**CHAPTER 1**

**INTRODUCTION**

**1.1 PROJECT OVERVIEW**

Liver diseases avert the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease using classiﬁcation algorithms is an eﬃcacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of liver disease at an early stage is a complex task for the doctors. The main objective of this project is to analyze the parameters of various classiﬁcation algorithms and compare their predictive accuracies so as to ﬁnd out the best classiﬁer for determining the liver disease. This Project examines data from liver patients concentrating on relationships between a key list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease. Here we are building a model by applying various machine learning algorithms ﬁnd the best accurate model. And integrate to ﬂask based web application. User can predict the disease by entering parameters in the web application.

**1.2 PURPOSE**

Current screening strategies for liver disease focus on detection of subclinical advanced liver ﬁbrosis but cannot identify those at high future risk of severe liver disease. Our aim was to develop and validate a risk prediction model for incident liver disease in the general population based on widely available factors. The diagnosis is often delayed until severe complications occur and prognosis becomes poor. In order to identify individuals in the general population who have a high risk of developing severe liver disease in the future, we developed and validated a Liver Disease risk prediction with or without measurement of the liver enzyme gamma-glutamyltransferase. The Liver Disease score can be used as part of health counseling, and for planning further liver investigations and follow-up.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 EXISTING PROBLEMS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. NO | AUTHOR NAME | THEME | AREA OF ESTIMATION | ALGORITHM | RESULTS |
| 1 | B. Sumathy | A Liver Damage Prediction Using Partial Differential Segmentation with Improved Convolutional Neural Network | Predicting Liver diseases using different algorithm with improved accuracy | Deep transfer learning algorithms | To increase the accuracy of the classifier, hybrid classifier approaches may be used. It may be necessary for the liver tumor classification method to evaluate a more significant number of picture samples. |
| 2 | DavidNam,JuliusChapiro | Artificial intelligence in liver diseases: Improving diagnostics, prognostics and response prediction | Predicting Liver diseases using different algorithm with improved accuracy | ML/DL  algorithms | At present, different AI approaches are required to process various types of clinical input data . Recently, there have been increasingly successful attempts to integrate multimodal data in non-medical fields,but such endeavours have not been systematically applied in a medical context beyond highly simplified laboratory conditions. |
| 3 | Nazmun Nahar and  Ferdous Ara | LIVER DISEASE PREDICTION BY USING DIFFERENT DECISION TREE TECHNIQUES | Predicting Liver diseases using different algorithm with improved accuracy | Decision Tree  Algorithm | In future, we will collect the very recent data from various regions across the world for liver disease diagnosis. The results of this study will encourage us to continue developing other advanced decision trees such as CART. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | Dr. S. Vijayarani , Mr.S.Dhayanand | Liver Disease Prediction using SVM and Naïve Bayes Algorithms | Pedicting Liver diseases using different algorithm with improved accuracy | SVM and Naïve Bayes Algorithms | From the experimental results, this work concludes, the SVM classifier is considered as a best algorithm because of its highest classification accuracy. On the other hand, while comparing the execution time, the Naïve Bayes classifier needs minimum execution time. |
| 5 | A.K.M Sazzadur Rahman, F. M. Javed Mehedi Shamrat, Zarrin Tasnim, Joy Roy, Syed Akhter Hossain | A Comparative Study On Liver Disease Prediction Using Supervised Machine Learning Algorithms | Predicting Liver diseases using different algorithm with improved accuracy | Supervised Machine Learning Algorithms | We just explored some popular supervised machine learning algorithms, more  algorithms can be picked to assemble an increasingly precise model of liver disease prediction and performance can be progressively improved. Additionally, this work likewise ready to assume a significant role in health care research and just as restorative focuses to anticipate liver infection. |
| 6 | M. Banu Priya, P. Laura  Juliet, P.R. Tamilselvi | Performance Analysis of Liver Disease Prediction Using Machine Learning Algorithms | Predicting Liver diseases using different algorithm with improved accuracy | Support Vector Machine, Random Forest | The method requires further improvement mostly regarding feature selection of the liver into multiple components:  renal cortex, renal column, renal medulla and renal pelvis. Apart from that, it is planned to expand the database on which the system will be tested |
| 7 | Ain Najwa Arbain,Yushalinie Pillay Balakrishnan | A Comparison of Data Mining Algorithms for Liver Disease Prediction on Imbalanced Data | Predicting Liver diseases using different algorithm with improved accuracy | Data Mining; Classification, SAS  Enterprise  Miner | Due to insufficient data in the data set, Random Forest algorithm is overfitted and is not the most suitable algorithm in this research despite having a perfect result from the ROC chart. Future research could be done on utilizing oversampling method to the dataset to address this issue. |
| 8 | Md.Mohaimenul Islam, Chieh-Chen Wu | Applications of Machine Learning in Fatty Live Disease Prediction | Predicting Liver diseases using different algorithm with improved accuracy | Support Vector Machine (SVM), Artificial Neural Network (ANN), and Logistic Regression (RF) | This prediction outcome has the potential to help clinicians make more precise and meaningful decisions about fatty liver disease diagnosis and treatment. |
| 9  9 | PSM Keerthana | A Prediction Model of Detecting Liver Diseases in Patients using Logistic Regression of Machine Learning | Predicting Liver diseases using different algorithm with improved accuracy | Logistic Regression, Machine Learning, Confusion Matrix, Cross- Validation | Liver disease owing to its subtle symptoms remains obscure and hence leading to an onerous diagnosis, often the symptoms become apparent when it is too late. Therefore, an endeavour is made for the forecast of liver sickness in patients utilizing machine  learning techniques. In this paper, we thus used the Machine Learning method of Logistic Regression to predict liver disease in patients. |

