PROJECT REPORT DOCUMENT

CONTEXT

TEAM ID:PNT2022TMID45417

1.INTRODUCTION

1.1 PROJECT OVERVIEW

The chronic kidney disease popularly called as (CKD). The main causes of CKD are blood pressure, blood sugar and glucose level, strain on small blood vessels, excess sedimentation of salts on the kidney and also due to age factors. The end stage pf CKD leads to kidney failure that makes kidney unable to remove wastes from the body. And also fails to control the fluid levels in the body. It shows no symptoms at the earlier stages and results in high risks. Using, some machine learning models we can predict the CKD as early as possible. Data science using Machine learning is the most efficient to predict the CKD and prevent the patients from the risks.

1.2 PURPOSE

The experimental procedure concludes that advances in machine learning, with assist of predictive analytics, represent a promising setting by which to recognize intelligent solutions, which in turn prove the ability of predication in the kidney disease domain and beyond.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

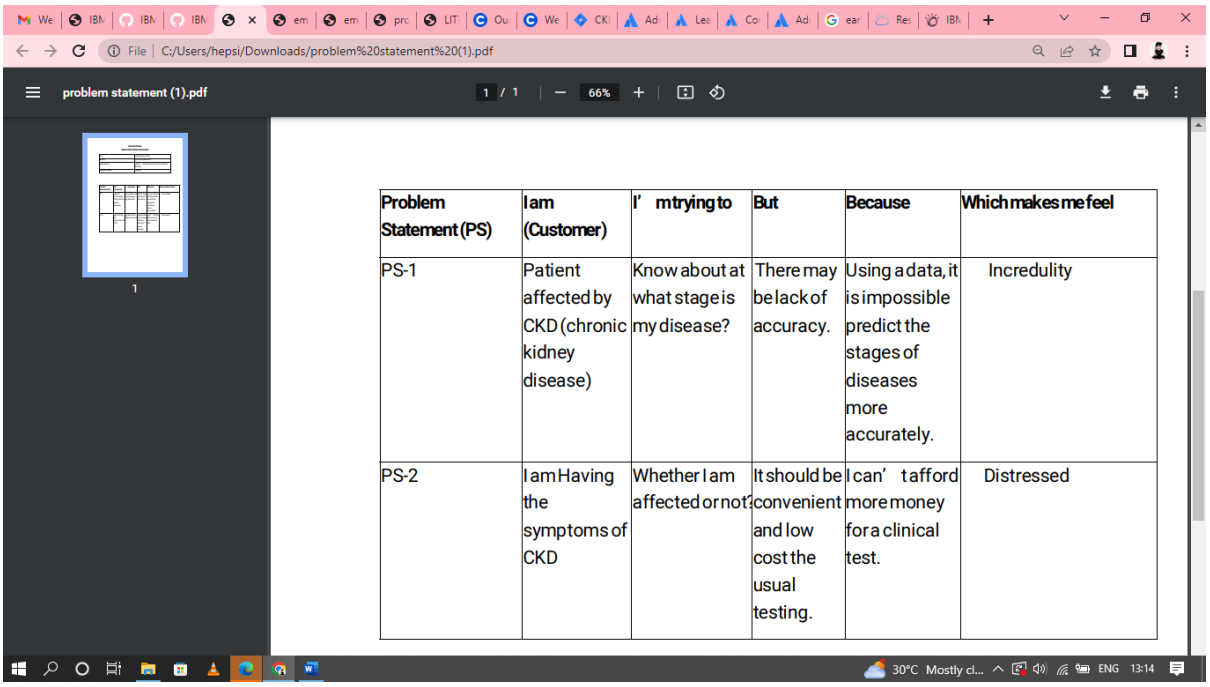
Shanila Yunus Yashfi, Md Ashikul Islam, Pritilata, Nazmus Sakib, Tanzila Islam, Mohammad Shahbaaz, Sadaf Salman Pantho [1] - “They have used UCI dataset and real time dataset and processed them. handled missing data, trained it and made a Random Forest and ANN model. Also implemented these two algorithms in python language. The gain using Random Forest algorithm is 97.12% and ANN is 94.5% respectively which is relatively very good” Siddheshwar Tekale , Pranjal Shingavi , Sukanya Wandhekar, Ankit Chatorikar[2] - “ have analyzed 14 different attributes related to CKD patients and predicted accuracy for different machine learning algorithms like Decision tree and Support Vector Machine. From the results analysis, it is observed that the decision tree algorithms give the accuracy of 91.75% and SVM gives accuracy of 96.75%.” PANKAJ CHITTORA 1, SANDEEP CHAURASIA1 , (Senior Member, IEEE), PRASUN CHAKRABARTI2,3, (Senior Member, IEEE), GAURAV KUMAWAT1 , TULIKA CHAKRABARTI4 , ZBIGNIEW LEONOWICZ 5 , (Senior Member, IEEE), MICHAŁ JASIŃSKI 5 , (Member, IEEE), ŁUKASZ JASIŃSKI 5 , RADOMIR GONO 6 , (Senior Member, IEEE), ELŻBIETA JASIŃSKA 7 , AND VADIM BOLSHEV 8 [3] – “three different techniques have been applied: correlation-based feature selection, Wrapper method and LASSO regression. In this perception, seven classifiers algorithm were applied viz. artificial neural

network, C5.0, logistic regression, CHAID, linear support vector machine (LSVM), K-Nearest neighbors and random tree. For each classifier, the results were computed based on full features, selected features by CFS, selected features by Wrapper, selected features by LASSO regression, SMOTE with selected features by LASSO, SMOTE with full features. It was observed that LSVM achieved the highest accuracy of 98.86% in SMOTE with full features." Saurabh Pal[4]-" Have used chronic kidney disease dataset collected from UCI machine leaning repository developed a chronic kidney disease prediction model using three machine learning classifiers Logistic Regression, Decision Tree and Support Vector Machine to measure the performance of the prediction model. After applying the base classifiers, we find decision tree classifier obtained better results in terms of Accuracy, Precision, Recall, F-score as 95.92%, 0.99, 0.98, and 0.98, respectively. The highest accuracy The application of MLAs in kidney diseases may enhance the ability of clinicians to predict CKD and RF, thus improving diagnostic assistance and providing suitable therapeutic decisions. However, it is necessary to improve the development process of MLA tools.

2.2 REFERENCES

[1] S. Y. Yashfi et al., "Risk Prediction of Chronic Kidney Disease Using Machine Learning Algorithms," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2020, pp. 1-5, Doi: 10.1109/ICCCNT49239.2020.9225548. [2] International Journal of Advanced Research in Computer and Communication Engineering Vol. 7, Issue 10, October 2018 Copyright to IJARCCE DOI 10.17148/IJARCCE.2018.71021 92 "Prediction of Chronic Kidney Disease Using Machine Learning Algorithm" Siddheshwar Tekale1, Pranjal Shingavi2, Sukanya Wandhekar3, Ankit Chatorikar4 Vidya Pratishthan's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati. [3] P. Chittora et al., "Prediction of Chronic Kidney Disease - A Machine Learning Perspective," in IEEE Access, vol. 9, pp. 17312-17334, 2021, Doi: 10.1109/ACCESS.2021.3053763. [4] Pal, S. Chronic Kidney Disease Prediction Using Machine Learning Techniques. Biomedical Materials & Devices (2022). https://doi.org/10.1007/s44174-022-00027-y [5] Schena, F.P., Anelli, V.W., Abbrescia, D.I. et al. Prediction of chronic kidney disease and its progression by artificial intelligence algorithms. J Nephrol (2022). https://doi.org/10.1007/s40620-022-01302-3

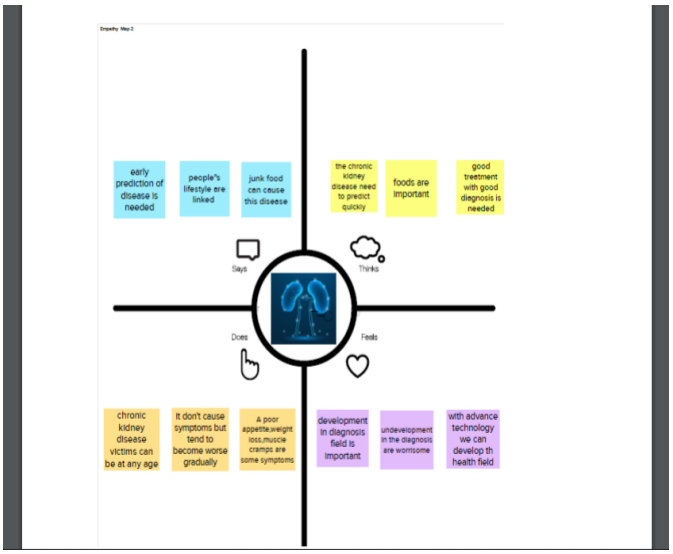
2.3 PROBLEM STATEMENT DEFINITION



Problem Statement (PS)	I am (Customer)	I' m trying to	But	Because	Which makes me feel
PS-1	Patient affected by CKD (chronic kidney disease)	Know about at what stage is my disease?	There may be lack of accuracy.	Using a data, it is impossible predict the stages of diseases more accurately.	Incredulity
PS-2	I am Having the symptoms of CKD	Whether I am affected or not?	It should be convenient and low cost the usual testing.	I can' t afford more money for a clinical test.	Distressed

