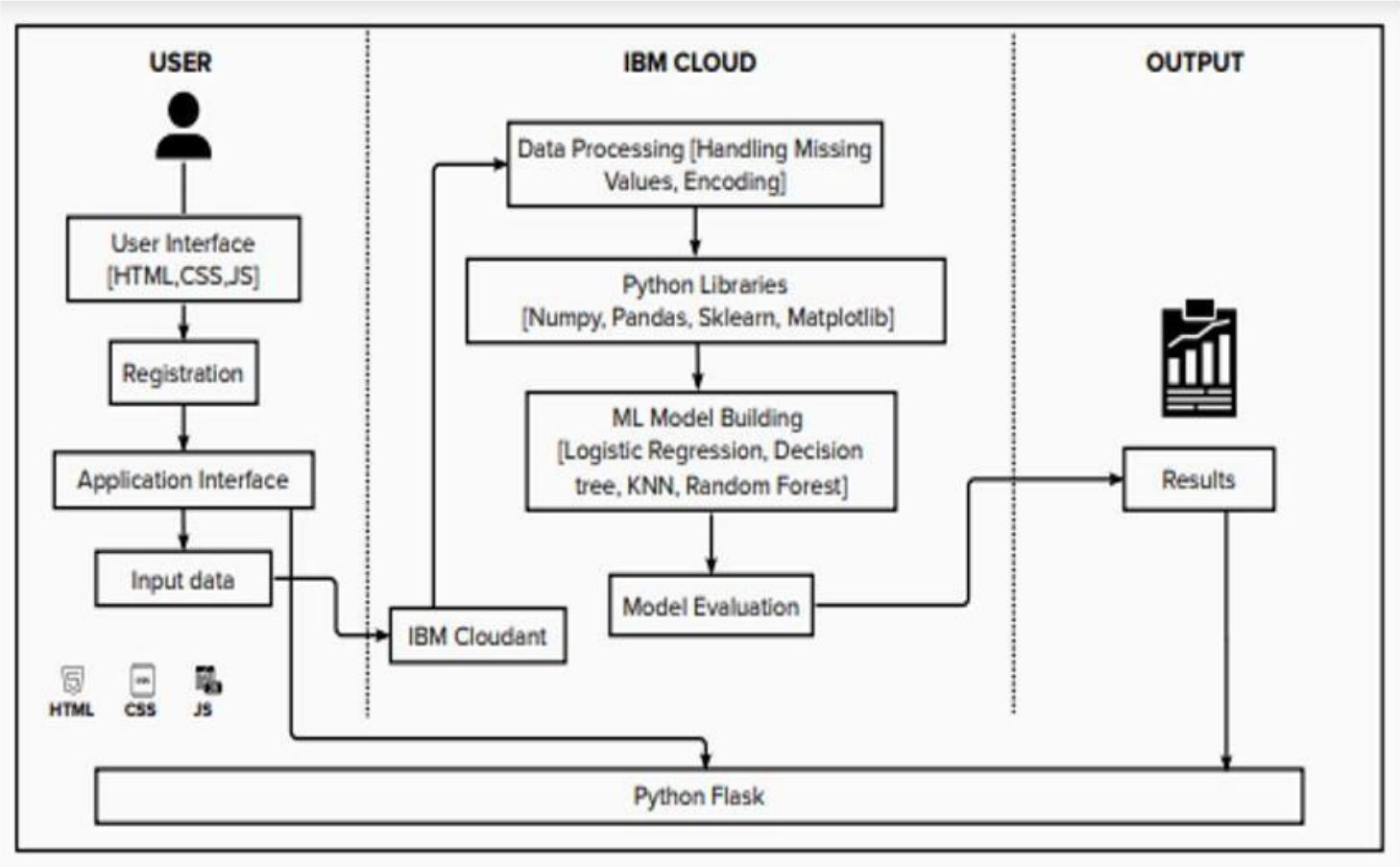
The Solution Architecture consist of the following elements/process for creation ofaccurate machine learning model they are as follows:-

- Dataset - The need of data is important to analyze the features for prediction
- Preprocessing Techniques -To handle missing data as well as standardizing the data for the model to process
- Feature Selection - Splitting the data into independent and dependent variables and dropping unwanted features using dimensionality reduction or dropping categorical variable with no importance
- Model creation and evaluation - model is created with various ml algorithm aiming for greater accuracy and evaluation is done repeatedly to improve accuracy
- Deploying model - model is deployed for the access of the service over the internet by customer
- User/customer-main actors for accessing the service

Solution Architecture Diagram:



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 & SCHEDULING

6.1 Sprint Planning & Estimation

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

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	As a user, I can register for the diagnosis tool using my email and password	7	High	Team Lead Member 1

Sprint-1		USN-2	As a user, I will receive confirmation email on registering for the diagnosis tool	6	High	Member 1 Member 2
Sprint-4		USN-3	As a user, I can register for the application through my Gmail	6	Low	Member 1 Member 2
Sprint-1	Login	USN-4	As a user, I can log into the application by entering my credentials	6	High	Team Lead Member 1
Sprint-3	Dashboard	USN-5	As a user, I can see my past records and activities	6	High	Team Lead Member 2
Sprint-2	Entry form	USN-6	As a user, I must enter my pre-diagnostic test results	7	High	Team Lead Member 1
Sprint-3	Report	USN-7	As a user, I can view the report generated by the tool	7	High	Member 2 Team Lead

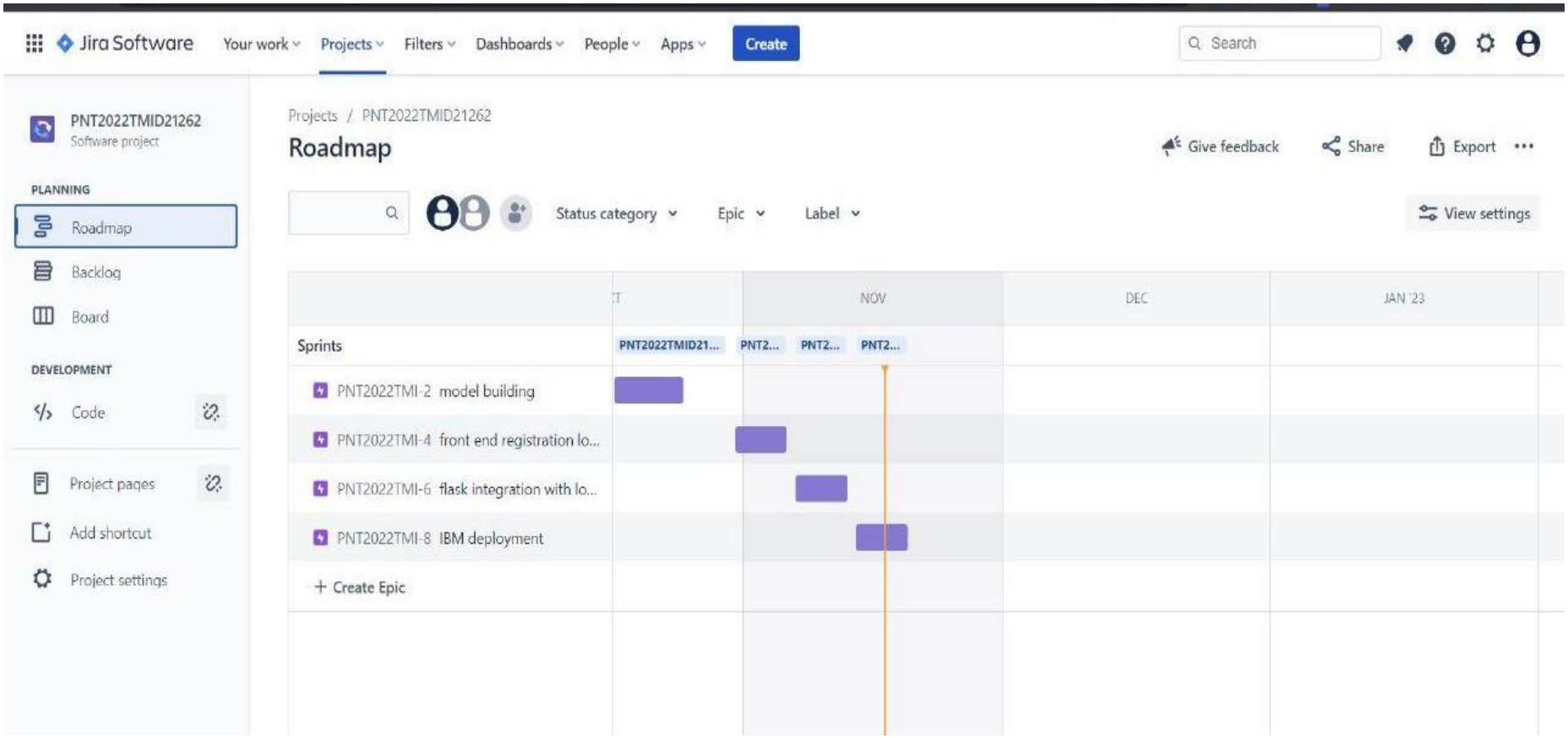
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**

Sprint-3	Remedies	USN-8	As a user, I will receive remedies to treat mysymptoms	6	Medium	Member 1 Member 2
Sprint-4	Queries	USN-9	As a customer care executive, I must assistusers that face problems through Q&A	6	Low	Member 1 Member 2
Sprint-4	Feedback	USN-10	As a customer care executive, I should get inputfor the tool's enhancement from users	7	Low	Team Lead Member 1
Sprint-2	Feature importance	USN-11	As an administrator, I should identify the mostsignificant factors that lead to CKD based on the present trend	6	High	Member 2 Member 1
Sprint-2	Train model	USN-12	As an administrator, I must use the mostsuitable ML model for detection of CKD	6	High	Team Lead Member 2