**Table 2.1 - Existing Problems**

**2.2 REFERENCES**

1. P.Karule, Dr. Sanjay Vasant Dudul, "Intelligent Diagnosis of Liver Diseases from Ultrasonic Liver Images: Neural Network Approach", July 2009.

2. G.Sakr, I.Elhajj, H. Huijer, "Support vector machine to deﬁne and detect

Agitation Translation", February 2010.

3. H. Al-Angari, A. Sahakian, "Automated Recognition of Obstructive Sleep

Apnea Syndrome Using Support Vector Machine Classiﬁer", May 2012.

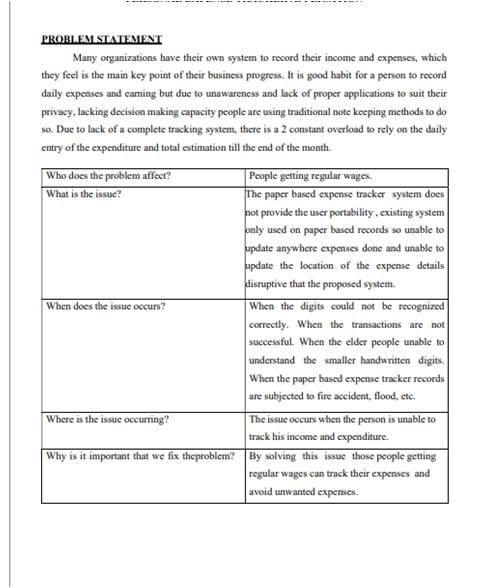
4. Adekola Olubukola Daniel, Ekanem Edikan Uwem, "Prediction and

Diagnosis of Liver Disease in Human Using Machine Learning", August 2021.

5. Sana Ansari, I. Shaﬁ, J.Ahmad, Syed Ismail Shah, "Neural network-based approach for the non-invasive diagnosis and classiﬁcation of hepatotropic viral disease", December 2021.

**2.3 PROBLEM STATEMENT DEFINITION**

Liver diseases avert the normal function of the liver. Mainly due to the large amount of alcohol consumption liver disease arises. Early prediction of liver disease using classiﬁcation algorithms is an eﬃcacious task that can help the doctors to diagnose the disease within a short duration of time. Discovering the existence of liver disease at an early stage is a complex task for the doctors. The main objective of this project is to analyse the parameters of various classiﬁcation algorithms and compare their predictive accuracies so as to ﬁnd out the best classiﬁer for determining the liver disease. This Project examines data from liver patients concentrating on relationships between a key list of liver enzymes, proteins, age and gender using them to try and predict the likeliness of liver disease. Here we are building a model by applying various machine learning algorithms ﬁnd the best accurate model and integrate to ﬂask based web application.

****

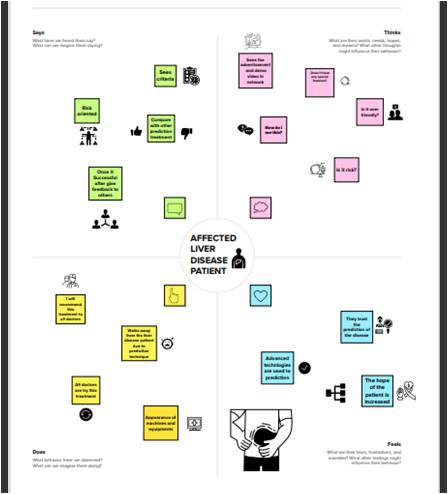
**Table 2.2 - Problem Statement Deﬁnition**