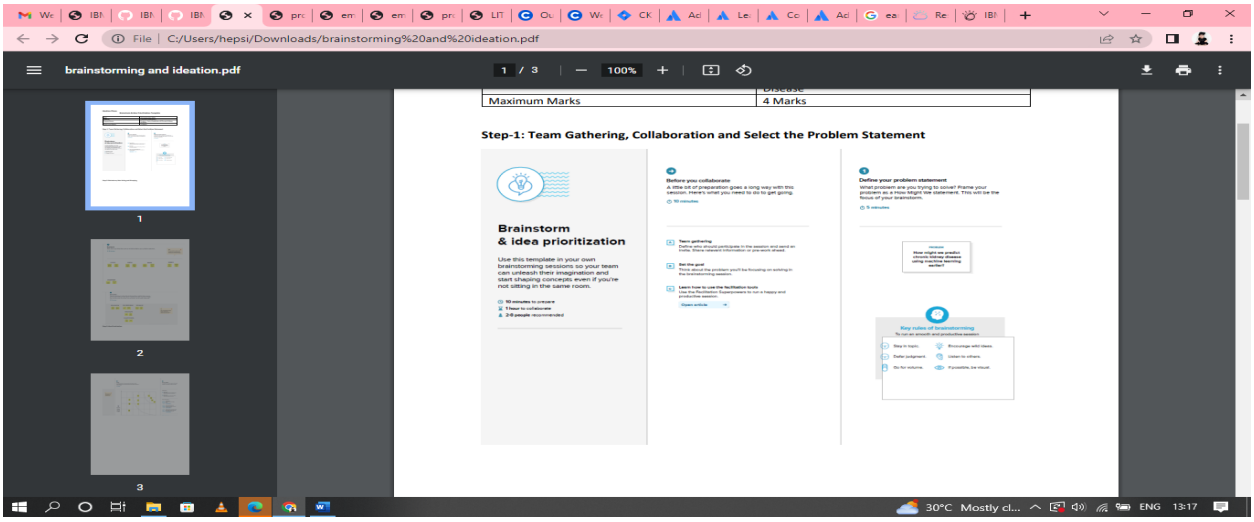
3. IDEATION & PROPOSED SOLUTION

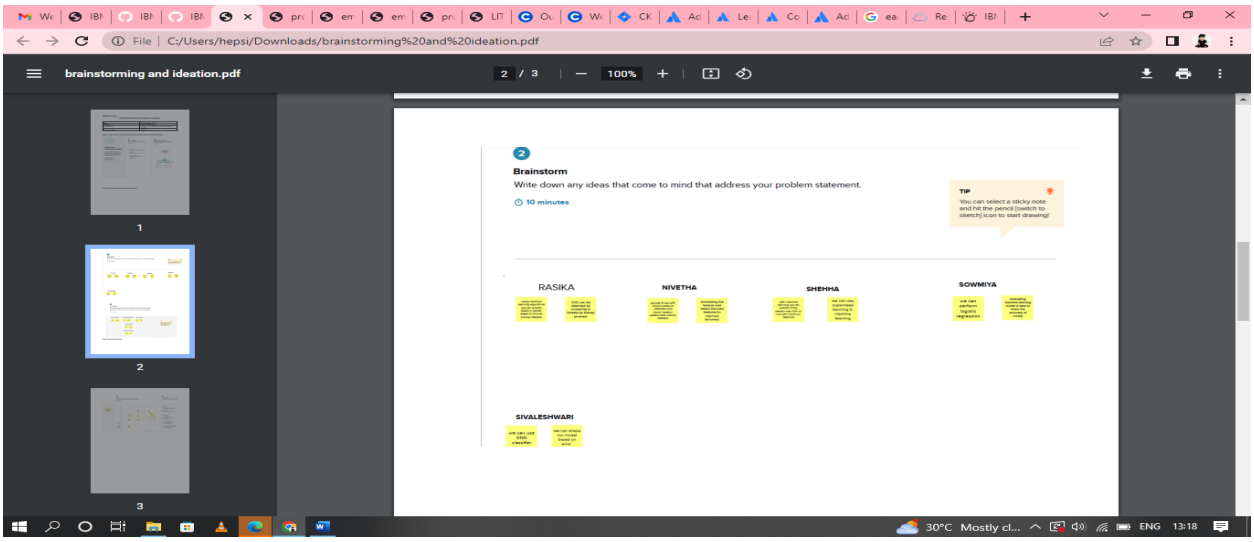
3.1 EMPATHY MAP CANVAS

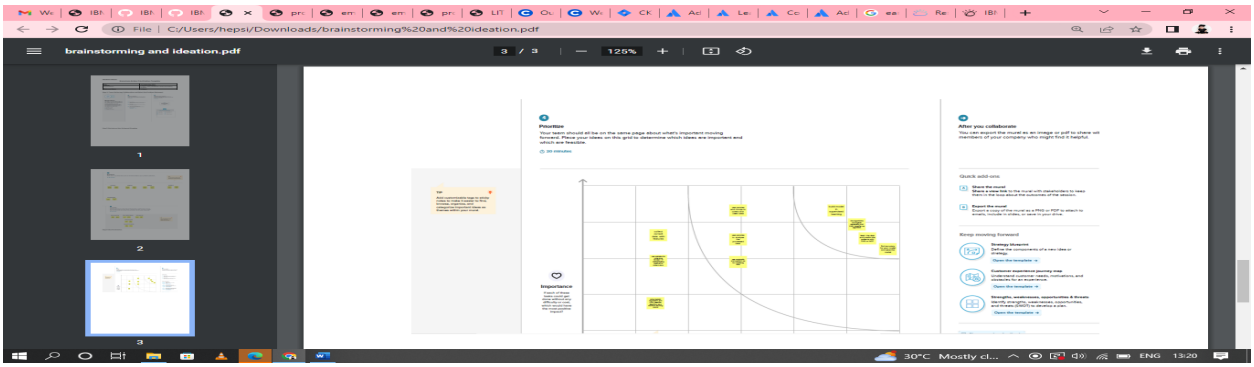


PAIN	GAIN
Chances of error or fault are Data requirement is more Inaccuracy of interpretation of data	It is automatic It is used in various fields It can handle varieties