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

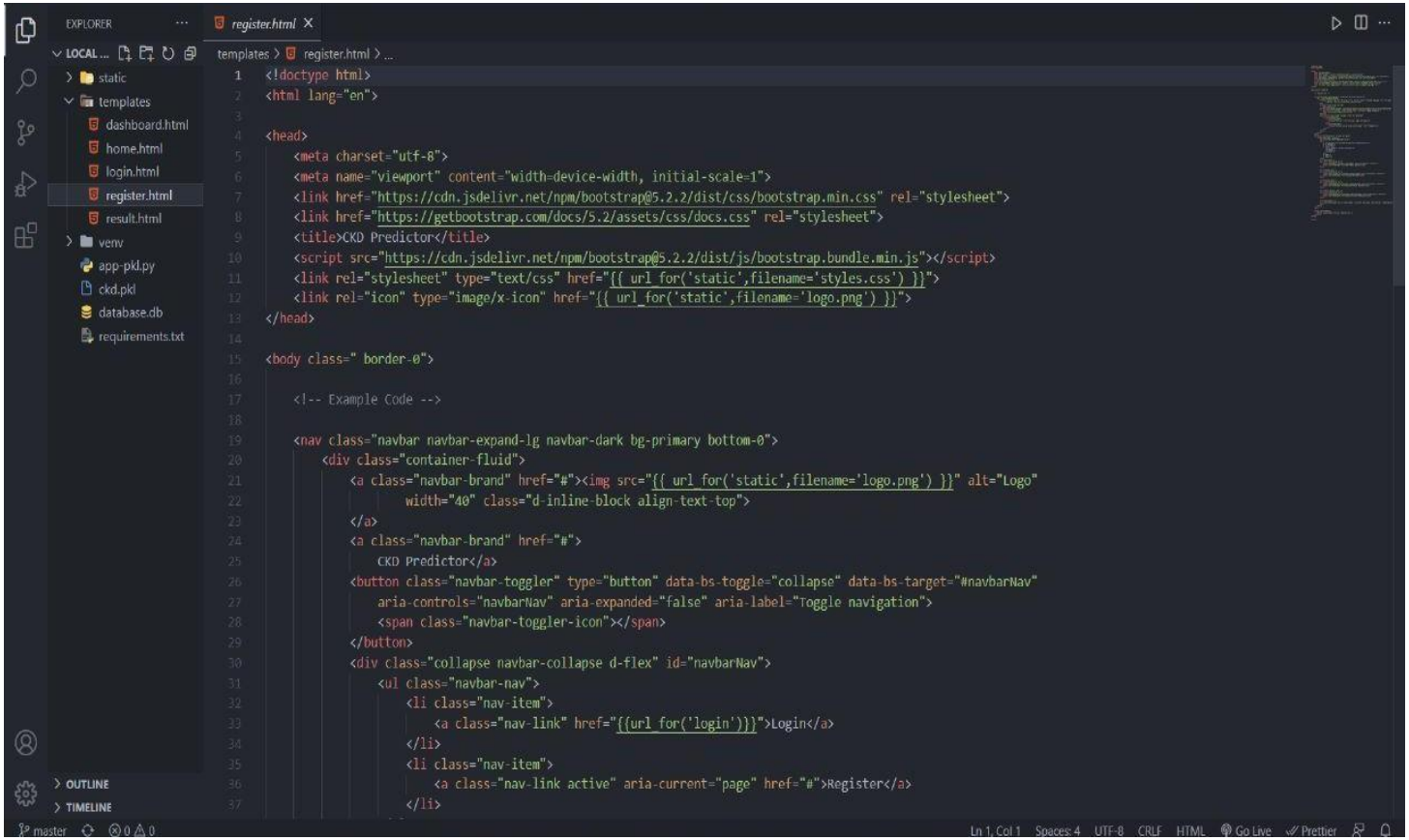
6.3 Reports from JIRA



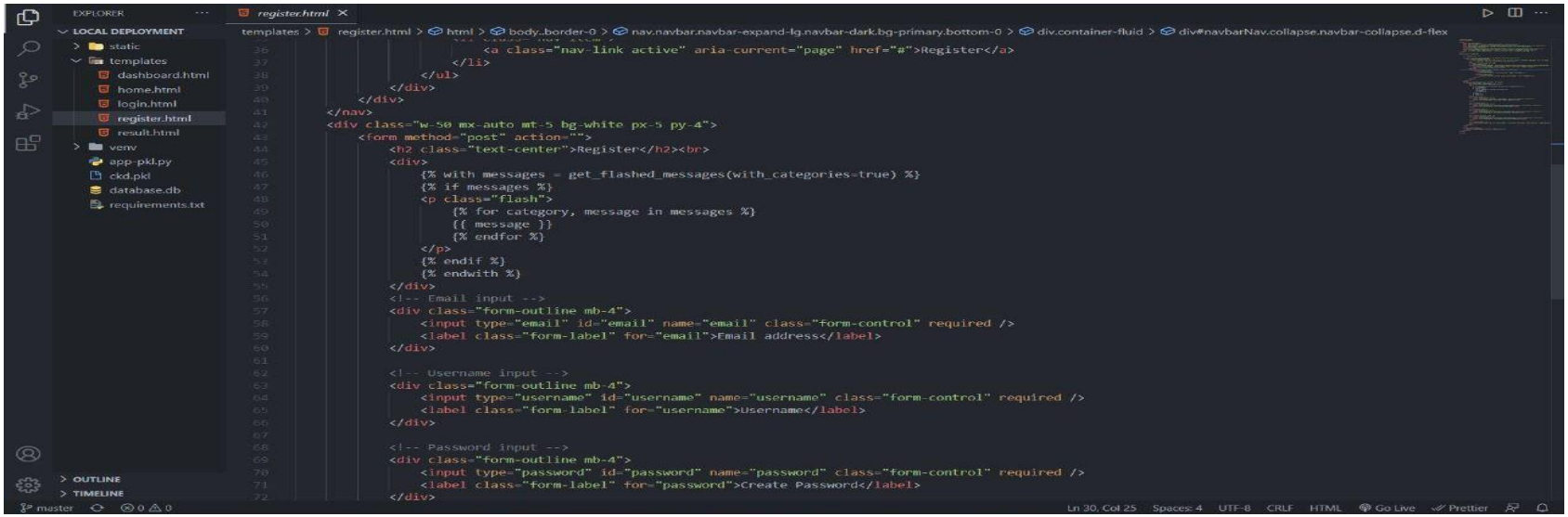
7. CODING & SOLUTIONING

User Registration and login :

Register.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>CKD Predictor</title>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"></script>
    <link rel="stylesheet" type="text/css" 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" href="{{url_for('login')}}">Login</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link active" aria-current="page" href="#">Register</a>
                    </li>
```


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>

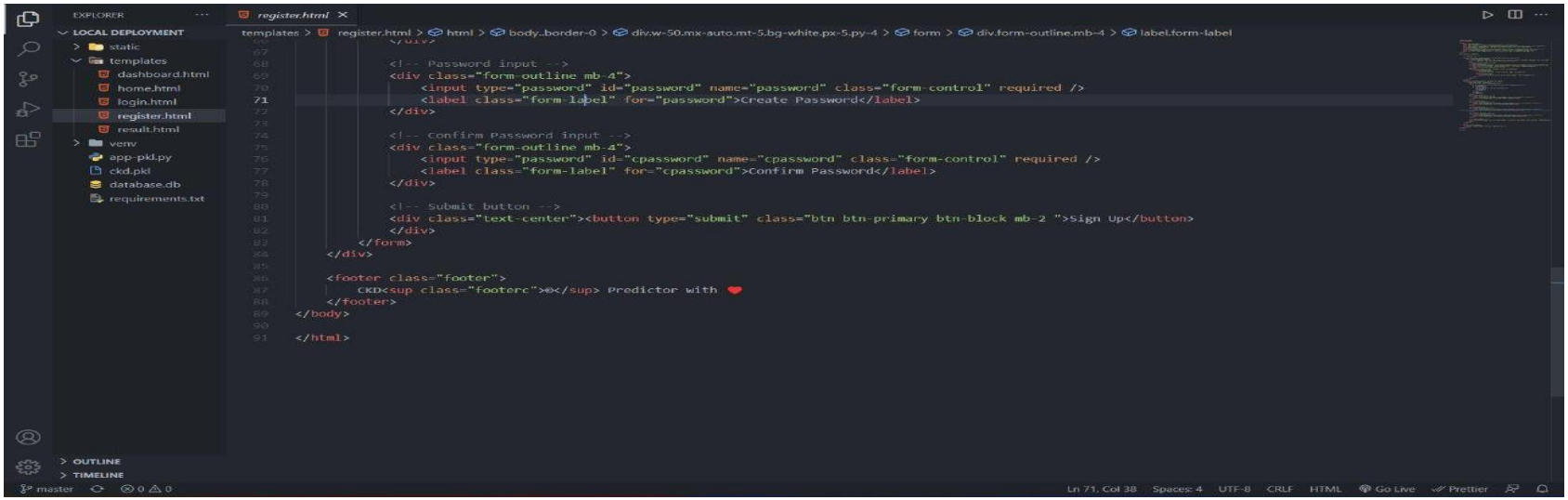
<!-- Username input -->

<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>

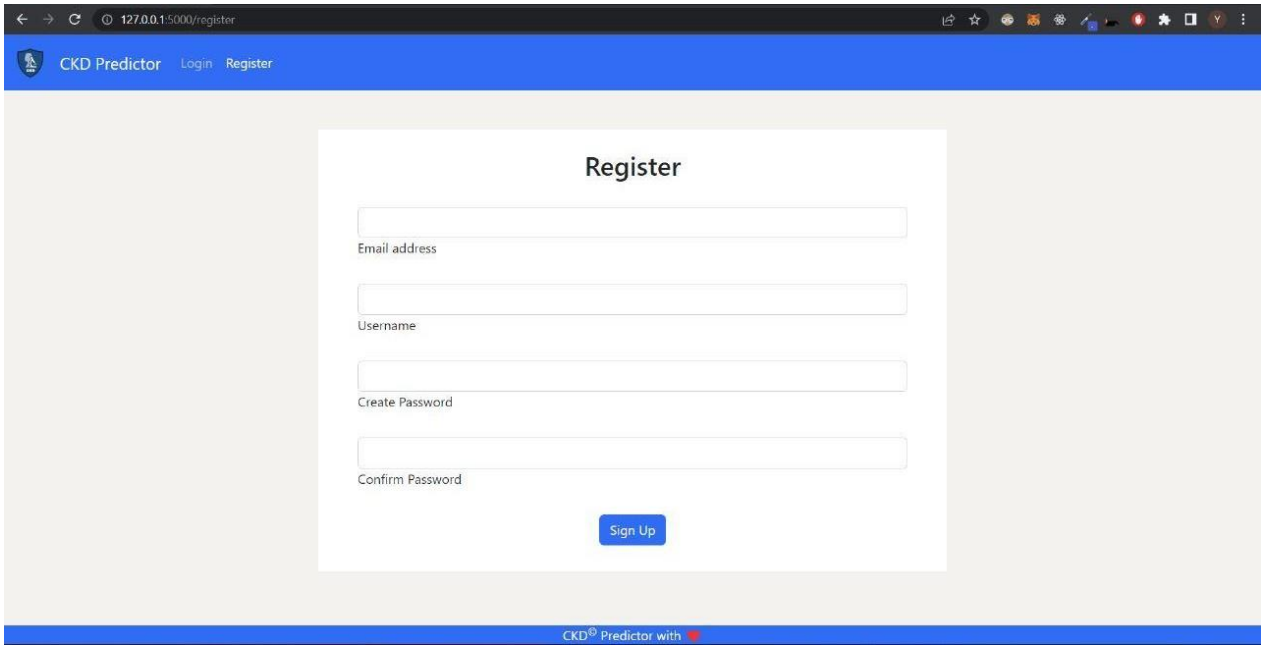
<!-- Confirm Password input -->

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

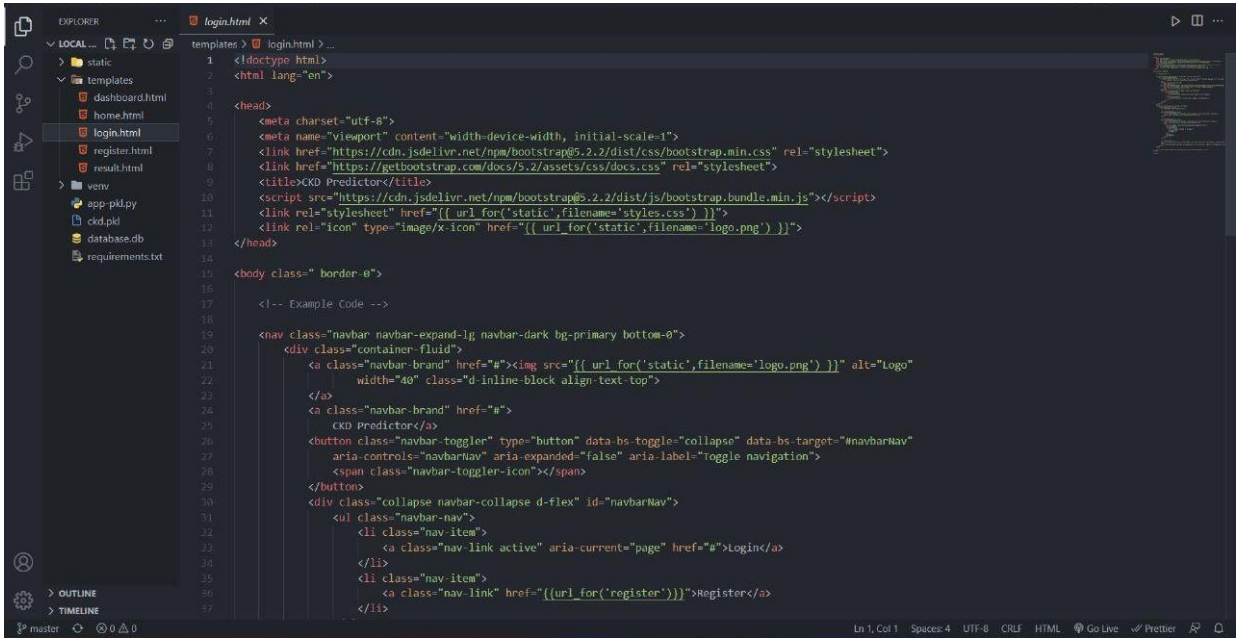
<!-- Submit button -->
<button type="submit" class="btn btn-primary btn-block mb-2 ">Sign Up</button>

</form>

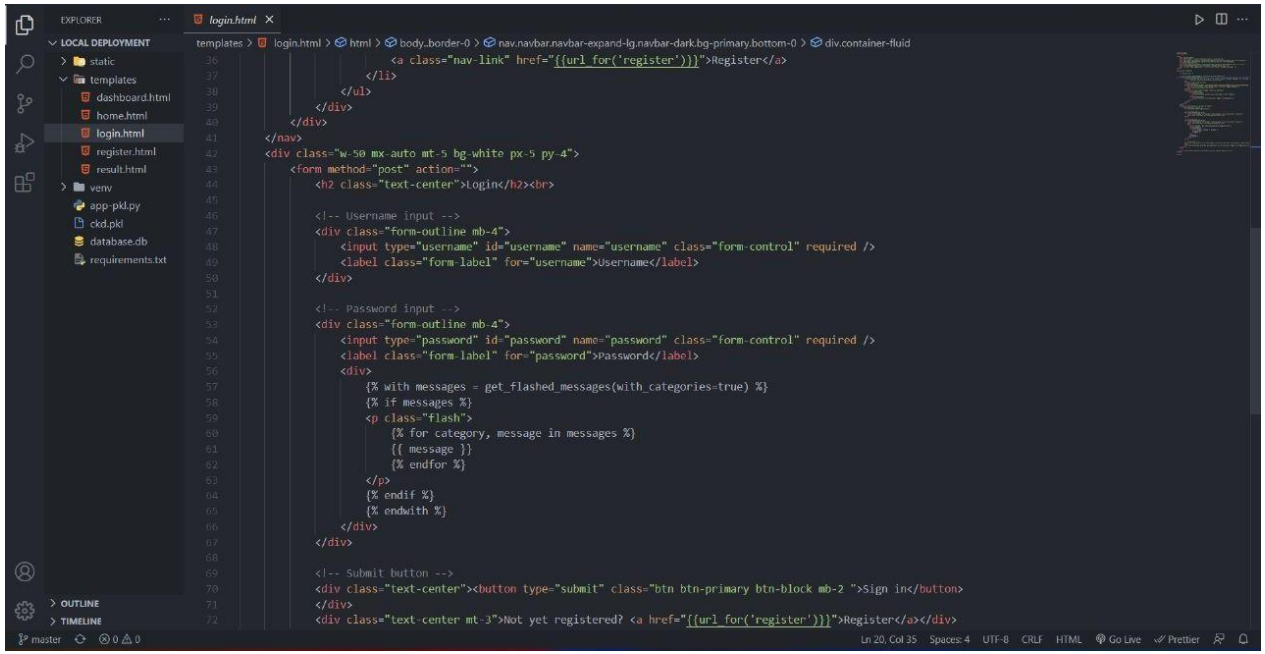
<footer class="footer">
CKD® Predictor with
</footer>
</body>
</html>
master
Ln 71, Col 38
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier



Login.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>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" aria-current="page" href="#">Login</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="{{url_for('register')}}">Register</a>
                    </li>
```