**CHAPTER**

**3**

**IDEATION AND PROPOSED SOLUTION**

**3.1 EMPATHY MAP CANVAS**

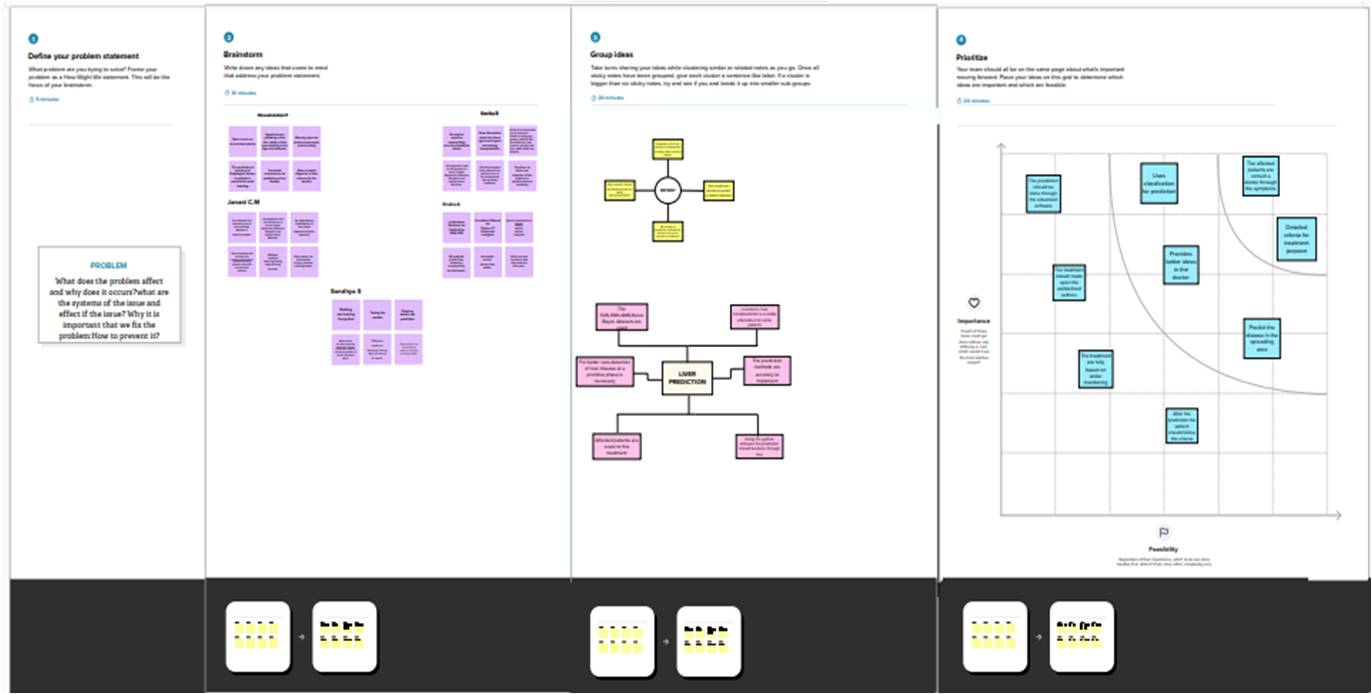


**Figure 3.1 - Empathy Map**

An empathy map is a widely-used visualization tool within the ﬁeld. In relation to **empathetic** design, the primary purpose of an empathy map is to bridge the understanding of the end user.

**3.2 IDEATION AND BRAINSTORMING**

**Step-1:**



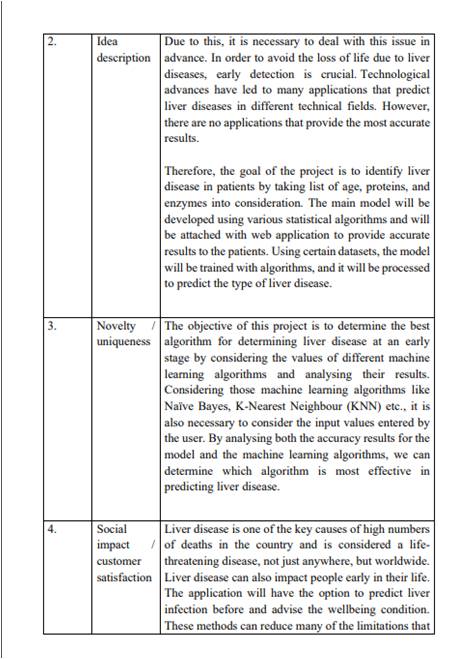
**Figure 3.2 - Ideation and Brainstorming**

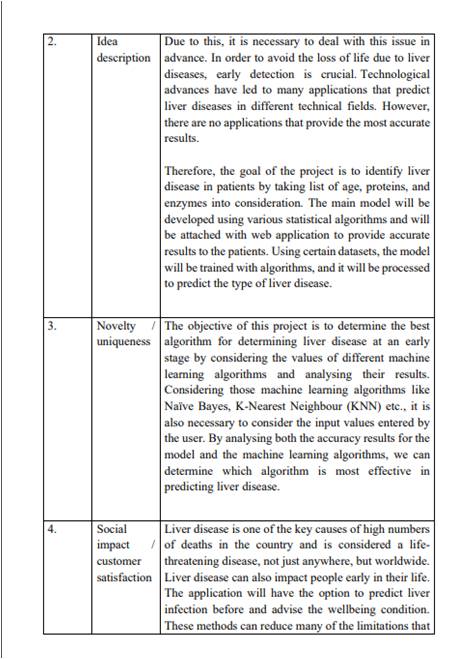
A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.