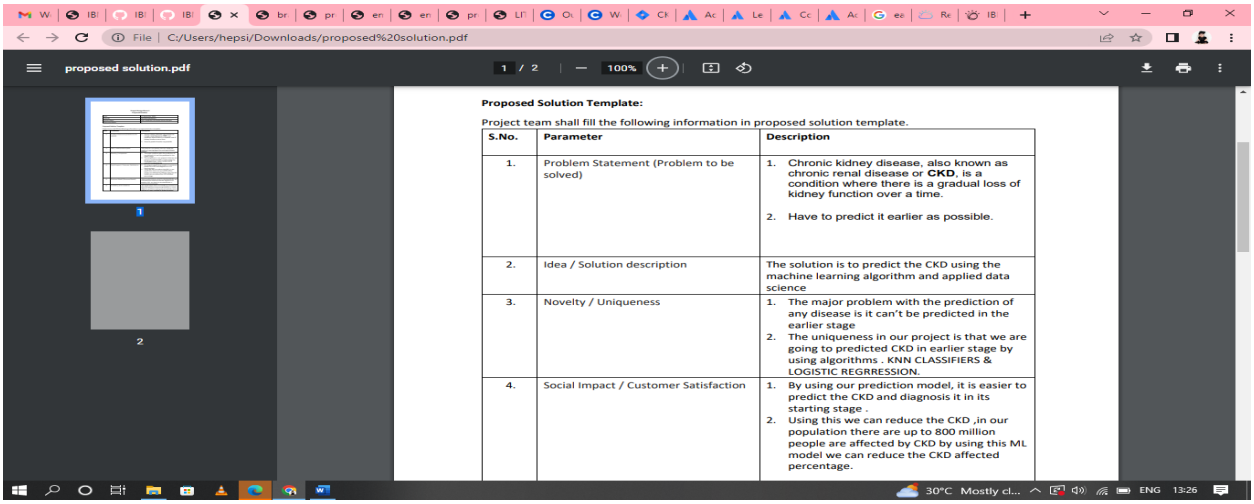
3.2 IDEATION & BRAINSTORMING







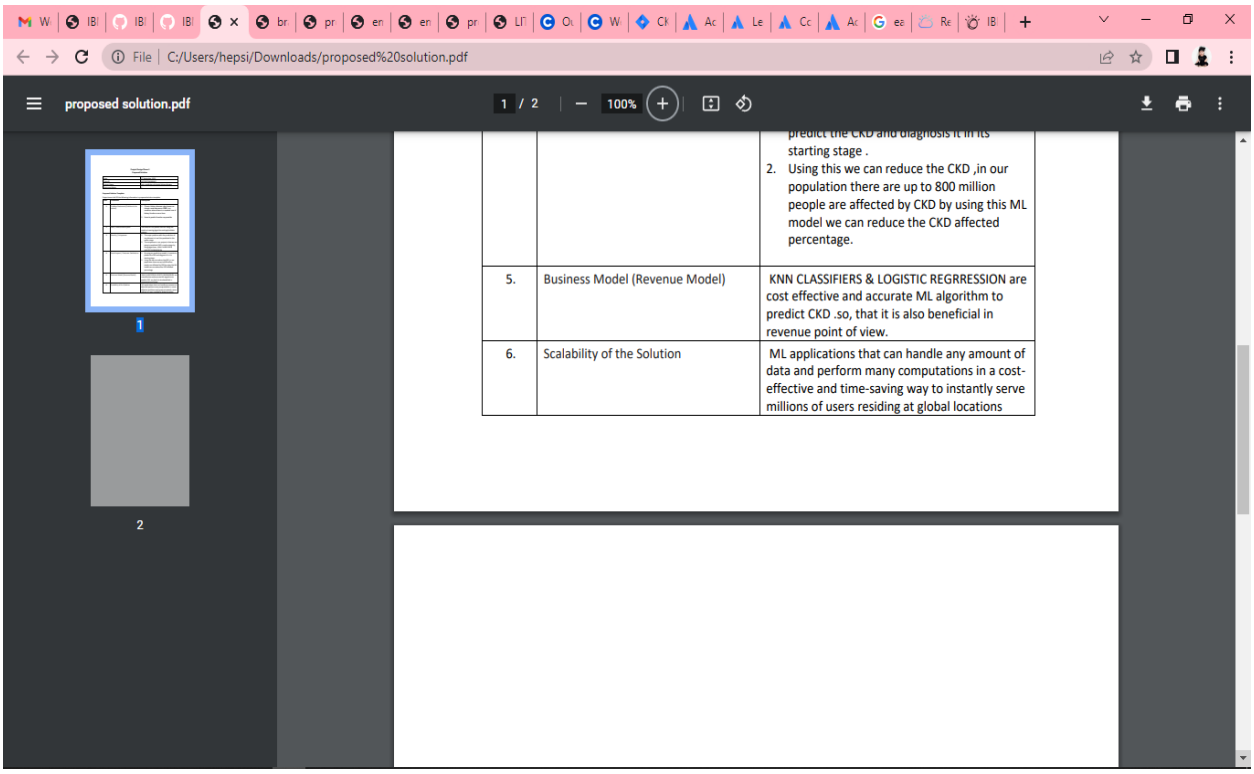
3.3 PROPOSED SOLUTION



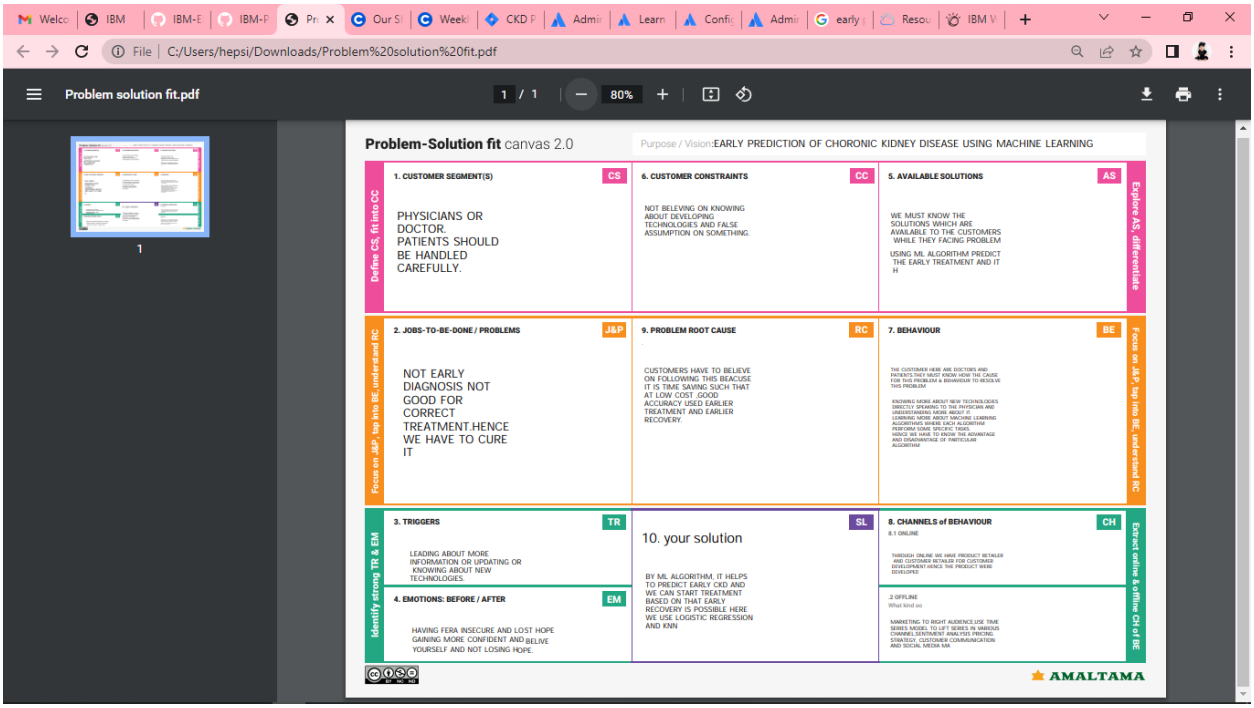
Proposed Solution Template:

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

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	1. Chronic kidney disease, also known as chronic renal disease or **CKD**, is a condition where there is a gradual loss of kidney function over a time. 2. Have to predict it earlier as possible.
2.	Idea / Solution description	The solution is to predict the CKD using the machine learning algorithm and applied data science
3.	Novelty / Uniqueness	1. The major problem with the prediction of any disease is it can't be predicted in the earlier stage 2. The uniqueness in our project is that we are going to predicted CKD in earlier stage by using algorithms . KNN CLASSIFIERS & LOGISTIC REGRESSION.
4.	Social Impact / Customer Satisfaction	1. By using our prediction model, it is easier to predict the CKD and diagnosis it in its starting stage . 2. Using this we can reduce the CKD ,in our population there are up to 800 million people are affected by CKD by using this ML model we can reduce the CKD affected percentage.