EXPLORER
login.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 > login.html > html > body..border-0 > nav.navbar.navbar-expand-lg.navbar-dark.bg-primary.bottom-0 > div.container-fluid
Register

</nav>

<form method="post" action="">
<h2 class="text-center">Login</h2>
<!-- 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">Password</label>

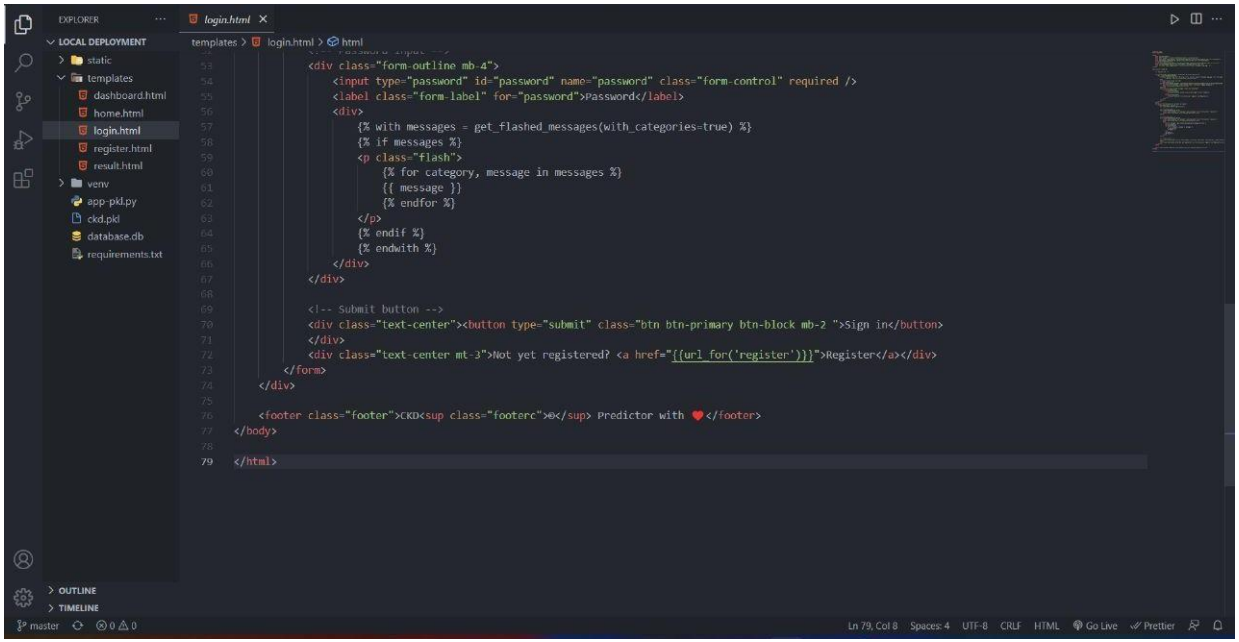
{% with messages = get_flashed_messages(with_categories=true) %}
{% if messages %}

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

{% endif %}
{% endwith %}

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

Not yet registered? Register
master
Ln 20, Col 35
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier


EXPLORER
login.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 > login.html > html

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

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

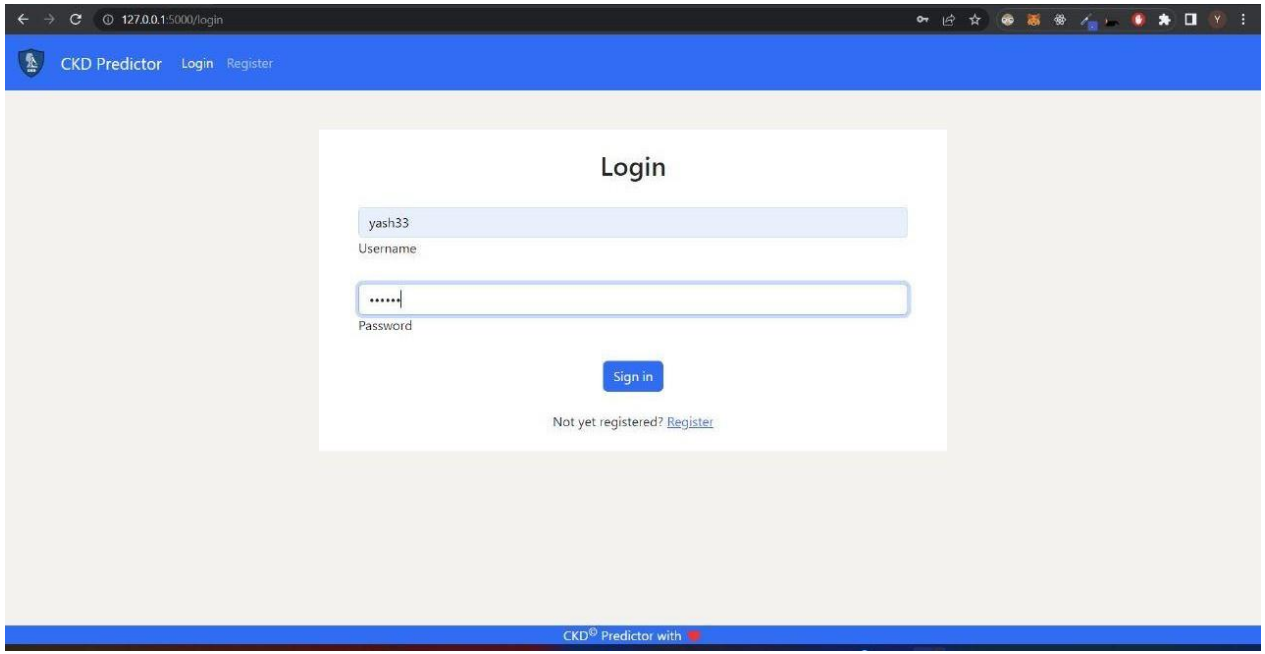
{% for category, message in messages %}
{{ message }}
{% endfor %}

{% endif %}
{% endwith %}

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

Not yet registered? Register
</form>

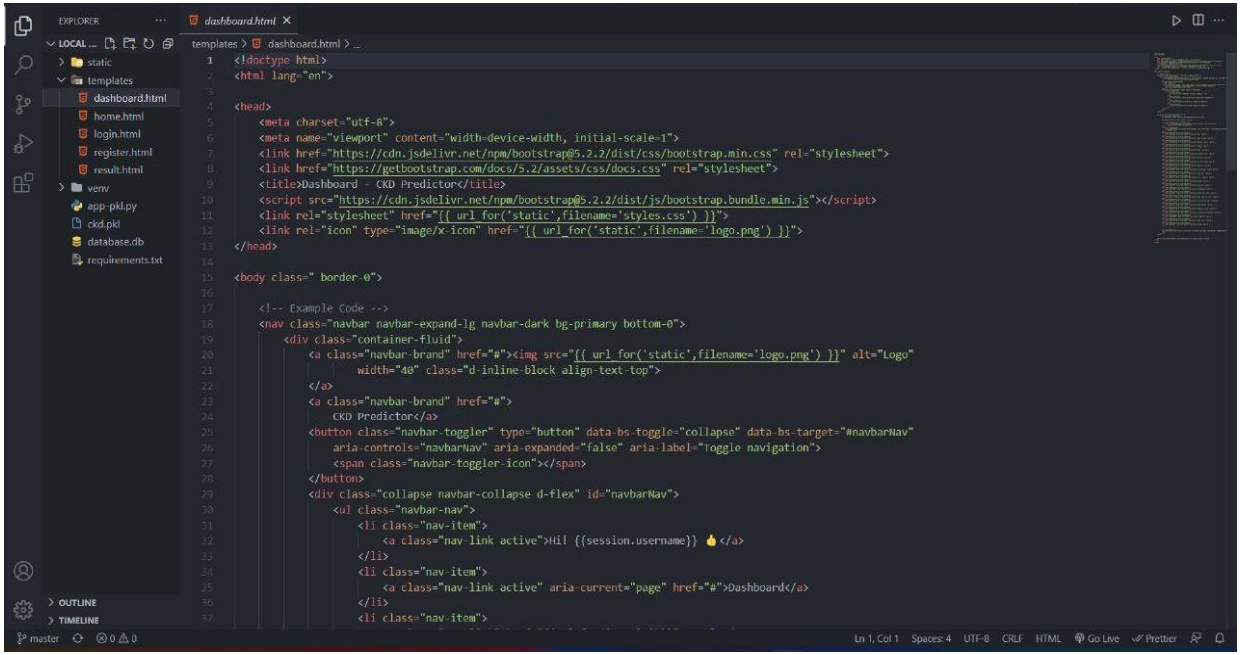
<footer class="footer">CKD^ɑ Predictor with ❤</footer>
</body>
</html>
master
Ln 79, Col 8
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier


127.0.0.1:5000/login
CKD Predictor
Login
Register
Login
yash33
Username
Password
Sign in
Not yet registered? Register
CKD© Predictor with

Dashboard and Result

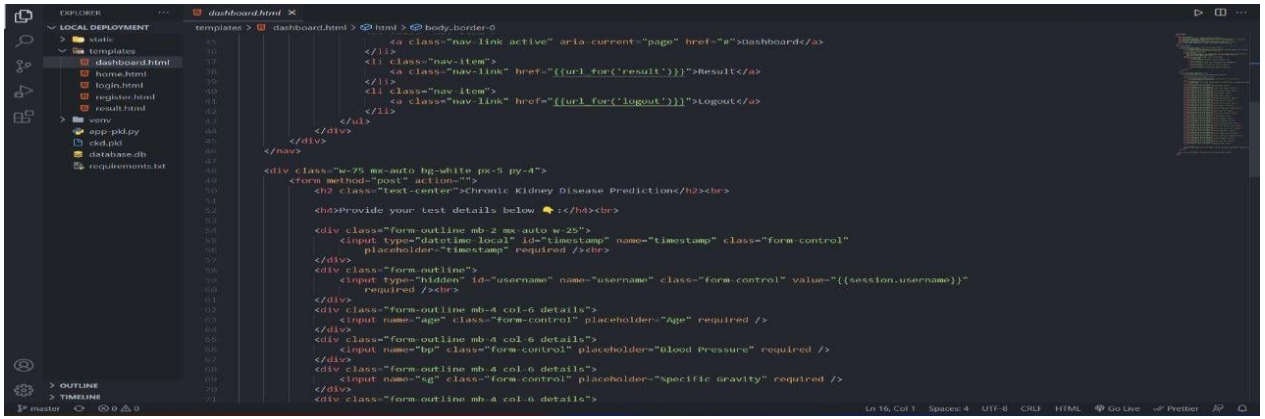
Dashboard.html

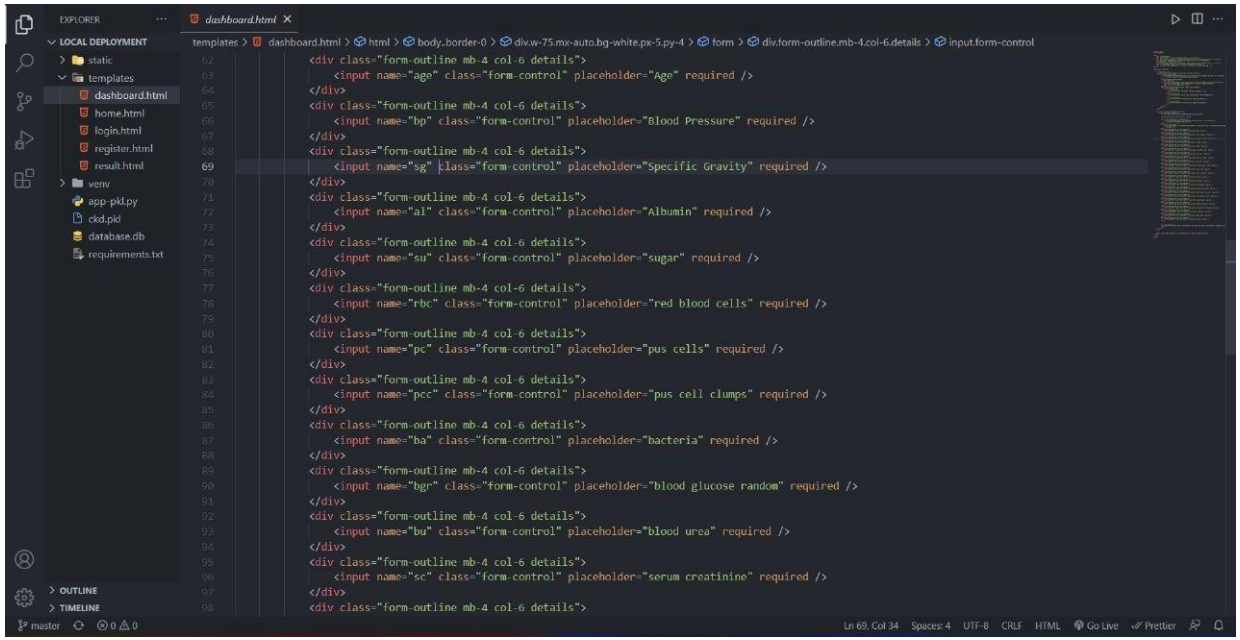
2.