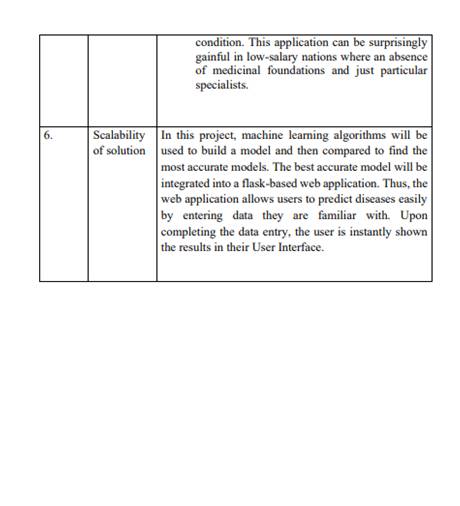
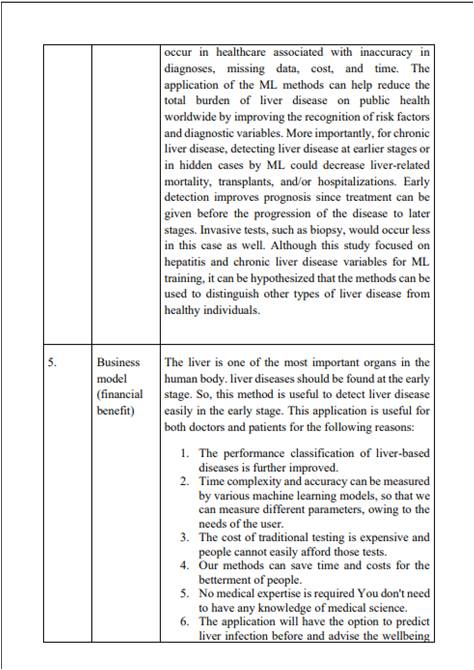
**6**

idea listing and grouping is used to organize and analyse large numbers of ideas by categorising them. By organising and reorganising ideas, students gain a better appreciation of, and dialogue about, their ideas. As students create idea clusters, new contexts and connections among themes emerge.

**3.3 PROPOSED SOLUTION**

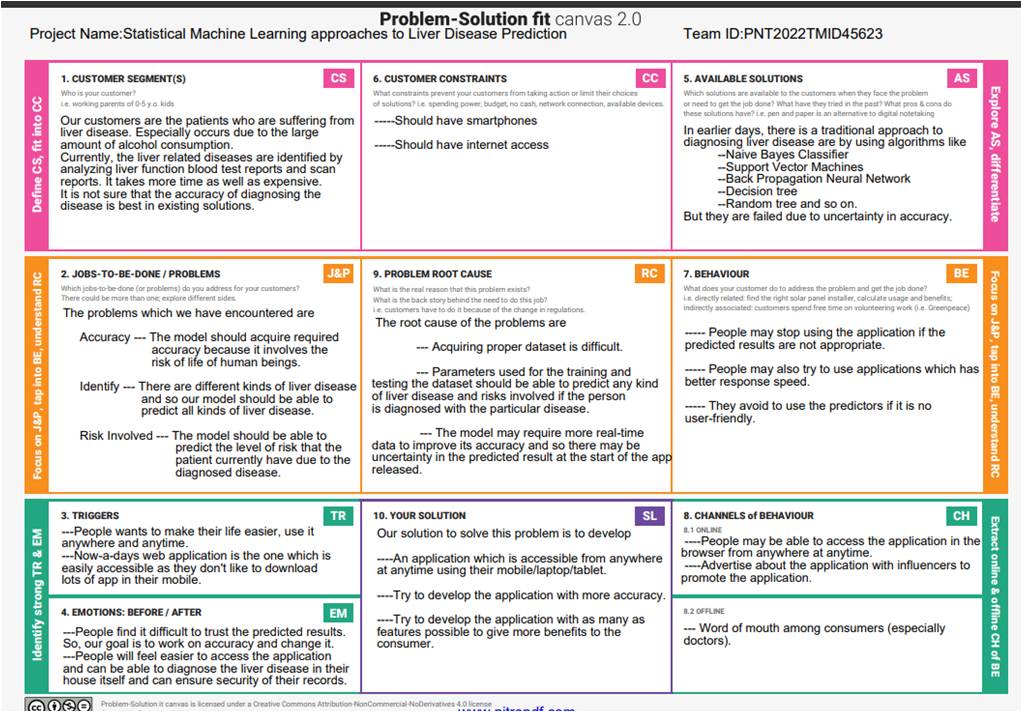
****

****

****

**Table 3.1 - Proposed Solution**

**3.4 PROBLEM SOLUTION FIT**



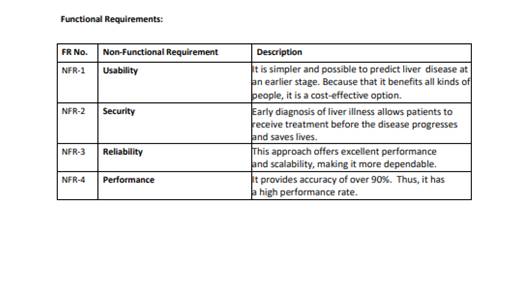
**Figure 3.5 - Solution ﬁt of design with user requirements**

This occurs when the user have evidence that customers care about certain jobs, pains, and gains. At this stage the user proved the existence of a problem and have designed a value proposition that addresses customers' jobs, pains and gains