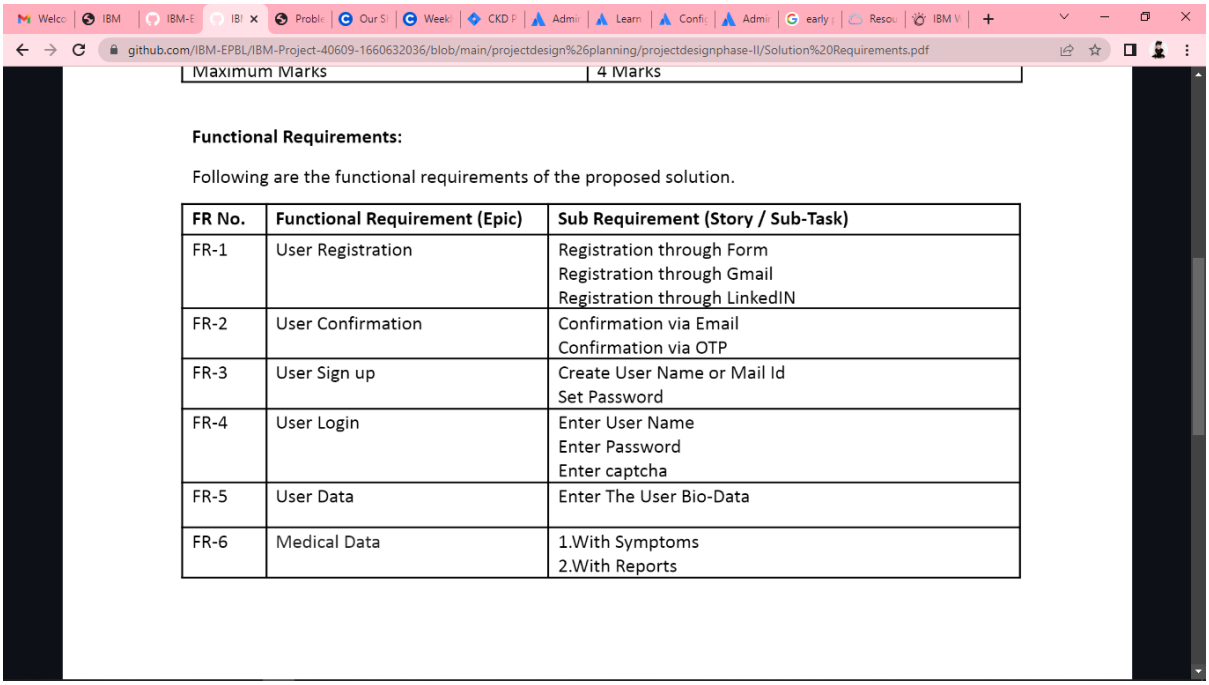


3.4 PROBLEM SOLUTION FIT



4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS



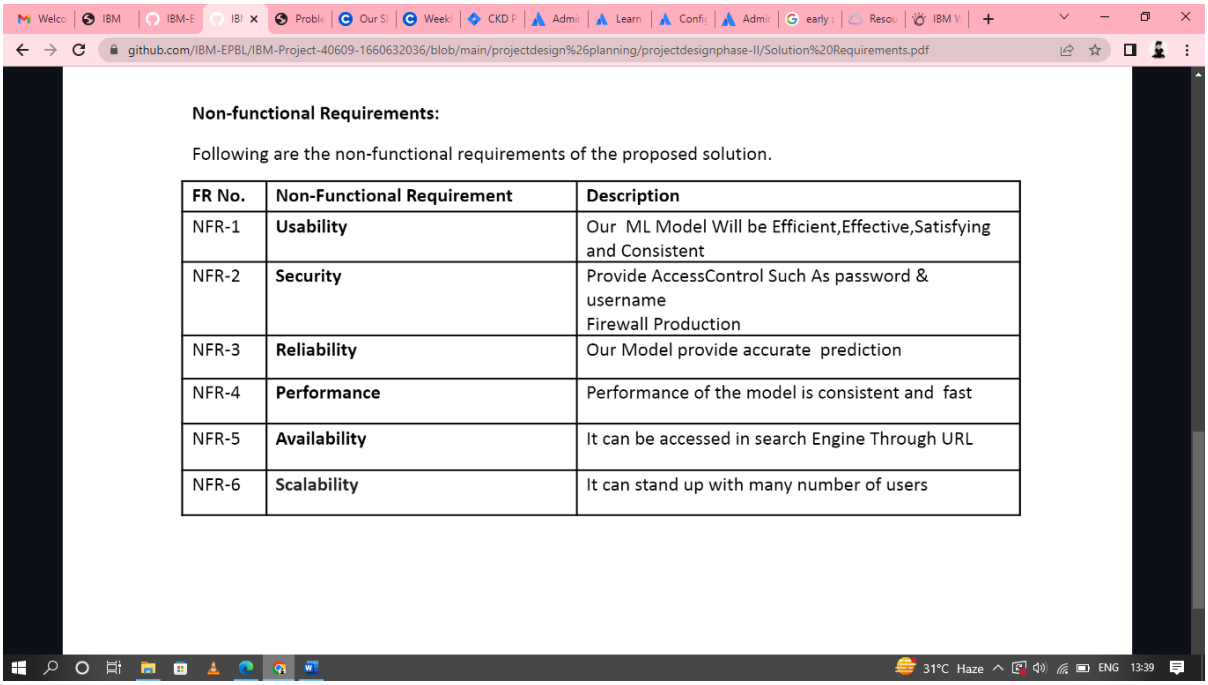
Maximum Marks	4 Marks

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 Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Sign up	Create User Name or Mail Id Set Password
FR-4	User Login	Enter User Name Enter Password Enter captcha
FR-5	User Data	Enter The User Bio-Data
FR-6	Medical Data	1.With Symptoms 2.With Reports

4.2 NON FUNCTIONAL REQUIREMENTS



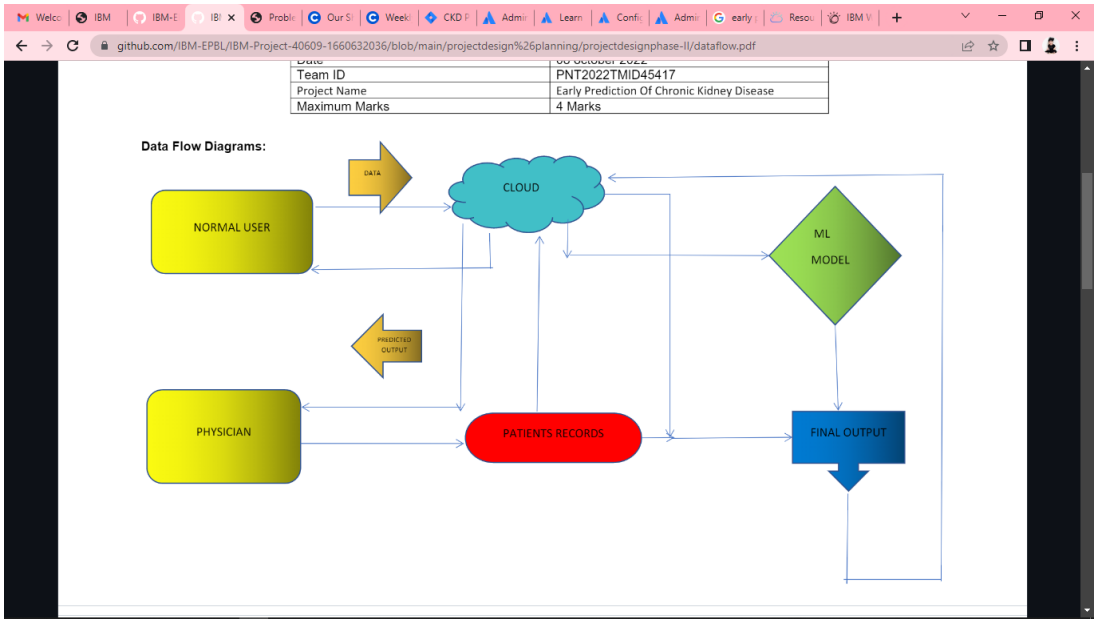
Non-functional Requirements:

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

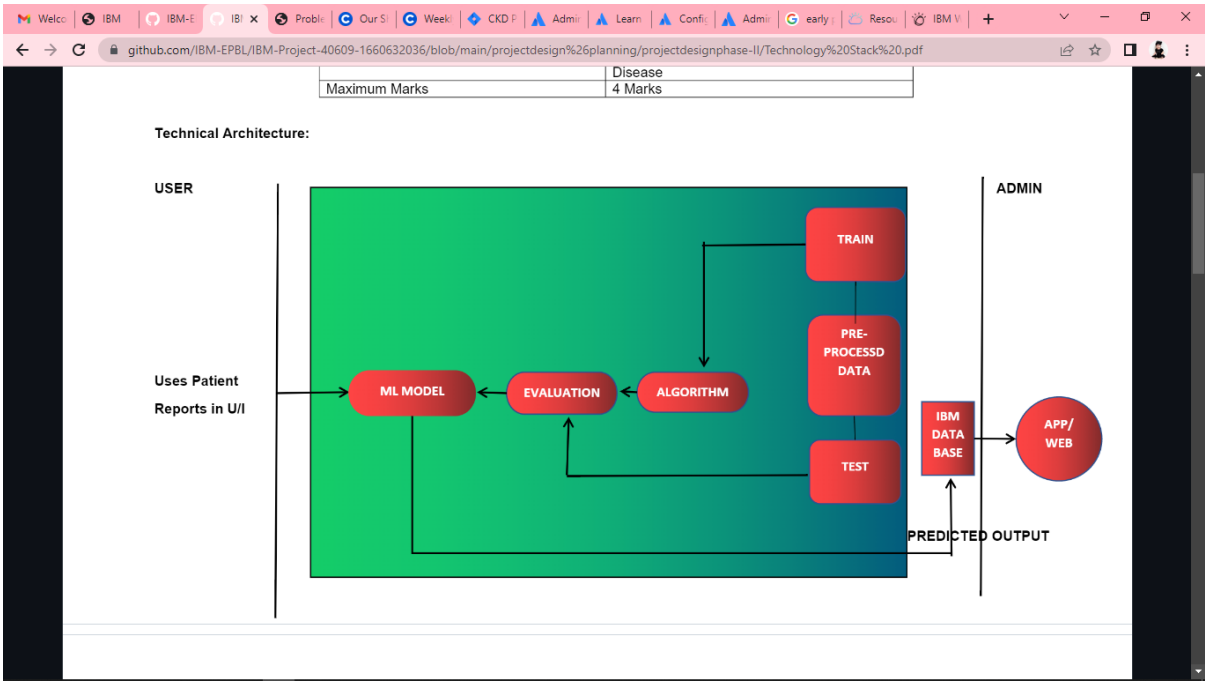
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Our ML Model Will be Efficient,Effective,Satisfying and Consistent
NFR-2	**Security**	Provide AccessControl Such As password & username Firewall Production
NFR-3	**Reliability**	Our Model provide accurate prediction
NFR-4	**Performance**	Performance of the model is consistent and fast
NFR-5	**Availability**	It can be accessed in search Engine Through URL
NFR-6	**Scalability**	It can stand up with many number of users