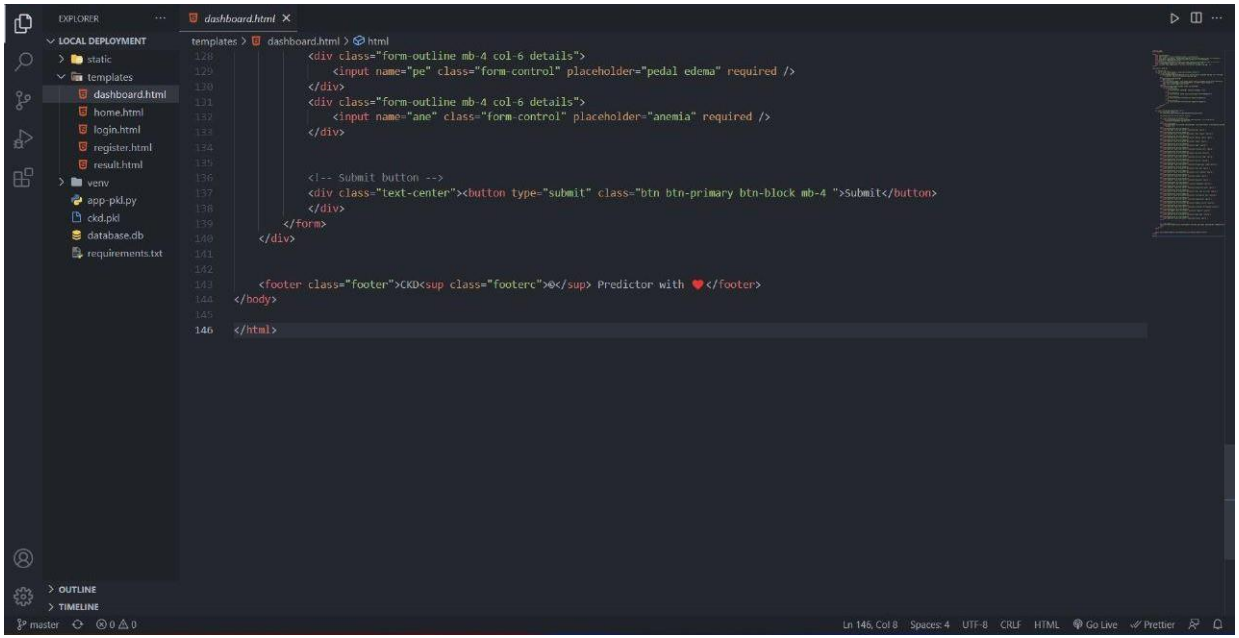



```
<!doctype html>
<html lang="en">

<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">
```

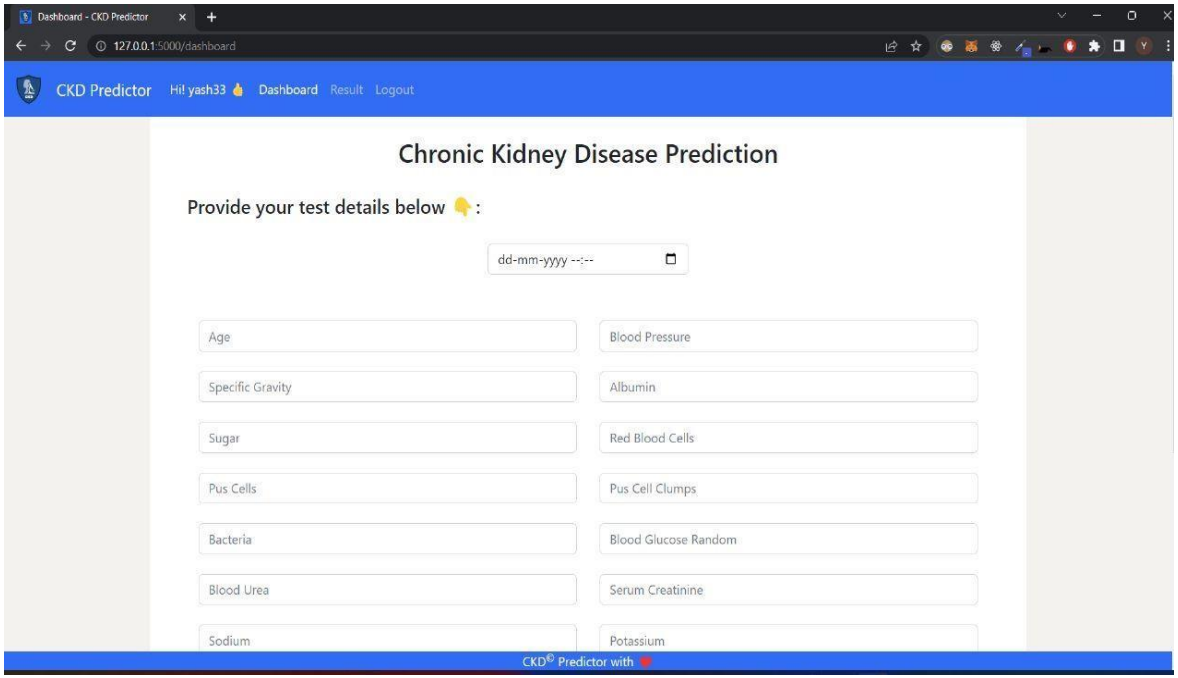
3.


EXPLORER
dashboard.html
templates > dashboard.html > html > body_border-0
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 16, Col 1
Spaces: 4
UTF-8
CRLF
HTML
Go Live
Prettier

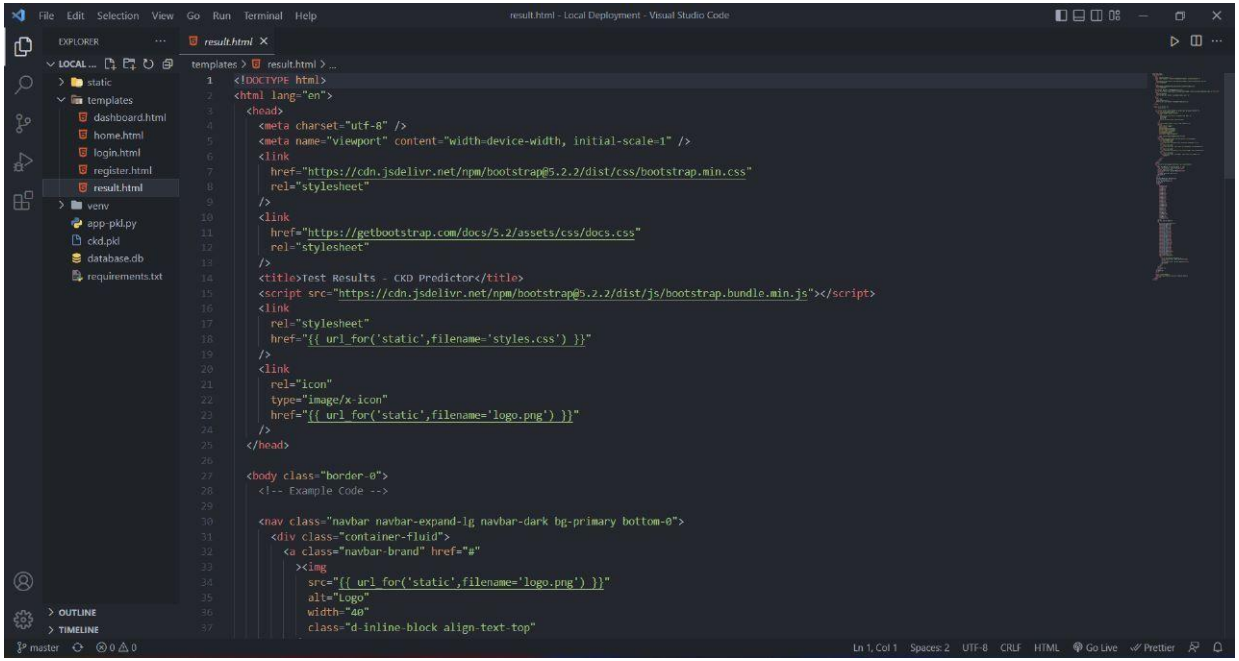

EXPLORER
dashboard.html
templates > dashboard.html > html > body_border-0 > div.w-75.mx-auto.bg-white.px-5.py-4 > form > div.form-outline.mb-4.col-6.details > input.form-control
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db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



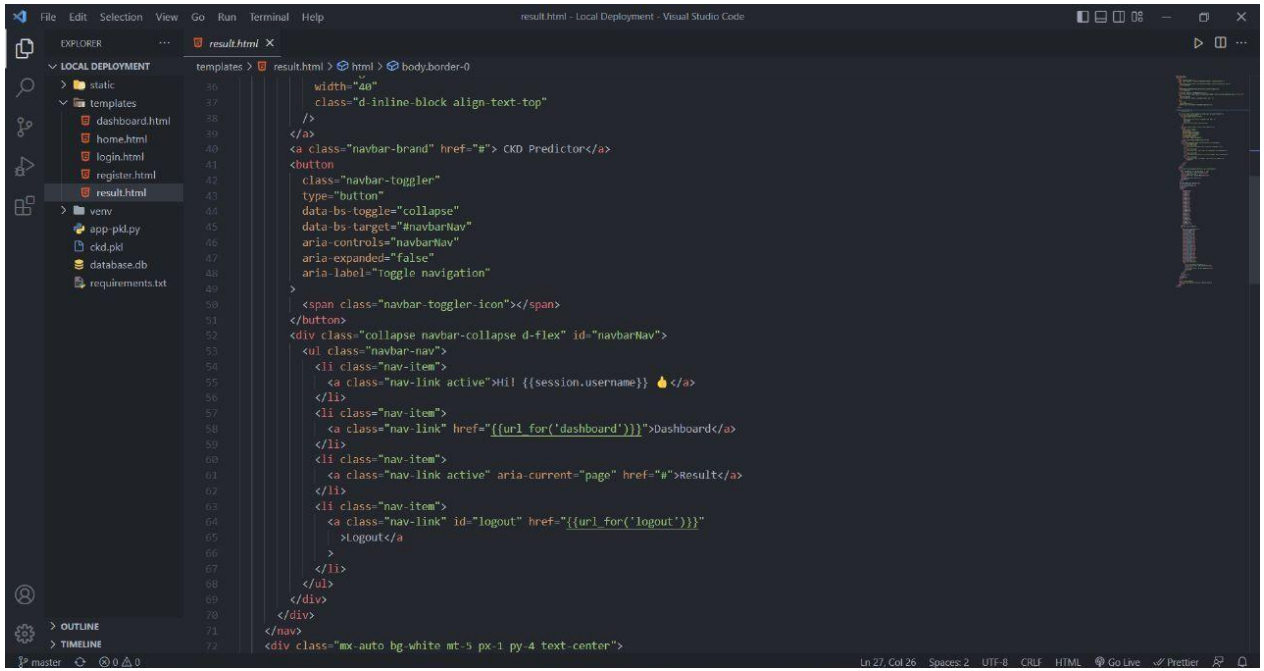

```
            <div class="form-outline mb-4 col-6 details">
                <input name="pe" class="form-control" placeholder="pedal edema" required />
            </div>
            <div class="form-outline mb-4 col-6 details">
                <input name="ane" class="form-control" placeholder="anemia" required />
            </div>


            <!-- Submit button -->
            <div class="text-center"><button type="submit" class="btn btn-primary btn-block mb-4 ">Submit</button>
            </div>
        </form>
    </div>


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

</html>
```


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


File Edit Selection View Go Run Terminal Help
result.html - Local Deployment - Visual Studio Code
EXPLORER
result.html
templates > result.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
master
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier
<!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"