**CHAPTER 4**

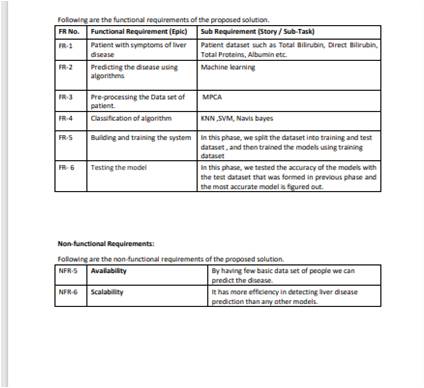
**REQUIREMENT ANALYSIS**

**4.1 FUNCTIONAL REQUIREMENTS**



**Table 4.1 - Functional Requirements**

**4.2 NON-FUNCTIONAL REQUIREMENTS**

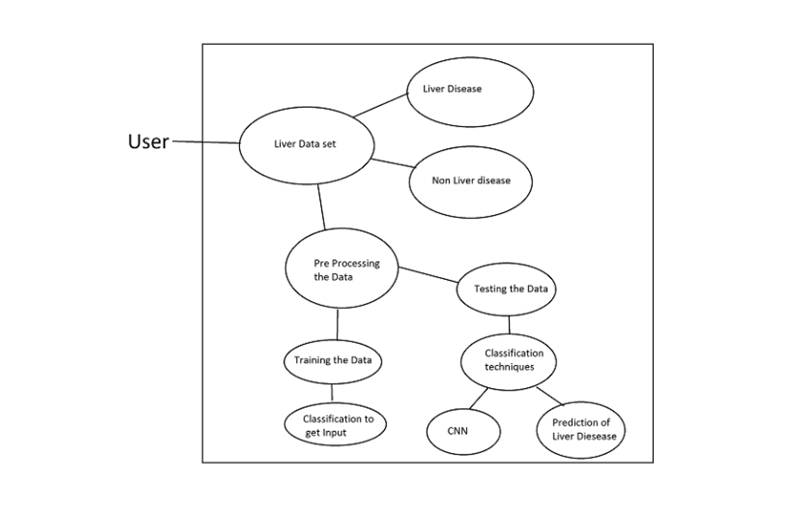
****

**Table 4.2 - Non Functional Requirements**

**CHAPTER 5**

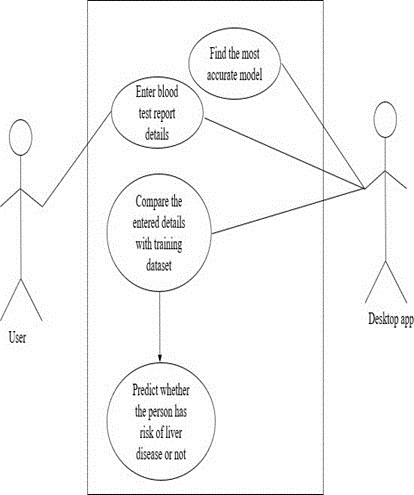
**PROJECT DESIGN**

**5.1 DATAFLOW DIAGRAMS**



**Figure 5.1 - Data ﬂow Diagram**

A data ﬂow diagram is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method.

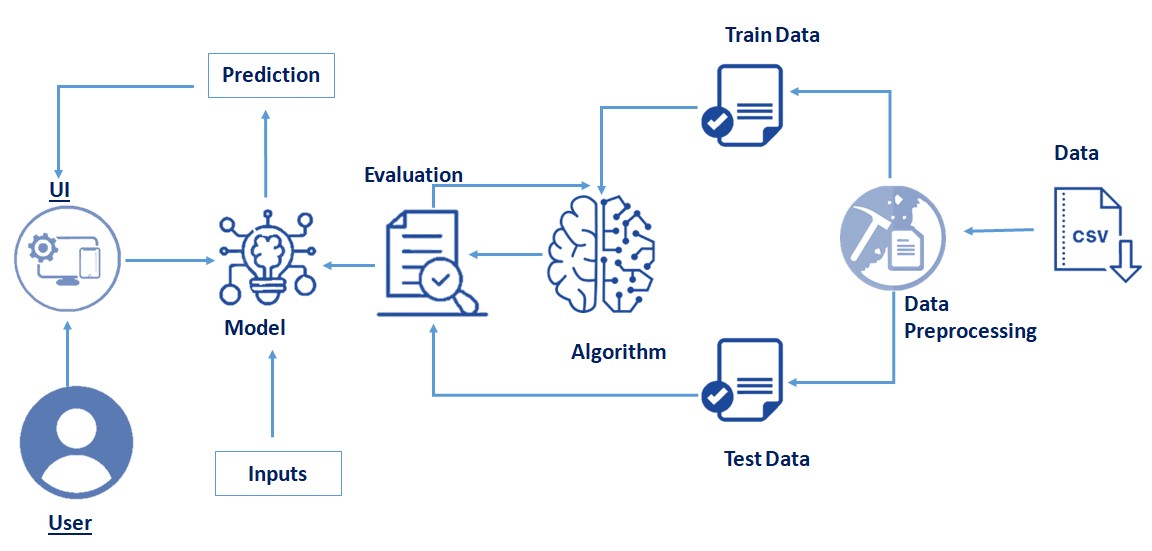


**Figure 5.2 - Describes the ﬂow of the project deployment**

The ﬂow through which applications, modules, updates, and patches are delivered from developers to users. The methods used by developers to build, test and deploy new code will impact how fast a product can respond to changes in customer preferences or requirements and the quality of each change.