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE



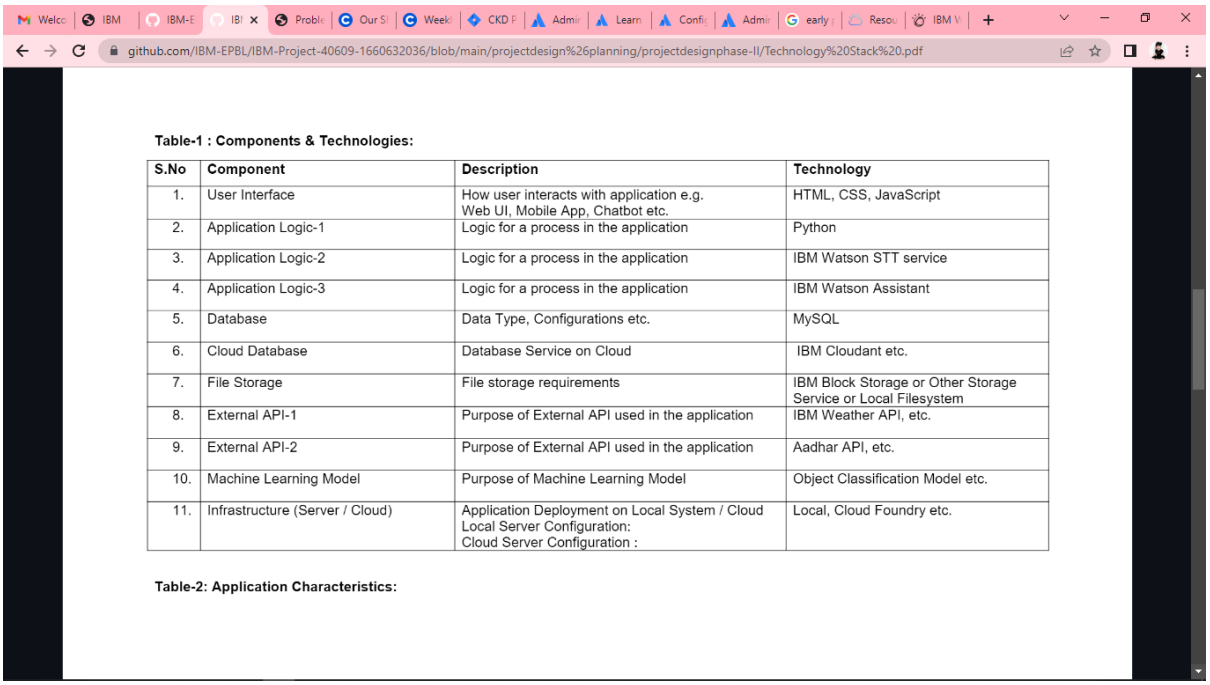


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Classification Model etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry etc.

Table-2: Application Characteristics:

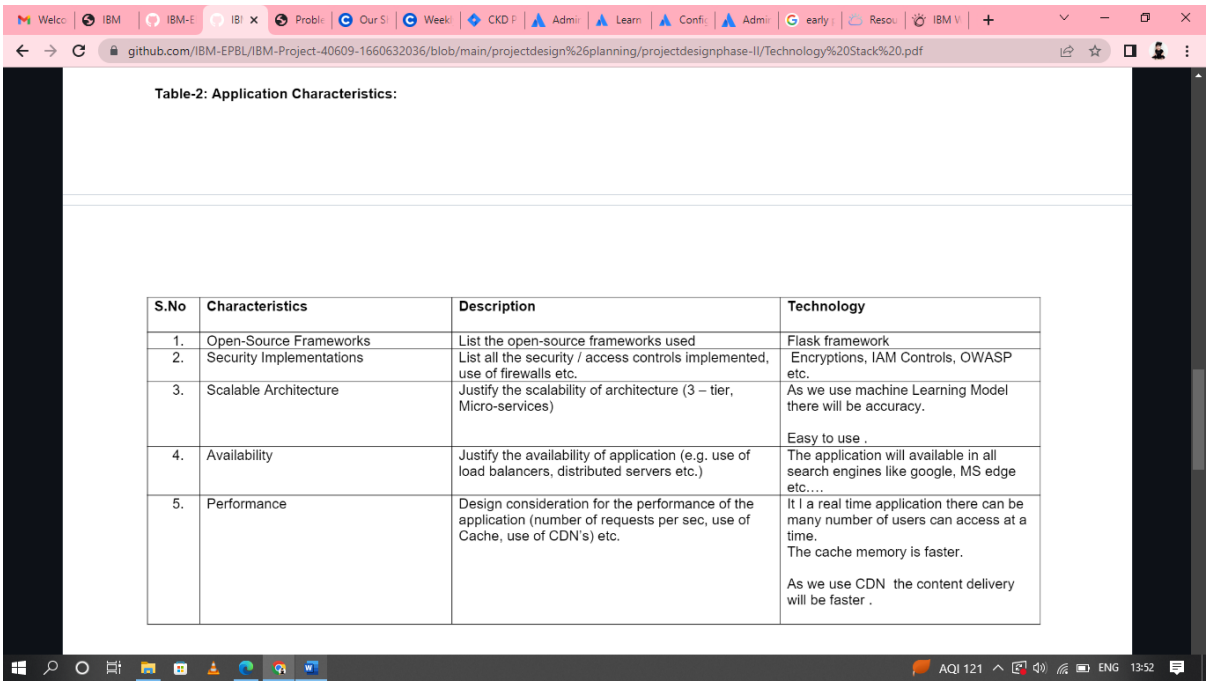
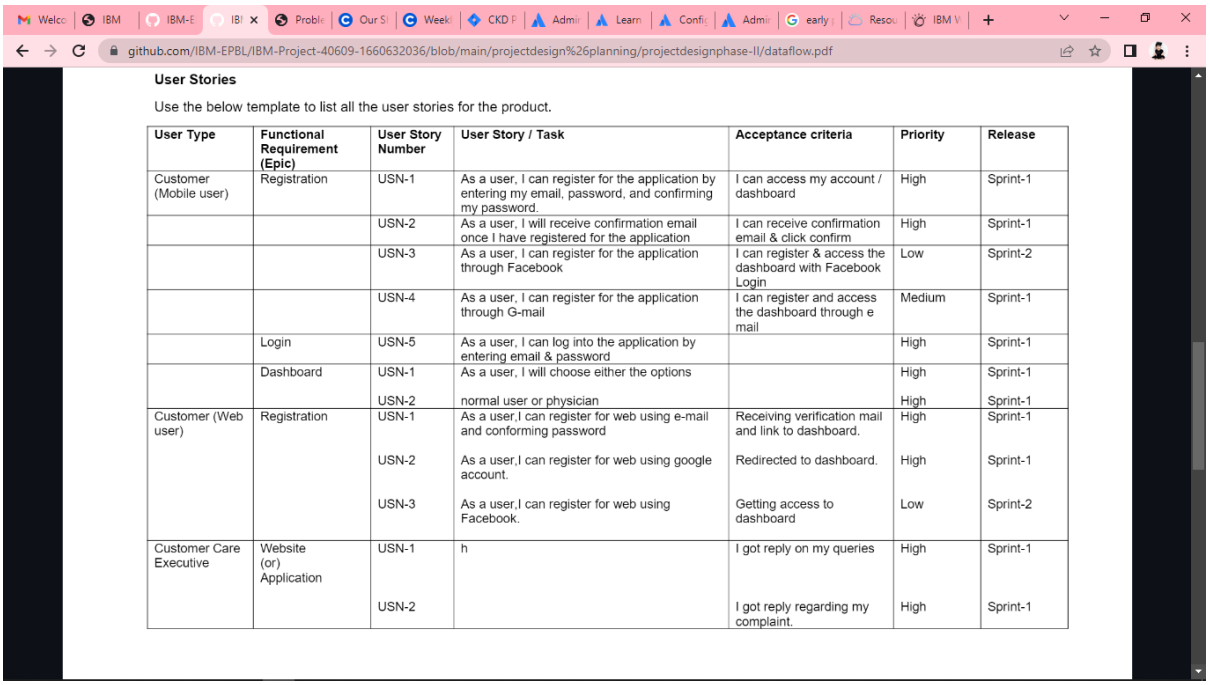


Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Flask framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	As we use machine Learning Model there will be accuracy. Easy to use .
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	The application will available in all search engines like google, MS edge etc....
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	It I a real time application there can be many number of users can access at a time. The cache memory is faster. As we use CDN the content delivery will be faster .