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}} 👍

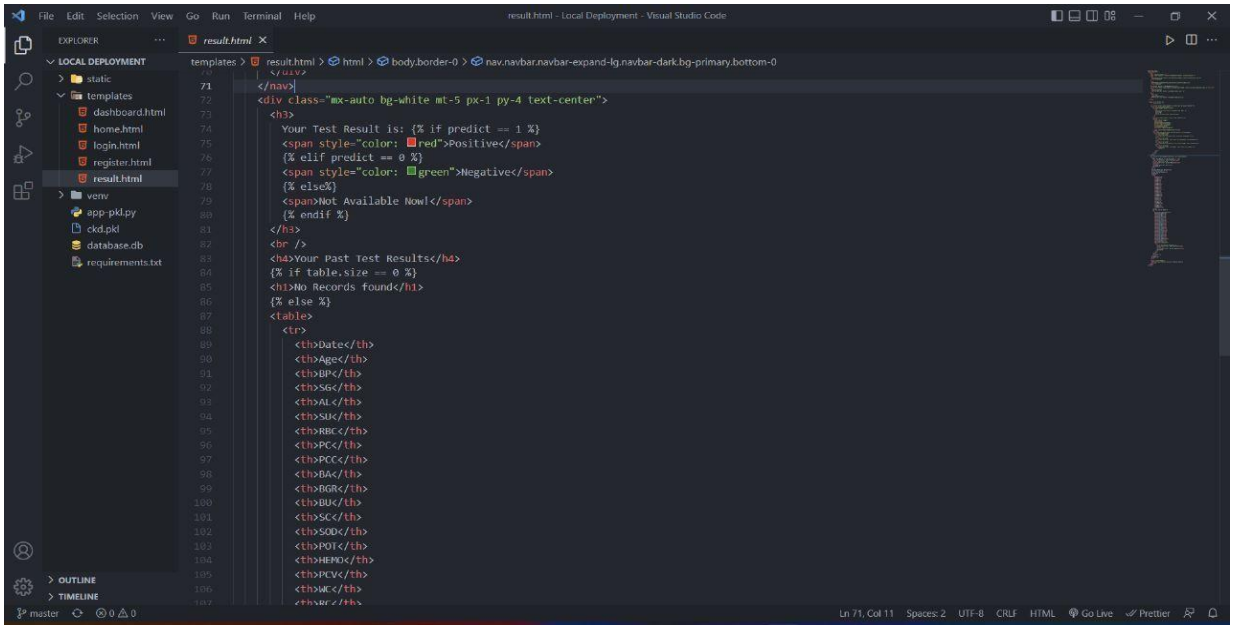
<li class="nav-item">
Dashboard

<li class="nav-item">
Result

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

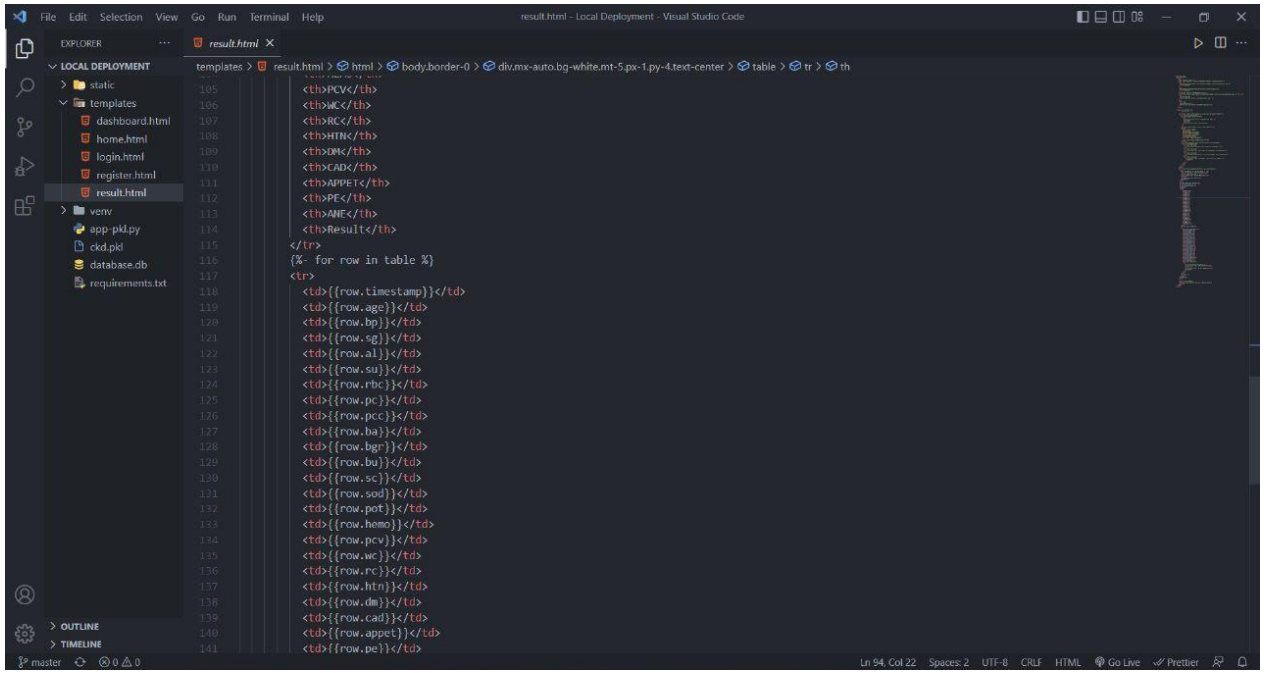
</nav>

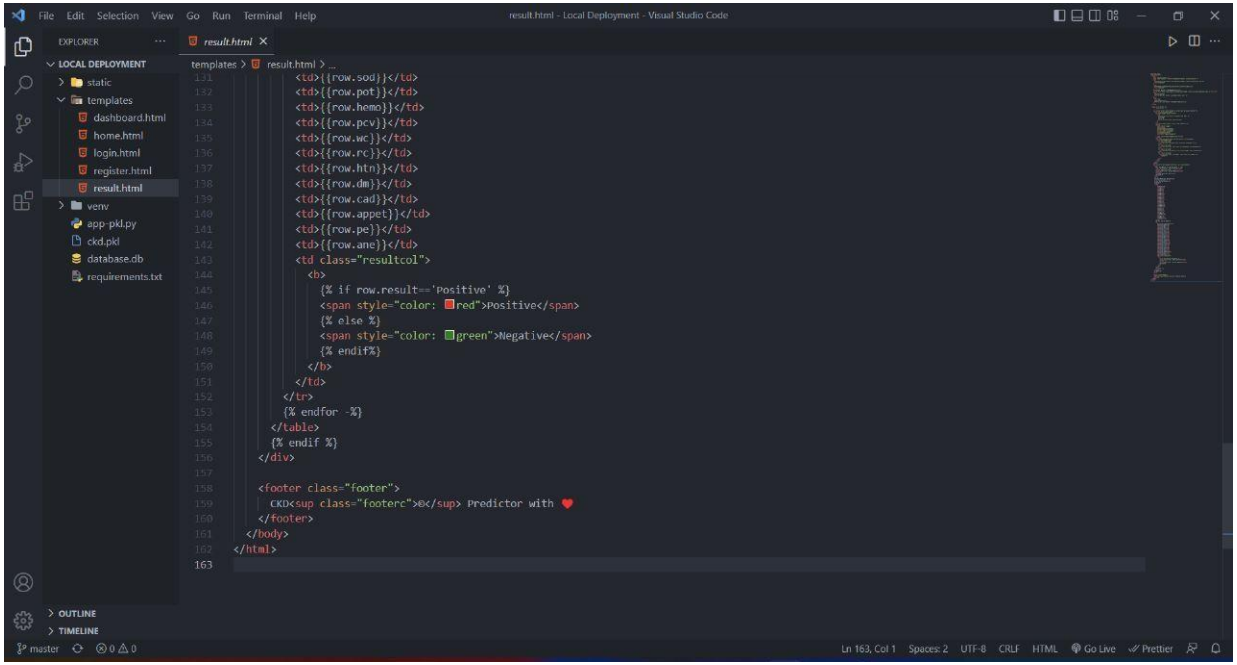
master
Ln 27, Col 26 Spaces: 2 UTF-8 CRLF HTML Go Live Prettier


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>BUC</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

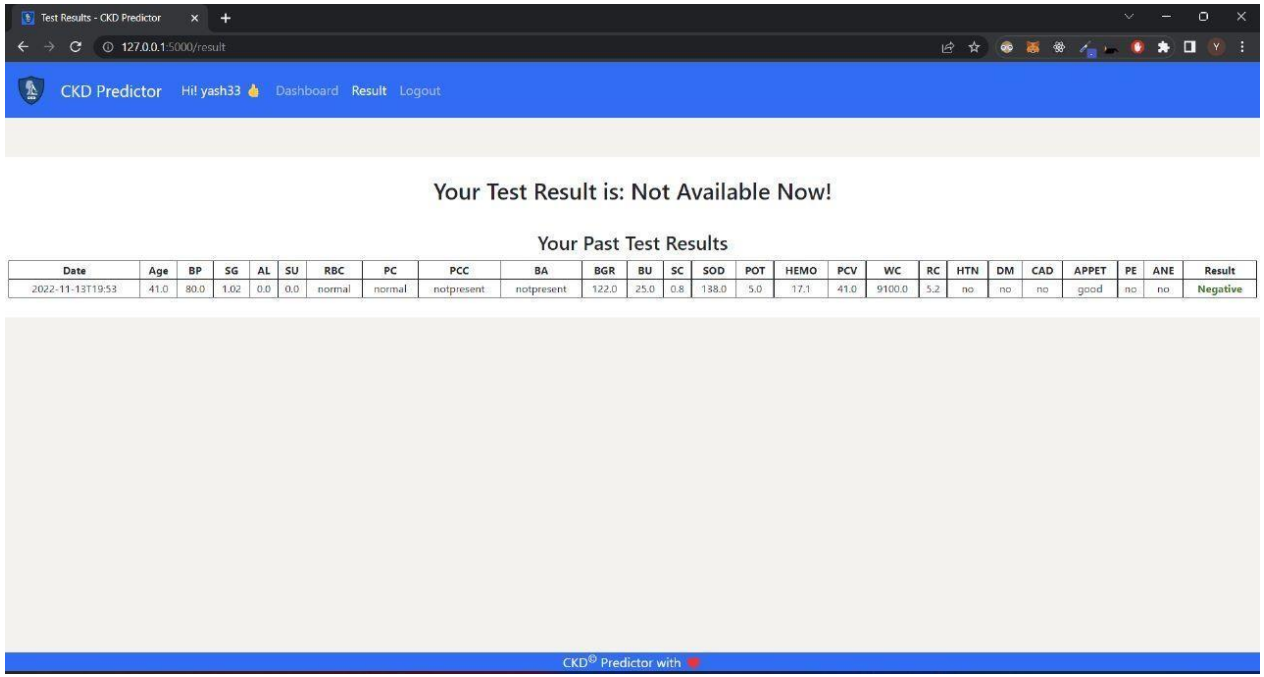

File Edit Selection View Go Run Terminal Help
result.html - Local Deployment - Visual Studio Code
EXPLORER
result.html
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
templates > result.html > 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>
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">

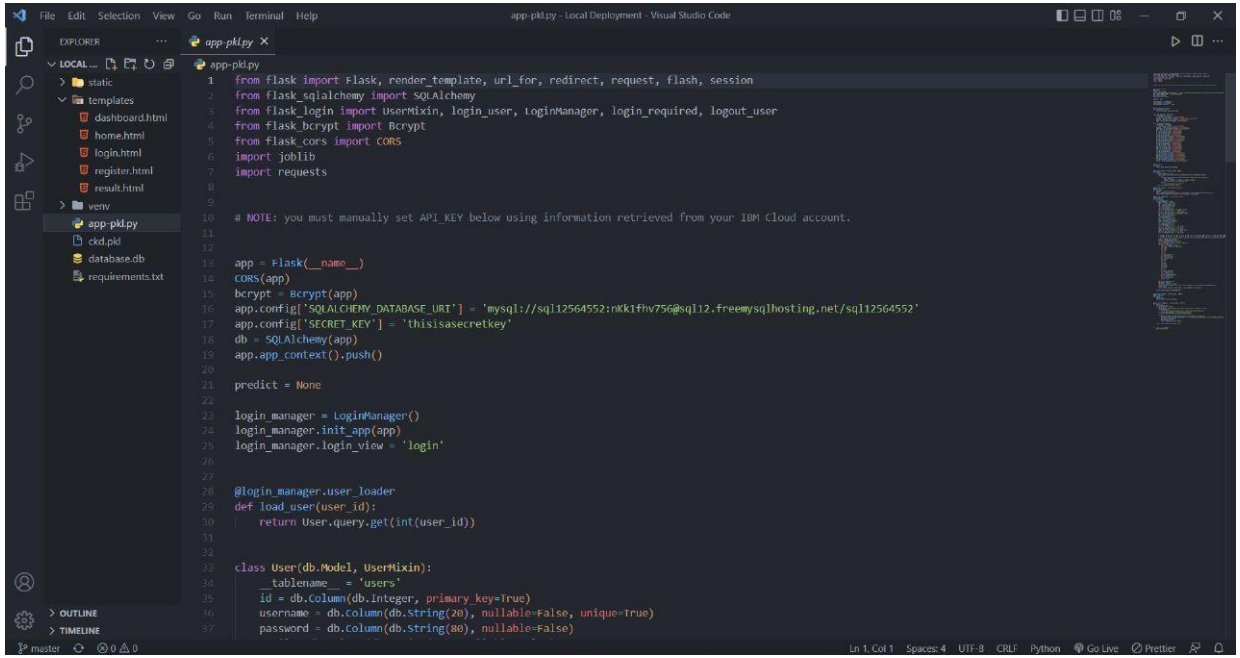
{% if row.result=='Positive' %}
Positive
{% else %}
Negative
{% endif%}