**14**

**5.2 SOLUTION AND TECHNICAL ARCHITECTURE**

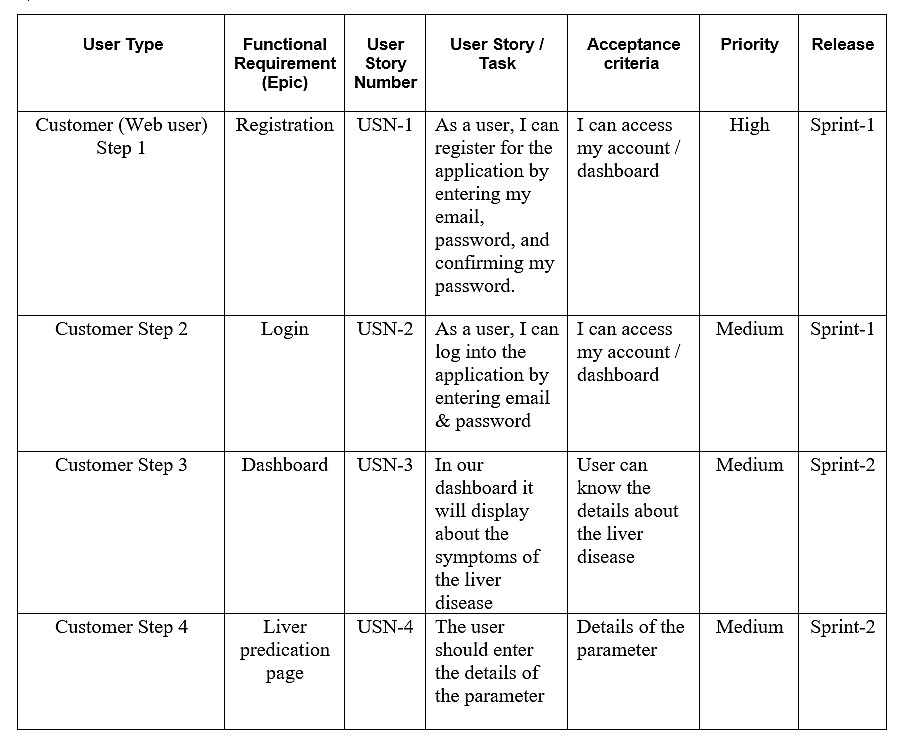


**Figure 5.3 - The process of architecttural description**

Solution Architects are most similar to project managers, ensuring that all parties, including stakeholders, are on the same page and moving in the right direction at all stages. Technical architects manage all activities leading to the successful implementation of a new application. A solution architect must have a technical background with at least eight years of work experience in one or more IT areas including but not limited to: IT architecture, infrastructure, and cloud development.