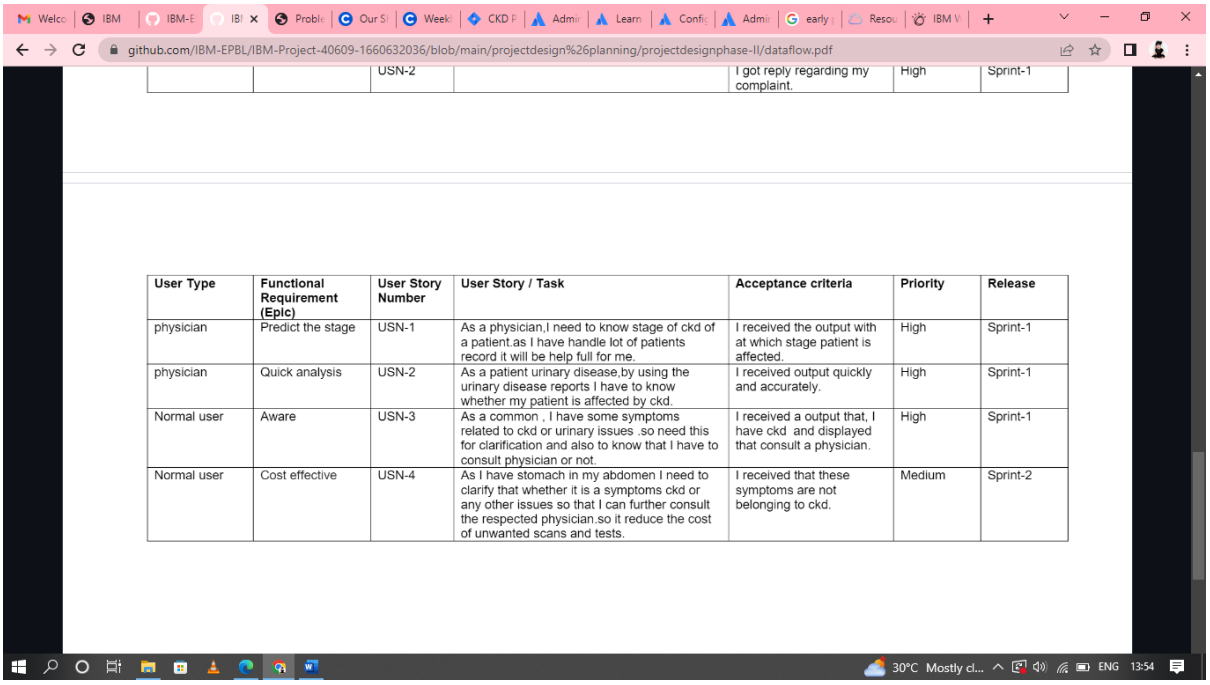
5.3 USER STORIES



User Stories

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

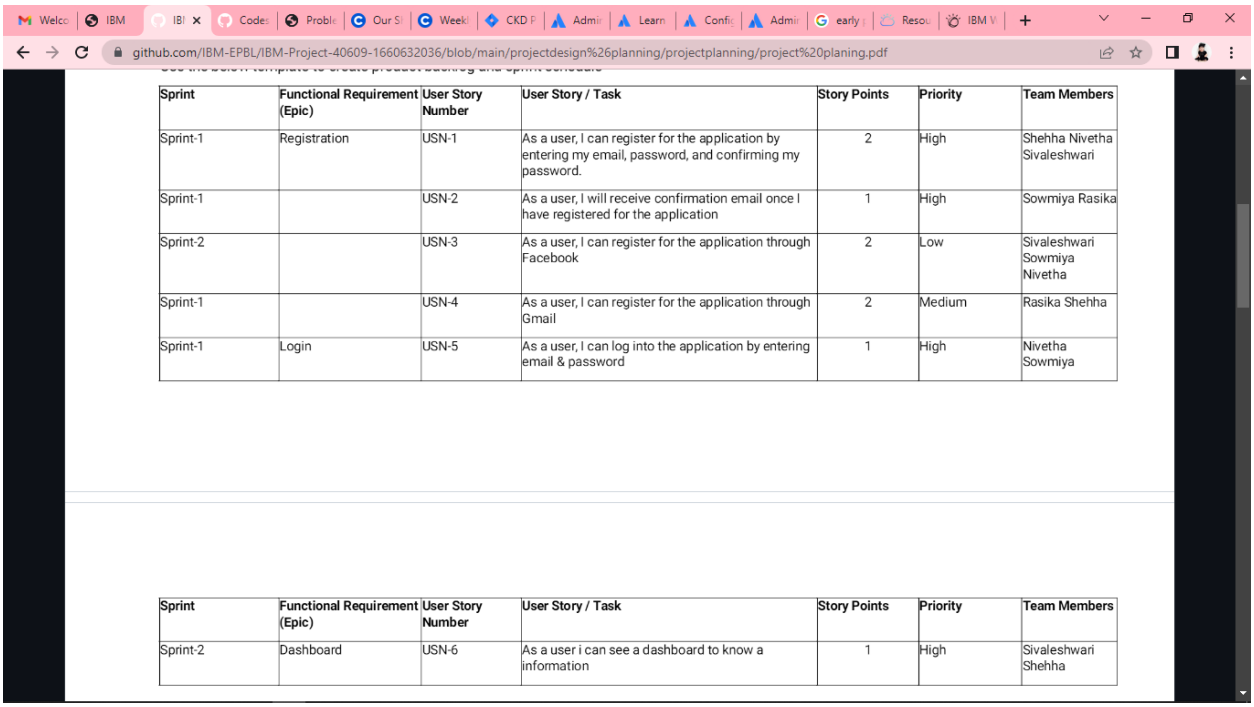
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		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
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through G-mail	I can register and access the dashboard through e mail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-1	As a user, I will choose either the options		High	Sprint-1
		USN-2	normal user or physician		High	Sprint-1
Customer (Web user)	Registration	USN-1	As a user,I can register for web using e-mail and conforming password	Receiving verification mail and link to dashboard.	High	Sprint-1
		USN-2	As a user,I can register for web using google account.	Redirected to dashboard.	High	Sprint-1
		USN-3	As a user,I can register for web using Facebook.	Getting access to dashboard	Low	Sprint-2
Customer Care Executive	Website (or) Application	USN-1	h	I got reply on my queries	High	Sprint-1
		USN-2		I got reply regarding my complaint.	High	Sprint-1



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
physician	Predict the stage	USN-1	As a physician,I need to know stage of ckd of a patient.as I have handle lot of patients record it will be help full for me.	I received the output with at which stage patient is affected.	High	Sprint-1
physician	Quick analysis	USN-2	As a patient urinary disease,by using the urinary disease reports I have to know whether my patient is affected by ckd.	I received output quickly and accurately.	High	Sprint-1
Normal user	Aware	USN-3	As a common , I have some symptoms related to ckd or urinary issues .so need this for clarification and also to know that I have to consult physician or not.	I received a output that, I have ckd and displayed that consult a physician.	High	Sprint-1
Normal user	Cost effective	USN-4	As I have stomach in my abdomen I need to clarify that whether it is a symptoms ckd or any other issues so that I can further consult the respected physician.so it reduce the cost of unwanted scans and tests.	I received that these symptoms are not belonging to ckd.	Medium	Sprint-2