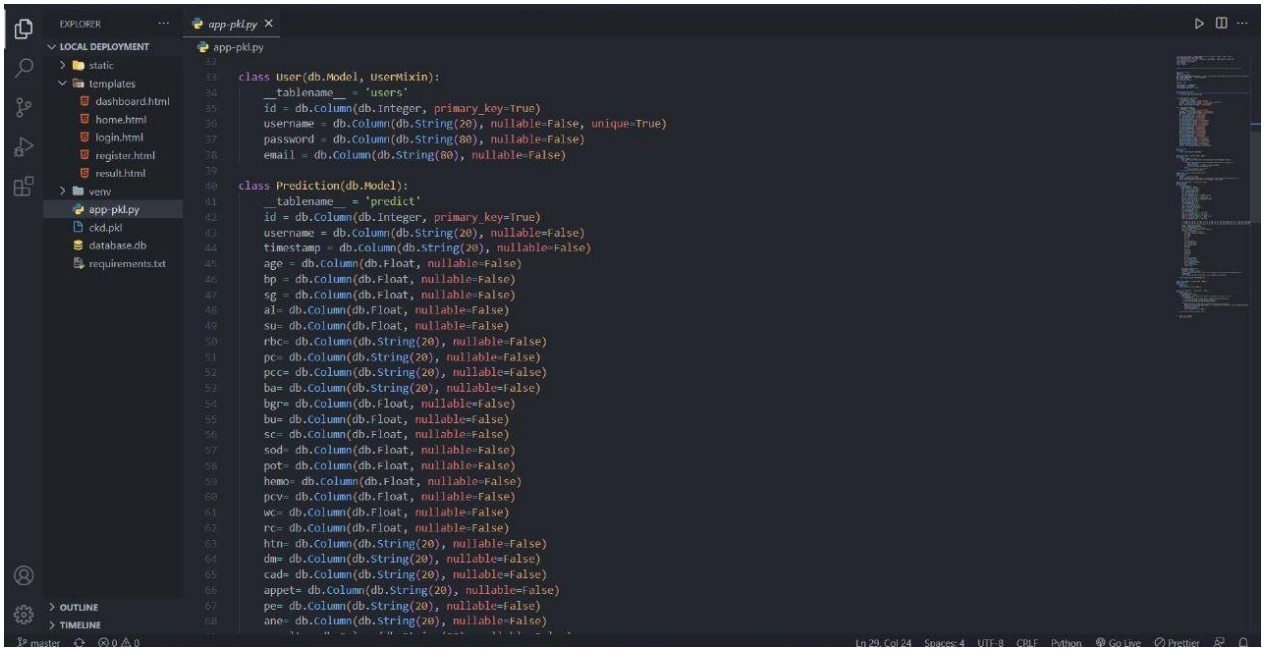
</td>
</tr>
{% endfor -%}
</table>
{% endif %}

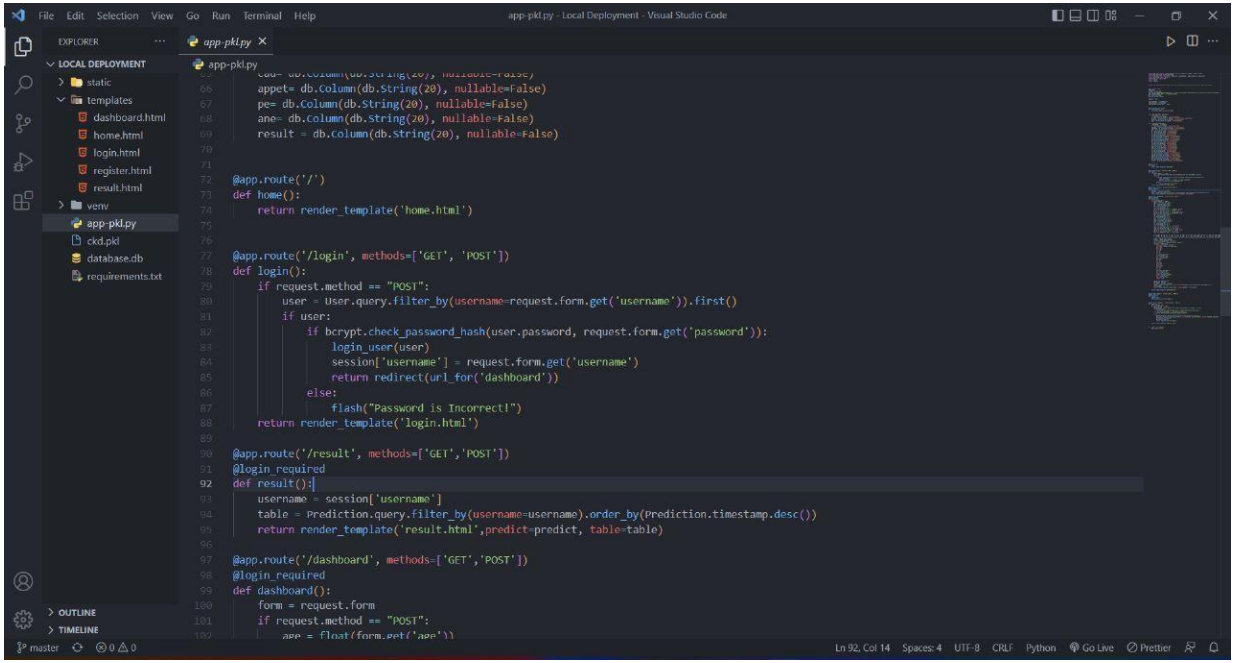
<footer class="footer">
CKD© Predictor with
</footer>
</body>
</html>
OUTLINE
TIMELINE
master
Ln 163, Col 1
Spaces: 2
UTF-8
CRLF
HTML
Go Live
Prettier

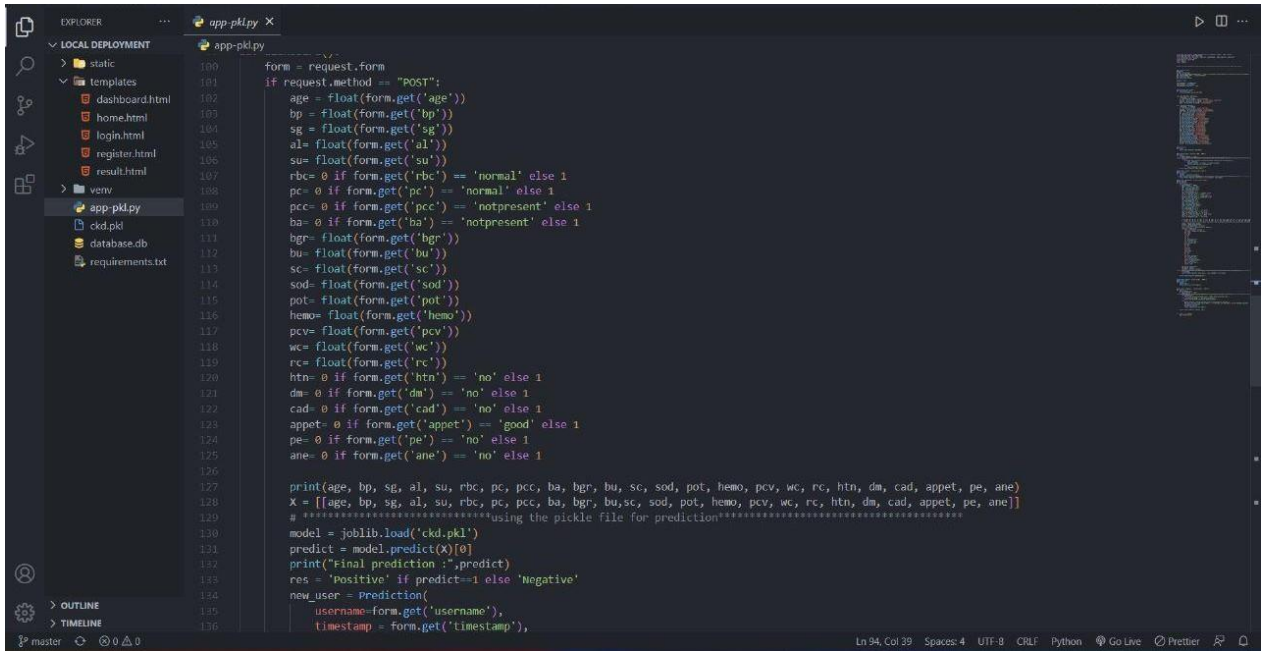

Test Results - CKD Predictor
127.0.0.1:5000/result
CKD Predictor
Hi! yash33
Dashboard
Result
Logout
Your Test Result is: Not Available Now!
Your Past Test Results
Date | Age | BP | SG | AL | SU | RBC | PC | PCC | BA | BGR | BU | SC | SOD | POT | HEMO | PCV | WC | RC | HTN | DM | CAD | APPET | PE | ANE | Result
2022-11-13T19:53 | 41.0 | 80.0 | 1.02 | 0.0 | 0.0 | normal | normal | notpresent | notpresent | 122.0 | 25.0 | 0.8 | 138.0 | 5.0 | 17.1 | 41.0 | 9100.0 | 5.2 | no | no | no | good | no | no | Negative
CKD© Predictor with

Flask Integration and Deployment

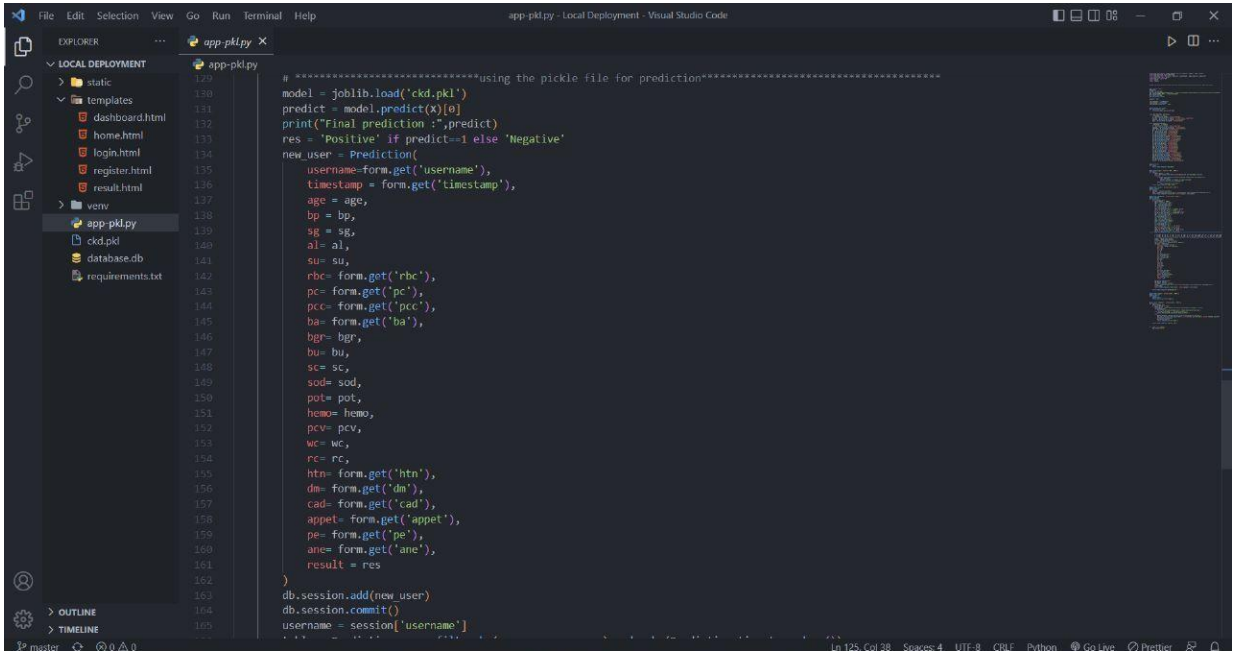



EXPLORER
app-pkl.py
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
class User(db.Model, UserMixin):
__tablename__ = 'users'
id = db.Column(db.Integer, primary_key=True)
username = db.Column(db.String(20), 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

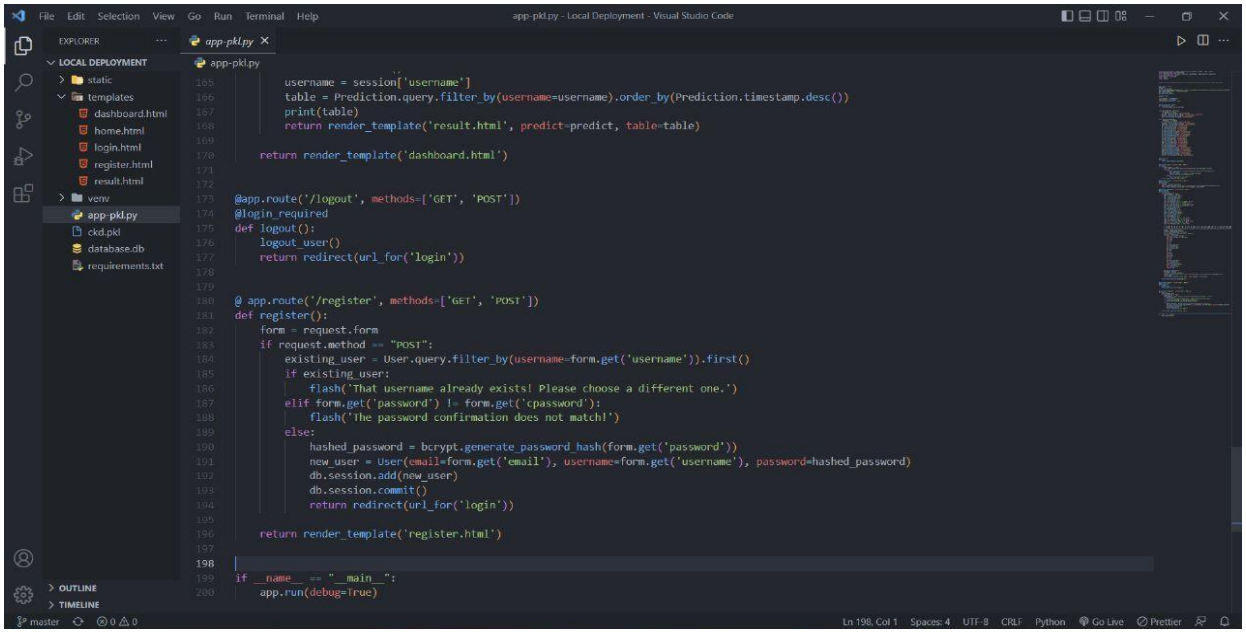

File Edit Selection View Go Run Terminal Help
app-pkl.py - Local Deployment - Visual Studio Code
EXPLORER
app-pkl.py
LOCAL DEPLOYMENT
static
templates
dashboard.html
home.html
login.html
register.html
result.html
venv
app-pkl.py
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
appet= db.Column(db.String(20), 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