**15**

**5.3 USER STORY**



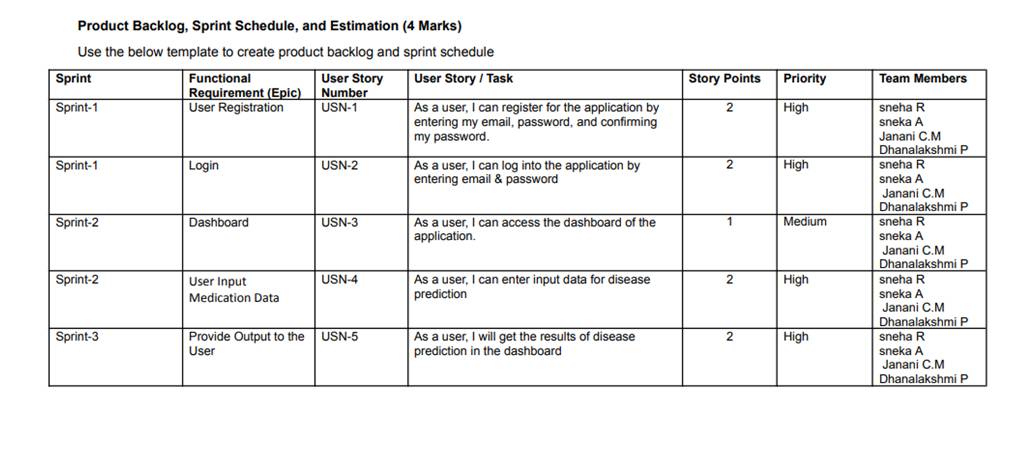
**Table 5.1 - User Story**

**16**

**CHAPTER 6**

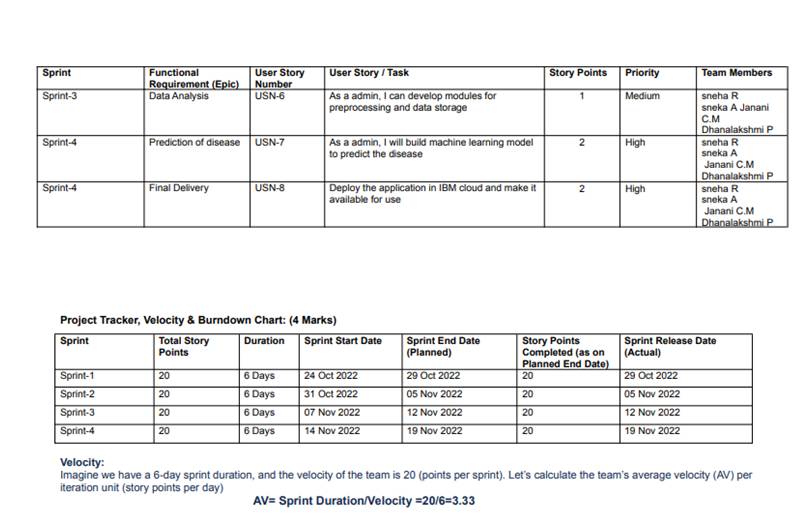
**PROJECT PLANNING AND SCHEDULING**

**6.1 SPRINT PLANNING AND ESTIMATION**



**Table 6.1 - Sprint planning and estimation**

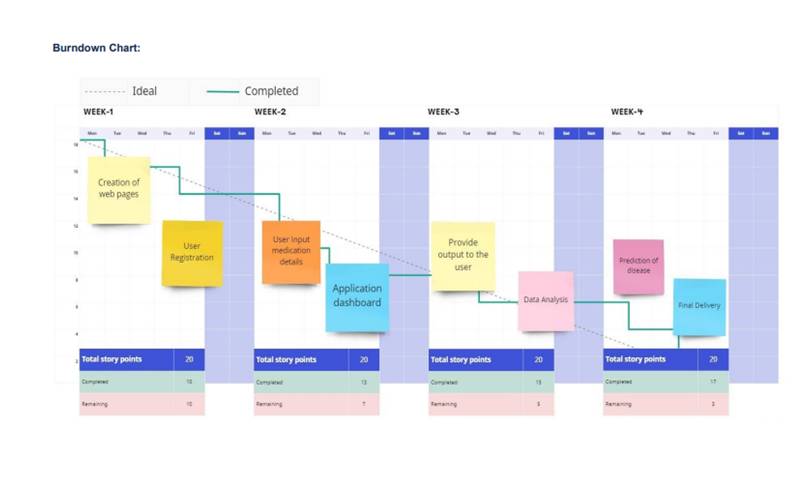
**6.2 SPRINT DELIVERY SCHEDULE**



**Table 6.2 - Sprint Delivery Schedule**

**6.3 REPORTS FROM JIRA**

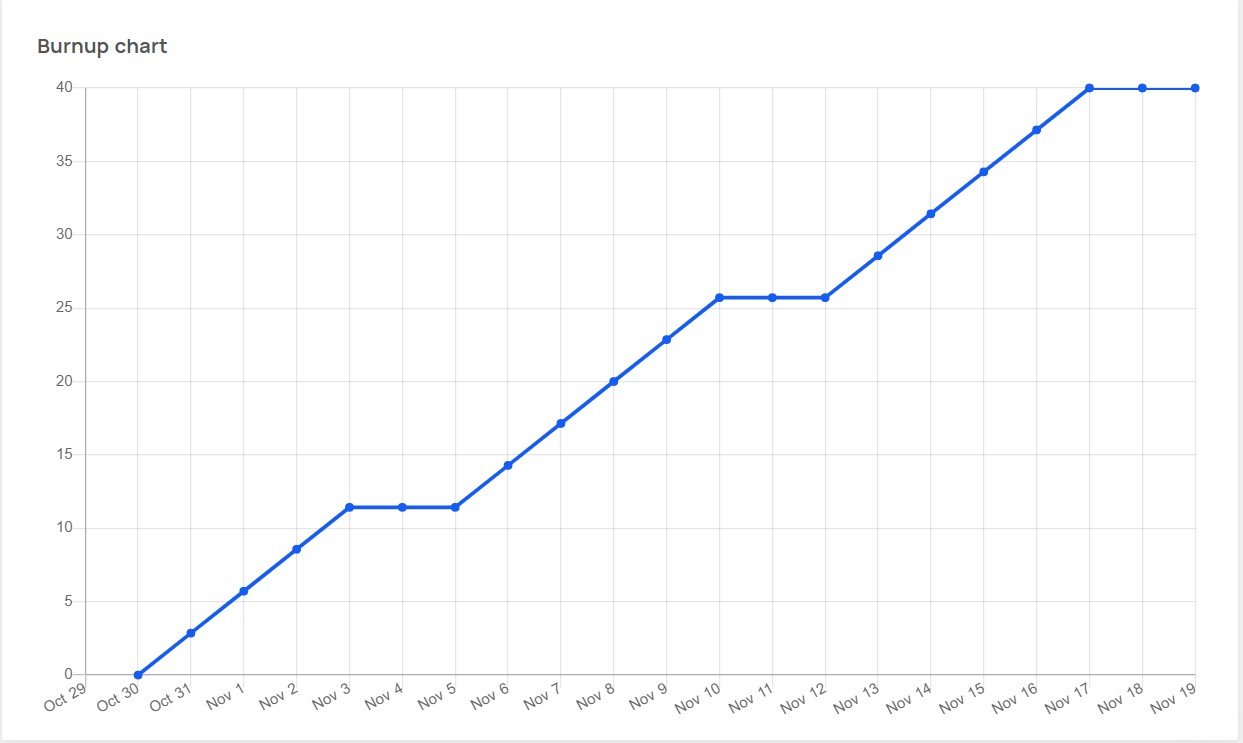
**Burdown Chart**



**Figure 6.1 - Burndown Chart**

A burndown chart shows the amount of work that has been completed in an epic or sprint, and the total work remaining. Burndown charts are used to predict your team's likelihood of completing their work in the time available.