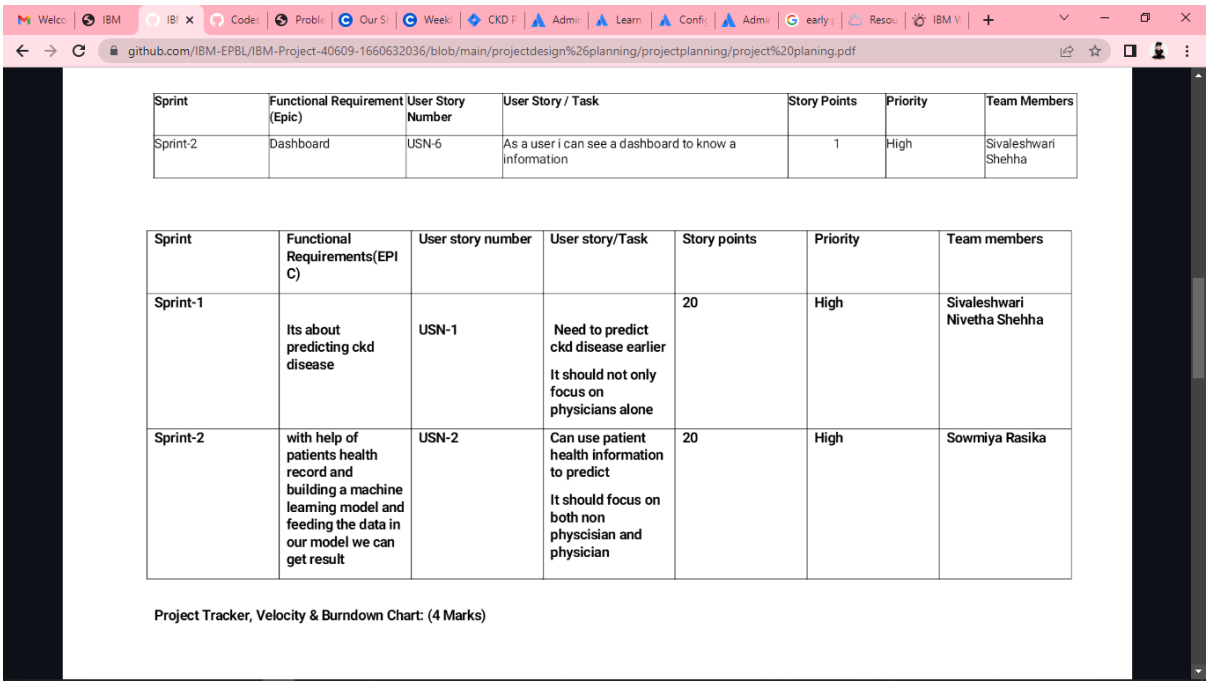
6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	Shehha Nivetha Sivaleshwari
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	Sowmiya Rasika
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	Sivaleshwari Sowmiya Nivetha
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	Rasika Shehha
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	Nivetha Sowmiya

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Dashboard	USN-6	As a user i can see a dashboard to know a information	1	High	Sivaleshwari Shehha

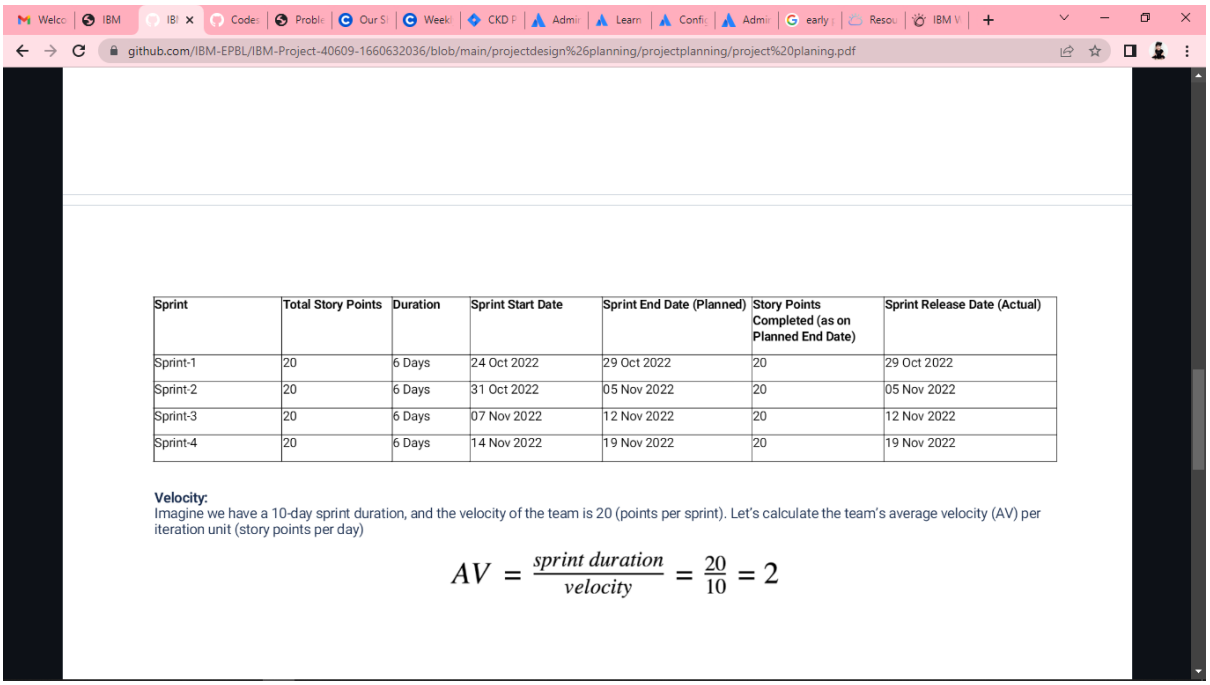


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	Dashboard	USN-6	As a user i can see a dashboard to know a information	1	High	Sivaleshwari Shehha

Sprint	Functional Requirements(EPIC)	User story number	User story/Task	Story points	Priority	Team members
Sprint-1	Its about predicting ckd disease	USN-1	Need to predict ckd disease earlier It should not only focus on physicians alone	20	High	Sivaleshwari Nivetha Shehha
Sprint-2	with help of patients health record and building a machine learning model and feeding the data in our model we can get result	USN-2	Can use patient health information to predict It should focus on both non physcisian and physician	20	High	Sowmiya Rasika

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

6.2 SPRINT DELIVERY SCHEDULE

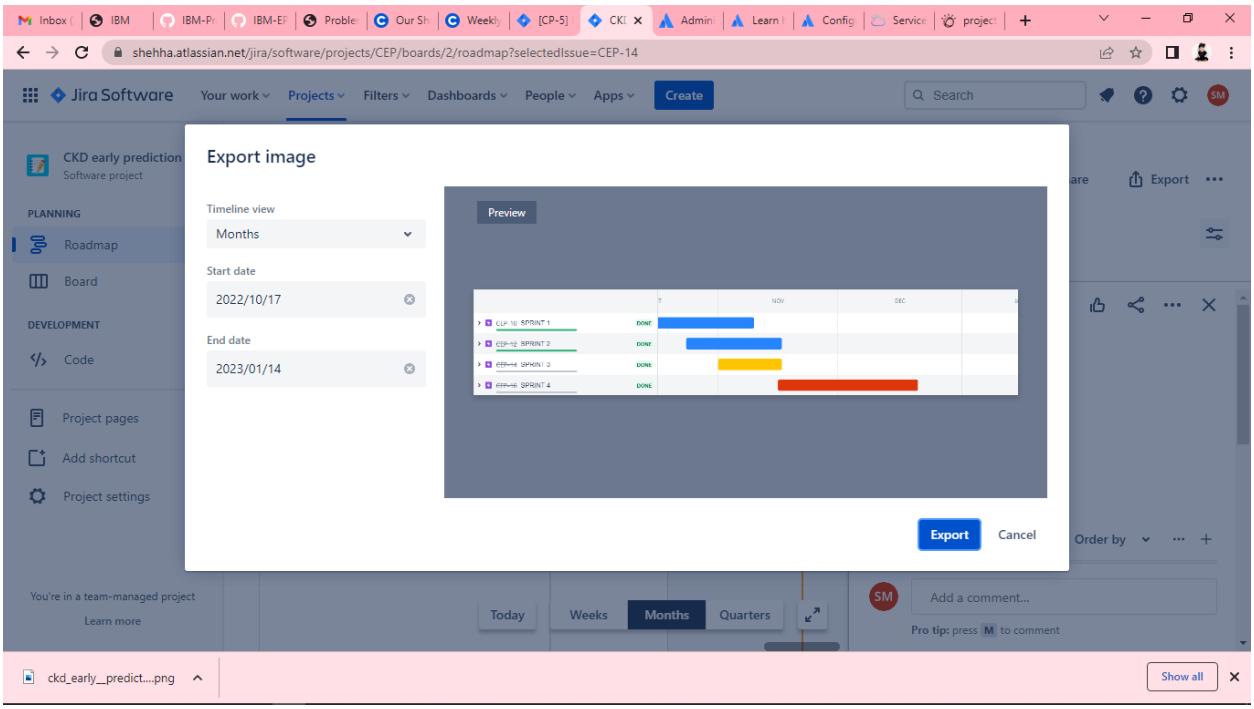


Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

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

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 REPORTS FROM JIRA



7. CODING & SOLUTIONS

7.1 FEATURE 1

#importing libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import os
os.getcwd()
path='C:\\Users\\ELCOT\\Downloads\\'
data=pd.read_csv(path+'chronickidneydisease.csv')
data.head(10)
data.tail(10)
data.shape
data.columns=['id','age','blood_pressure','specific_gravity','albumin','sugar','red_blood_cells',
        'pus_cell','pus_cell_clumps','bacteria','blood glucose random',
        'blood_urea','serum_creatinine','sodium','potassium','hemoglobin','packed_cell_volume',
        'white_blood_cell_count', 'red_blood_cell_count'
,'hypertension','diabetesmellitus','coronary_artery_disease',
        'apettite','pedal_edema','anemia','class']
data.columns
data.info()
data.drop(['id'],axis=1,inplace=True)
data
#target column
data['class'].unique()
#rectify target column
data['class']=data['class'].replace('ckd\t','ckd')
data['class'].unique()
#fetching categorical column
cat=data.select_dtypes(include=['object']).columns.tolist()
cat
#removing column which are not categorical
cat.remove('red_blood_cell_count')
cat.remove('packed_cell_volume')
cat.remove( 'white_blood_cell_count')
cat
num=data.select_dtypes(include=['float64']).columns.tolist()#fetch numerical column
num.remove('specific_gravity')#remove which are not numerical
num.remove('albumin')
num.remove('sugar')
num
#adding column which is numerical
num.append('red_blood_cell_count')
```