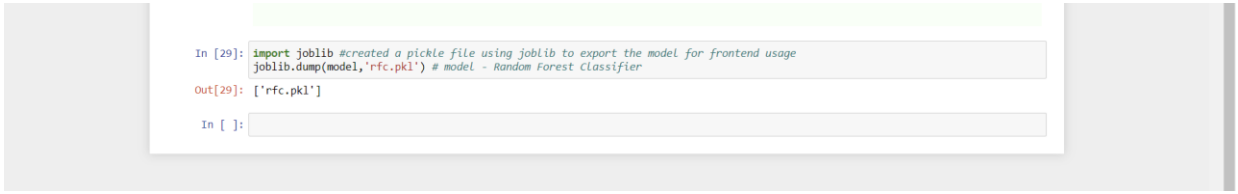
```
    form = request.form
    if request.method == "POST":
        age = float(form.get('age'))
        bp = float(form.get('bp'))
        sg = float(form.get('sg'))
        al= float(form.get('al'))
        su= float(form.get('su'))
        rbc= 0 if form.get('rbc') == 'normal' else 1
        pc= 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'),
```



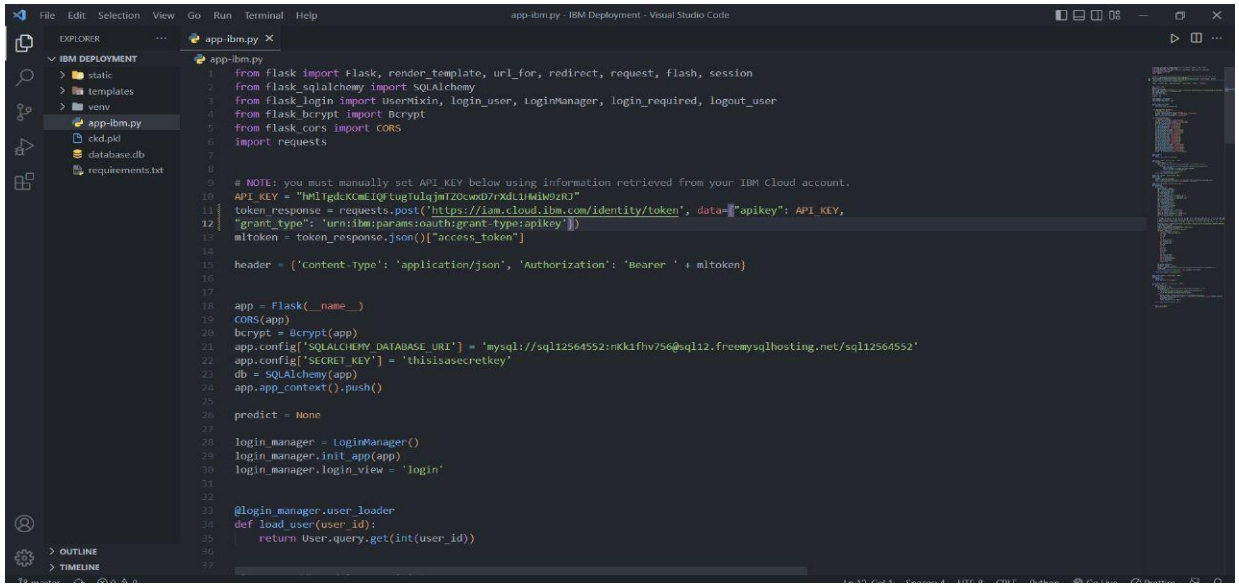
```
        # ****************************using the pickle file for prediction*************************************
        model = joblib.load('ckd.pkl')
        predict = model.predict(X)[0]
        print("Final prediction :",predict)
        res = 'Positive' if predict==1 else 'Negative'
        new_user = 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']
```