```
num.append('packed_cell_volume')
num.append( 'white_blood_cell_count')
num
sns.pairplot(data,hue='class')
fig=plt.figure(figsize=(10,5))
fig
sns.barplot(x='blood glucose random',y='class',data=data)
cat.append('specific_gravity')#adding column which is categorical
cat.append('albumin')
cat.append('sugar')
cat
a=data['coronary_artery_disease'].unique()
#b=data['sugar'].unique()
#c=data['albumin'].unique()
#d=data['specific_gravity'].unique()
#e=data['anemia'].unique()
#f=data['pedal_edema'].unique()#
#g=data['apettite'].unique()
h=data['diabetesmellitus'].unique()
#i=data['bacteria'].unique()
#j=data['hypertension'].unique()#
#k=data['red_blood_cell_count'].unique()
#l=data['pus_cell'].unique()
#m=data['pus_cell_clumps'].unique()
a,h
#rectifying the categorical column classes
data['coronary_artery_disease']=data['coronary_artery_disease'].replace('\tno','no')
data['coronary_artery_disease'].unique()
data['diabetesmellitus']=data.diabetesmellitus.replace(to_replace={'yes':'yes','\tyes':'yes','\tno':'no'
})
data['diabetesmellitus'].unique()
#handling missing value
data.isna().sum()
#before handling the numeric variable which is considered as string should be convert to
numerical
data.red_blood_cell_count=pd.to_numeric(data.red_blood_cell_count,errors='coerce')
data.packed_cell_volume=pd.to_numeric(data.packed_cell_volume,errors='coerce')
data.white_blood_cell_count=pd.to_numeric(data.white_blood_cell_count,errors='coerce')
#handle numerical column null values
data['blood_pressure'].fillna(data['blood_pressure'].mean(),inplace=True)
data['blood_urea'].fillna(data['blood_urea'].mean(),inplace=True)
```

```
data['blood glucose random'].fillna(data['blood glucose random'].mean(),inplace=True)
data['serum_creatinine'].fillna(data['serum_creatinine'].mean(),inplace=True)
data['sodium'].fillna(data['sodium'].mean(),inplace=True)
data['potassium'].fillna(data['potassium'].mean(),inplace=True)
data['hemoglobin'].fillna(data['hemoglobin'].mean(),inplace=True)
data['pus_cell'].fillna(data['pus_cell'].mode()[0],inplace=True)
data['age'].fillna(data['age'].mode()[0],inplace=True)
data['pus_cell_clumps'].fillna(data['pus_cell_clumps'].mode()[0],inplace=True)
data['bacteria'].fillna(data['bacteria'].mode()[0],inplace=True)
data['red_blood_cell_count'].fillna(data['red_blood_cell_count'].mode()[0],inplace=True)
data['red_blood_cells'].fillna(data['red_blood_cells'].mode()[0],inplace=True)
data['white_blood_cell_count'].fillna(data['white_blood_cell_count'].mode()[0],inplace=True)
data['packed_cell_volume'].fillna(data['packed_cell_volume'].mode()[0],inplace=True)
data['hypertension'].fillna(data['hypertension'].mode()[0],inplace=True)
data['diabetesmellitus'].fillna(data['diabetesmellitus'].mode()[0],inplace=True)
data['coronary_artery_disease'].fillna(data['coronary_artery_disease'].mode()[0],inplace=True)
data['apettite'].fillna(data['apettite'].mode()[0],inplace=True)
data['pedal_edema'].fillna(data['pedal_edema'].mode()[0],inplace=True)
data['anemia'].fillna(data['anemia'].mode()[0],inplace=True)
data['specific_gravity'].fillna(data['specific_gravity'].mode()[0],inplace=True)
data['albumin'].fillna(data['albumin'].mode()[0],inplace=True)
data['sugar'].fillna(data['sugar'].mode()[0],inplace=True)
data.isnull().sum()
from sklearn.preprocessing import LabelEncoder
for i in cat:
    print('label of encoder= ',i)
    lei=LabelEncoder()
    print(data[i])
    data[i]=lei.fit_transform(data[i])
    print(data[i])
    print('*'*100)
data.corr().T
selcols=['red_blood_cells','pus_cell',
 'diabetesmellitus', 'coronary_artery_disease','blood_urea','pedal_edema','anemia',
 'blood glucose random']
x=pd.DataFrame(data,columns=selcols)
y=pd.DataFrame(data,columns=['class'])
print(x.shape)
print(y.shape)
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)
```

```
print(x_train.shape)
print(y_train.shape)
print(x_test.shape)
print(y_test.shape)
from sklearn.linear_model import LogisticRegression
lgr=LogisticRegression()
lgr.fit(x_train,y_train)
y_pred=lgr.predict(x_test)
y_pred1=lgr.predict([[140,45,0,0,0,0,0,0]])
print(y_pred1)
from sklearn.metrics import accuracy_score
acc=accuracy_score(y_test,y_pred)
acc
from sklearn.metrics import confusion_matrix
conf_mat=confusion_matrix(y_test,y_pred)
conf_mat
import pickle
pickle.dump(lgr,open('CKD.pkl','wb'))
```

7.2 FEATURE 2

We import pandas for the analytics of the data set. Numpy library is used for the computation of array and seaborn for the visualization.We split the data in to train and test , and we logistic regression model to train the model and predict the ckd,by integrate the model in the web application developed in the flask.

8.TESTING

8.1 TEST CASES

We split the data in to train and test and train the model

```
from sklearn.model_selection import train_test_split

#SPLITING

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=2)

print(x_train.shape)

print(y_train.shape)
```

```
print(x_test.shape)

print(y_test.shape)

#IMPORT MODEL FOR TRAIN

from sklearn.linear_model import LogisticRegression

lgr=LogisticRegression()

lgr.fit(x_train,y_train)

#TEST

y_pred=lgr.predict(x_test)

y_pred1=lgr.predict([[140,45,0,0,0,0,0,0]])

print(y_pred1)
```

8.2 USER ACCEPTANCE TESTING

User Acceptance Testing (UAT), which is performed on most UIT projects, sometimes called beta testing or end-user testing, is a phase of software development in which the software is tested in the "real world" by the intended audience or business representative. This type of testing is not intended to be menu-driven, but rather to be performed by business users to verify that the application will meet the needs of the end-user, with scenarios and data representative of actual usage in the field.We make a web application to provide the better user interface.

9.RESULTS

9.1 PERFORMANCE METRICES

We get the performance accuracy of 91% for the prediction of our data set using logistic regression.

10. ADVANTAGES& DISADVANTAGES

ADVANTAGES:

- The website provide better user interface.
- It provide output with simple steps

- It is used to predict the disease quickly.
- It will be available for everyone.

DISADVANTAGES:

- Some times the prediction may be unreliable.
- When data is not accurate , the prediction will be wrong.

11. CONCLUSION:

Chronic Kidney Disease (CKD) is a serious condition that affects over 14% of the world's population. Because it can be predicted with a 91% overall accuracy, individuals may detect it early and receive treatment with the least amount of expense and risk. In practice, effective feature engineering minimizes the amount of characteristics required for the prediction algorithm and thus, the number of required medical tests. In comparison to similar work performed with the same dataset, filling in missing values based on their distribution and the co-occurrence of other characteristics using a Logistic Regression leads to improved accuracy in prediction models. Furthermore, compared to other models, the extratrees classifier and the random forest classifier are the superior algorithms for making predictions for CKD since they have 100% overall accuracy and less bias towards certain features. This project includes data preparation, treatment of missing information, and feature selection to predict the presence or absence of CKD. The necessity of adding domain expertise into feature selection while analyzing clinical data linked to CKD is further highlighted by this work. In light of this, it may be desirable to investigate the usage of a Logistic Regression based technique in the future to manage missing values in datasets relevant to various illnesses. Additionally, by including information about food kinds, water consumption habits, and genetics into the study, additional understanding of CKD may be acquired.

12. FUTURE SCOPE

The future scope of the project is very high.It will prevent the CKD in earlier and reduce the risk factors. As it can be used by normal people other than physician it give guidence to the people whether they to consult the respective physician or no need.

13.APPENDIX

SOURCE CODE:IBM

GITHUB: https://github.com/IBM-EPBL/IBM-Project-40609-1660632036

DEMO LINK