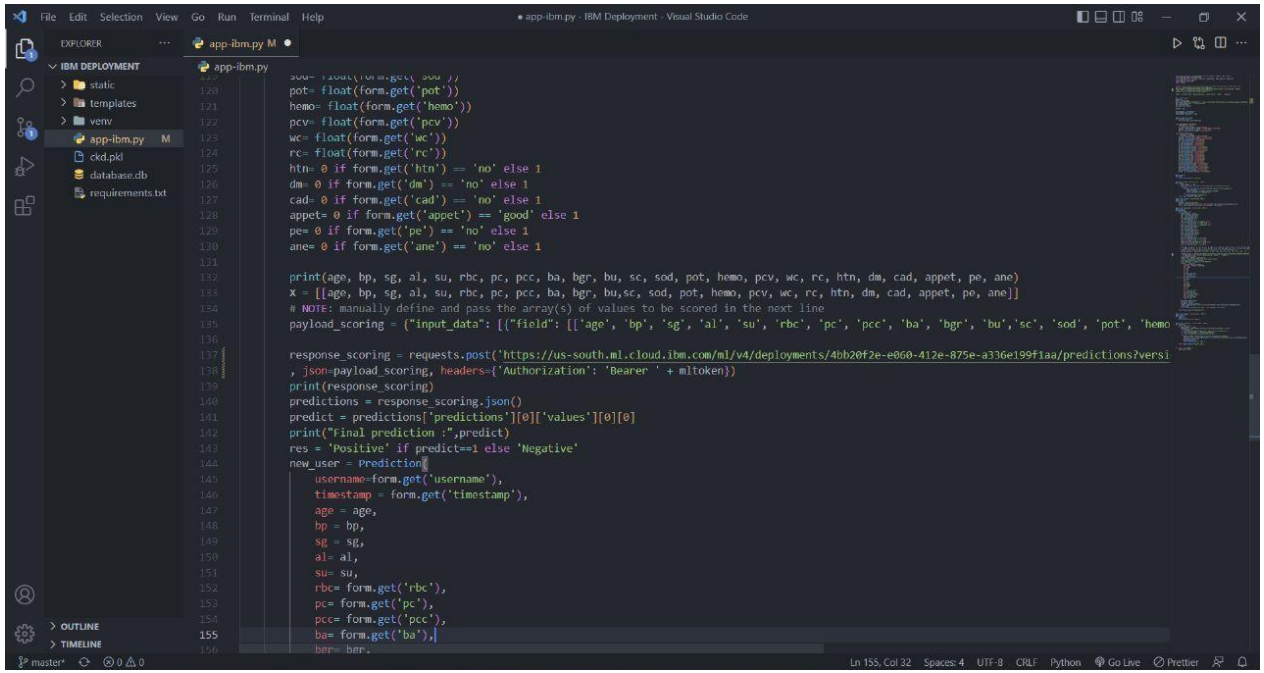


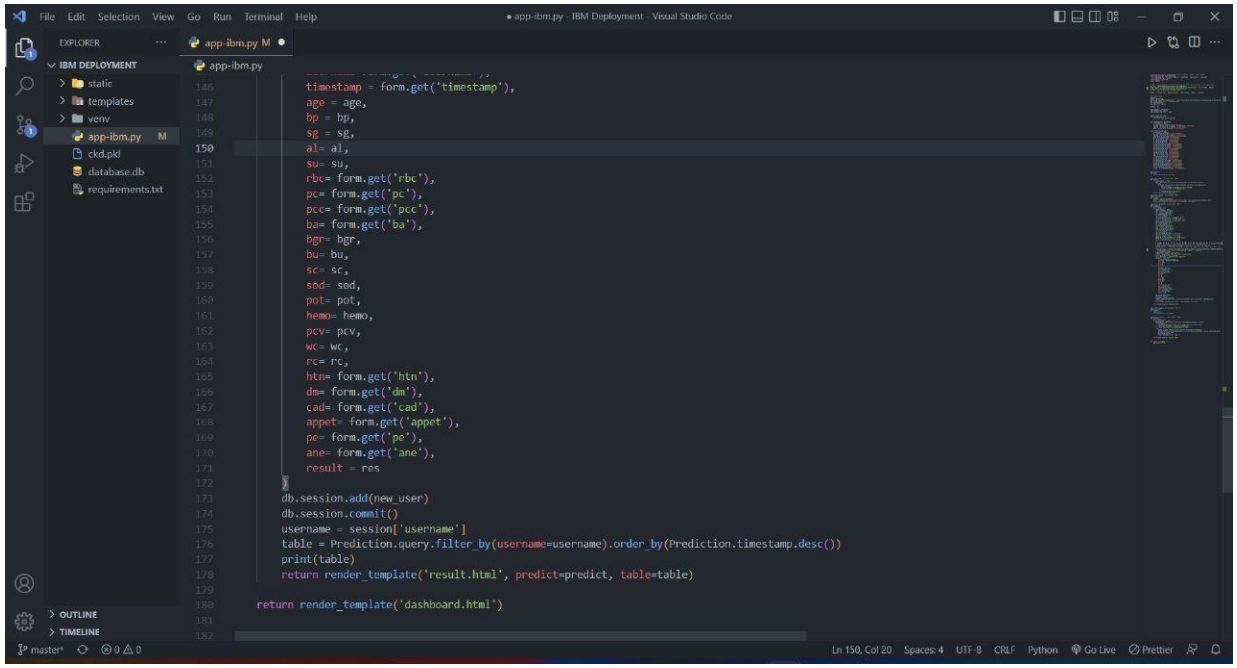
Using pickle to integrate with flask



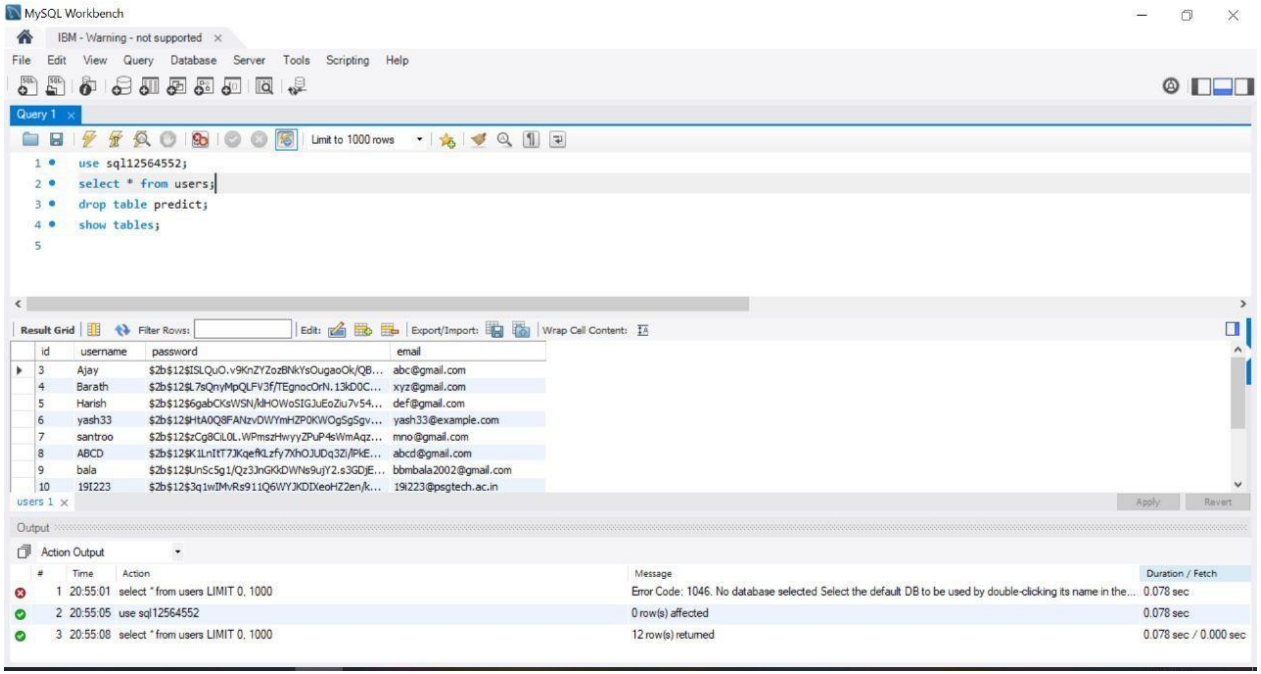
Flask changes for ibm deployment

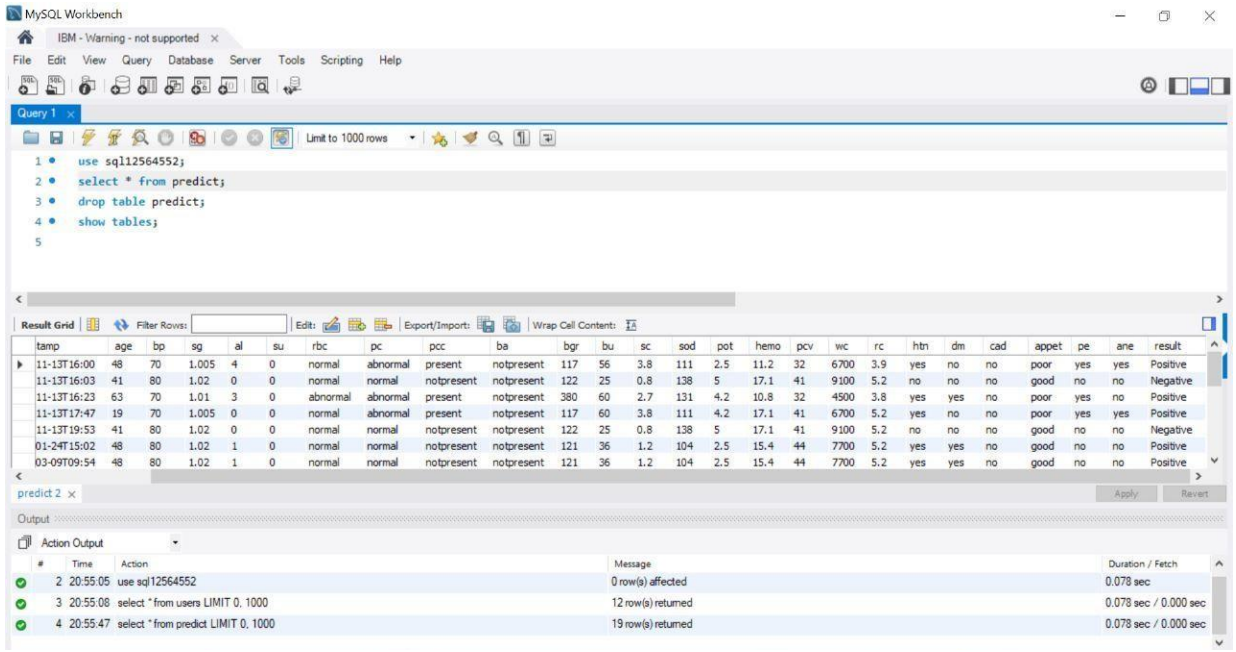



File Edit Selection View Go Run Terminal Help
app-ibm.py - IBM Deployment - Visual Studio Code
EXPLORER
IBM DEPLOYMENT
static
templates
venv
app-ibm.py M
ckd.pkl
database.db
requirements.txt
OUTLINE
TIMELINE
app-ibm.py M
pot= float(form.get('pot'))
hemo= float(form.get('hemo'))
pcv= float(form.get('pcv'))
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'),
bgr= bgr,
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
OUTLINE
TIMELINE
app-ibm.py M
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')
master*
Ln 150, Col 20
Spaces: 4
UTF-8
CRLF
Python
Go Live
Prettier

Database (mysql)



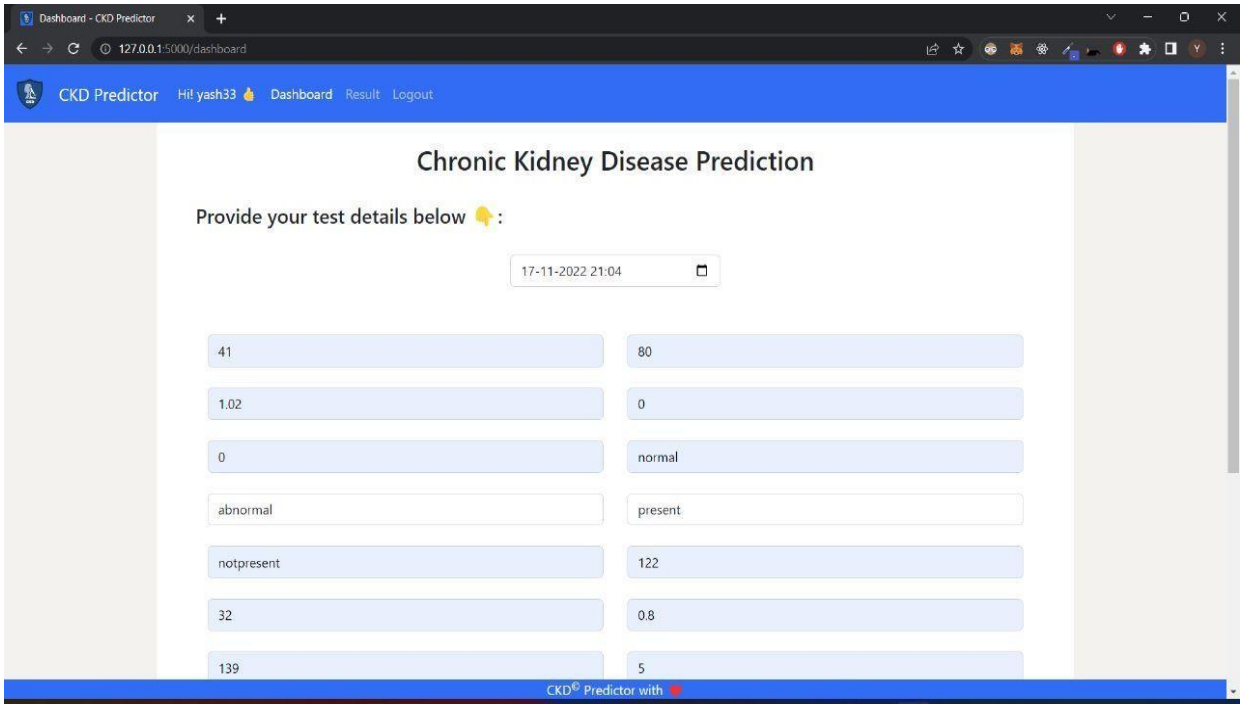

MySQL Workbench
IBM - Warning - not supported
File Edit View Query Database Server Tools Scripting Help
Query 1
Limit to 1000 rows
use sql12564552;
select * from predict;
drop table predict;
show tables;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
tamp age bp sg al su rbc pc pcc ba bgr bu sc sod pot hemo pcv wc rc htn dm cad appet pe ane result
11-13T16:00 48 70 1.005 4 0 normal abnormal present notpresent 117 56 3.8 111 2.5 11.2 32 6700 3.9 yes no no poor yes yes Positive
11-13T16:03 41 80 1.02 0 0 normal normal notpresent notpresent 122 25 0.8 138 5 17.1 41 9100 5.2 no no no good no no Negative
11-13T16:23 63 70 1.01 3 0 abnormal abnormal present notpresent 380 60 2.7 131 4.2 10.8 32 4500 3.8 yes yes no poor yes no Positive
11-13T17:47 19 70 1.005 0 0 normal abnormal present notpresent 117 60 3.8 111 4.2 17.1 41 6700 5.2 yes no no poor yes yes Positive
11-13T19:53 41 80 1.02 0 0 normal normal notpresent notpresent 122 25 0.8 138 5 17.1 41 9100 5.2 no no no good no no Negative
01-24T15:02 48 80 1.02 1 0 normal normal notpresent notpresent 121 36 1.2 104 2.5 15.4 44 7700 5.2 yes yes no good no no Positive
03-09T09:54 48 80 1.02 1 0 normal normal notpresent notpresent 121 36 1.2 104 2.5 15.4 44 7700 5.2 yes yes no good no no Positive
predict 2
Apply
Revert
Output
Action Output
Time Action Message Duration / Fetch
2 20:55:05 use sql12564552 0 row(s) affected 0.078 sec
3 20:55:08 select * from users LIMIT 0, 1000 12 row(s) returned 0.078 sec / 0.000 sec
4 20:55:47 select * from predict LIMIT 0, 1000 19 row(s) returned 0.078 sec / 0.000 sec

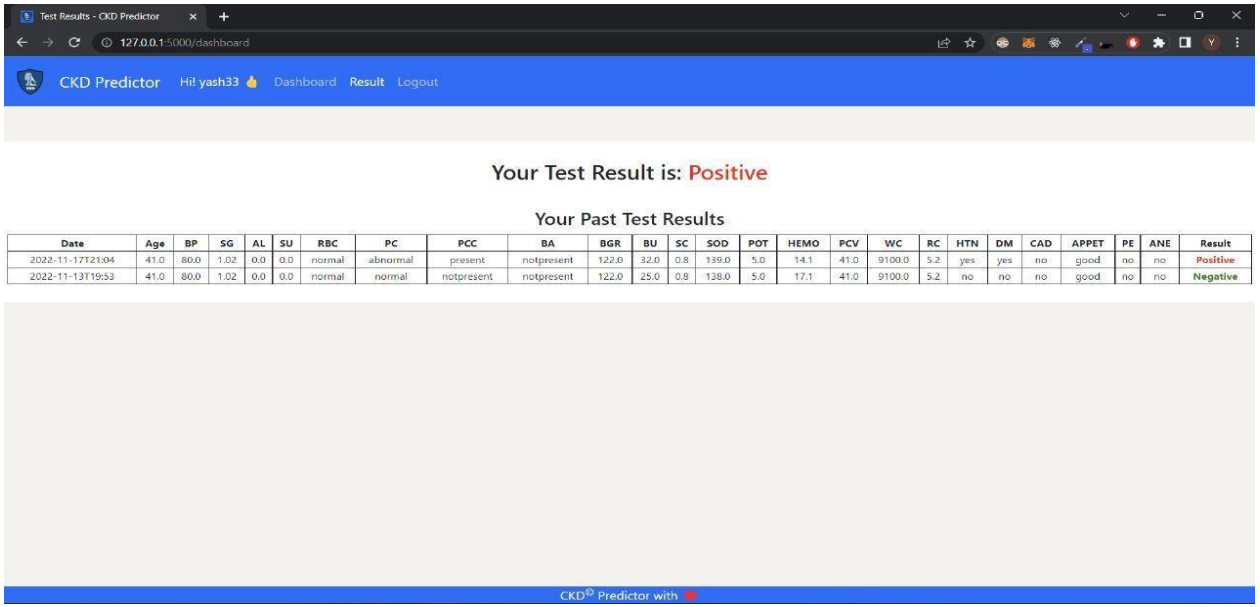
8. TESTING

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:




Test Results - CKD Predictor
127.0.0.1:5000/dashboard
CKD Predictor
Hi! yash33
Dashboard
Result
Logout
Your Test Result is: Positive
Your Past Test Results
Date Age BP SG AL SU RBC PC PCC BA BGR BU SC SOD POT HEMO PCV WC RC HTN DM CAD APPET PE ANE Result
2022-11-17T21:04 41.0 80.0 1.02 0.0 0.0 normal abnormal present notpresent 122.0 32.0 0.8 139.0 5.0 14.1 41.0 9100.0 5.2 yes yes no good no no Positive
2022-11-13T19:53 41.0 80.0 1.02 0.0 0.0 normal normal notpresent notpresent 122.0 25.0 0.8 138.0 5.0 17.1 41.0 9100.0 5.2 no no no good no no Negative
CKD© Predictor with

8.2 User Acceptance Testing

1. Purpose of Document

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

2. Defect Analysis

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

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

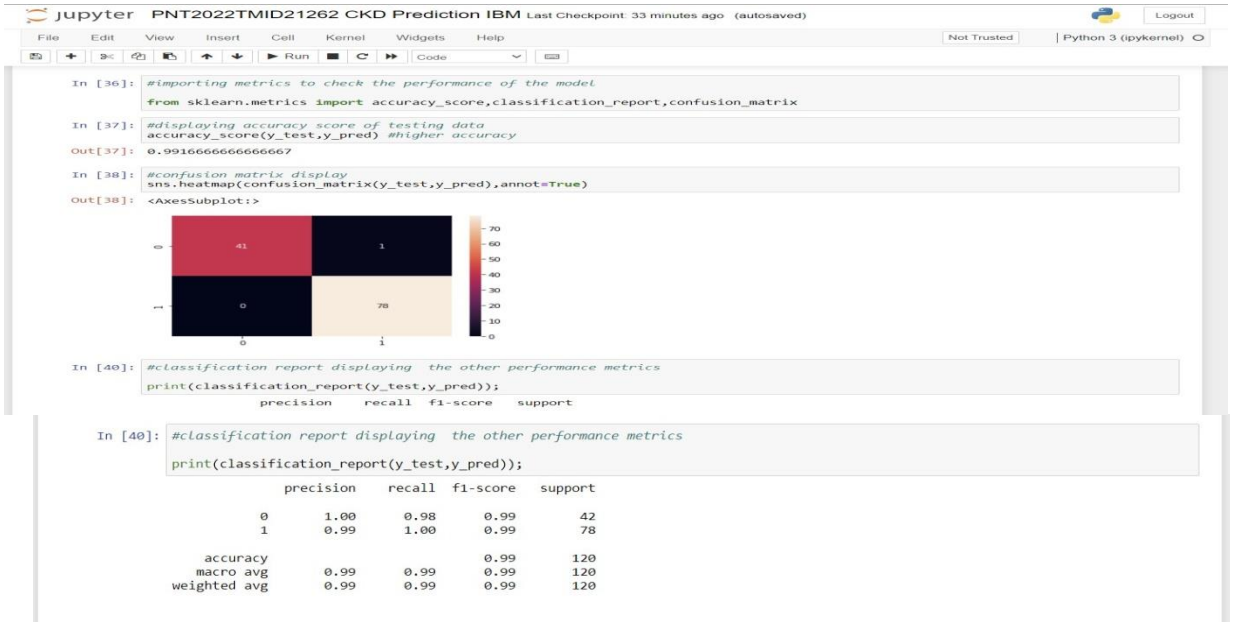
3. Test Case Analysis

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

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

9. RESULTS

4.9.1 Performance Metrics (Random Forest Classifier)



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 renaldisease 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 alot of advantages but it comes with certain disadvantages here are some ofthem .

- Chances of prediction to be wrong for least number of timewhich can cause problems
- Vast feature in dataset on discovery of time for the diseasemaking the model inefficient to keep up the metrics
- Since its a web application it requires scaling of web applicationto 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 learningmodel 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,andblood 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 betime 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 existingdataset 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-25744-1659972038

OUR REPOISOTORIES - https://github.com/Savundhariya7?tab=repositories