**19**



**Burnup Chart**

**Figure 6.2 - Burnup Chart**

A burnup chart highlights the work you've completed against your total project scope while a burn down chart highlights the amount of work remaining in a project. A burnup chart contains a work completed line and a project scope line. It displays the scope of a project and the work completed

**20**

**CHAPTER 7**

**CODING AND SOLUTION**

**7.1 FEATURE 1**

The ﬁrst feature of the deployement is the process of

Random Forest

Classiﬁer is used to train and test the model for detecting the Liver Disease with the help of collected and pre-processed dataset collections.

# Train Test Split:

from sklearn.model\_selection import train\_test\_split

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X\_smote,y\_smote, test\_size=0.3, random\_state=33)

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classiﬁcation and Regression problems. It builds decision trees on different samples and takes their majority vote for classiﬁcation and average in case of regression.

# RandomForestClassiﬁer:

from sklearn.ensemble import RandomForestClassiﬁer RandomForest = RandomForestClassiﬁer() RandomForest = RandomForest.ﬁt(X\_train,y\_train)

# Predictions:

y\_pred = RandomForest.predict(X\_test)

# Performance:

print('Accuracy:', accuracy\_score(y\_test,y\_pred))

print(confusion\_matrix(y\_test,y\_pred))

**21**

print(classiﬁcation\_report(y\_test,y\_pred))

Gradient boosting classiﬁers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive

model. Decision trees are usually used when doing gradient boosting.

# GradientBoostingClassiﬁer:

from sklearn.ensemble import GradientBoostingClassiﬁer GradientBoost = GradientBoostingClassiﬁer() GradientBoost = GradientBoost.ﬁt(X\_train,y\_train)

# Predictions:

y\_pred = GradientBoost.predict(X\_test)

# Performance:

print('Accuracy:', accuracy\_score(y\_test,y\_pred)) print(confusion\_matrix(y\_test,y\_pred)) print(classiﬁcation\_report(y\_test,y\_pred))

AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classiﬁcation problem. The most suited and therefore most common algorithm used with AdaBoost are decision

trees with one level.

# AdaBoostClassiﬁer:

from sklearn.ensemble import AdaBoostClassiﬁer

AdaBoost = AdaBoostClassiﬁer() AdaBoost = AdaBoost.ﬁt(X\_train,y\_train)

**22**

# Predictions:

y\_pred = AdaBoost.predict(X\_test)

# Performance:

print('Accuracy:', accuracy\_score(y\_test,y\_pred)) print(confusion\_matrix(y\_test,y\_pred)) print(classiﬁcation\_report(y\_test,y\_pred))

**7.2 FEATURE 2**

Python ﬂask is the ﬁrst feature that helps to complete this project. It

allows the user to create local server and host the website in a local machine.

from ﬂask import Flask, render\_template, request import numpy as np

import pickle import requests

import json

Here we import all the necessary features of this project involving in Python

ﬂask.

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

mltoken}

app = Flask(\_name\_)

model = pickle.load(open('liver2.pkl', 'rb'))

[@app.route](mailto:@app.route)('/',methods=['GET'])

def Home():

return render\_template('index.html')

Here we created a local client’s own server which serves the .html pages to the

**23**

users.

[@app.route](mailto:@app.route)("/predict", methods=['POST'])

def predict():

if request.method == 'POST': Age = int(request.form['Age']) Gender = int(request.form['Gender'])

Total\_Bilirubin = ﬂoat(request.form['Total\_Bilirubin']) Alkaline\_Phosphotase = int(request.form['Alkaline\_Phosphotase']) Alamine\_Aminotransferase = int(request.form['Alamine\_Aminotransferase'])

Aspartate\_Aminotransferase = int(request.form['Aspartate\_Aminotransferase']) Total\_Protiens = ﬂoat(request.form['Total\_Protiens'])

Albumin = ﬂoat(request.form['Albumin']) Albumin\_and\_Globulin\_Ratio = ﬂoat(request.form['Albumin\_and\_Globulin\_Ratio'] values = np.array([[Age,Gender,Total\_Bilirubin,Alkaline\_Phosphotase,Alamine\_Aminotran sferase,Aspartate\_Aminotransferase,Total\_Protiens,Albumin,Albumin\_and\_Glob ulin\_Ratio]])

prediction = model.predict(values)

return render\_template('result.html', prediction=prediction if \_name\_ == "\_main\_":

app.run(debug=True)

Here we use the inputs from the html pages which has to be get by using request method in Python Flask. By validating the values from the database, we allow the user to access the home page. render\_template: Used for rendering html pages on browser. url\_for: Passing the control of the program to another function. session: Creates a separate session for the individual use.

**CHAPTER 8**

**TESTING**

**8.1 TEST CASES**

|  |  |  |  |
| --- | --- | --- | --- |
| **TEST CASE ID** | 45623 | **TEST CASE DESCRIPTION** | STATISTICAL MACHINE LEARNING APPROACHES TO LIVER DISEASE PREDICTION |

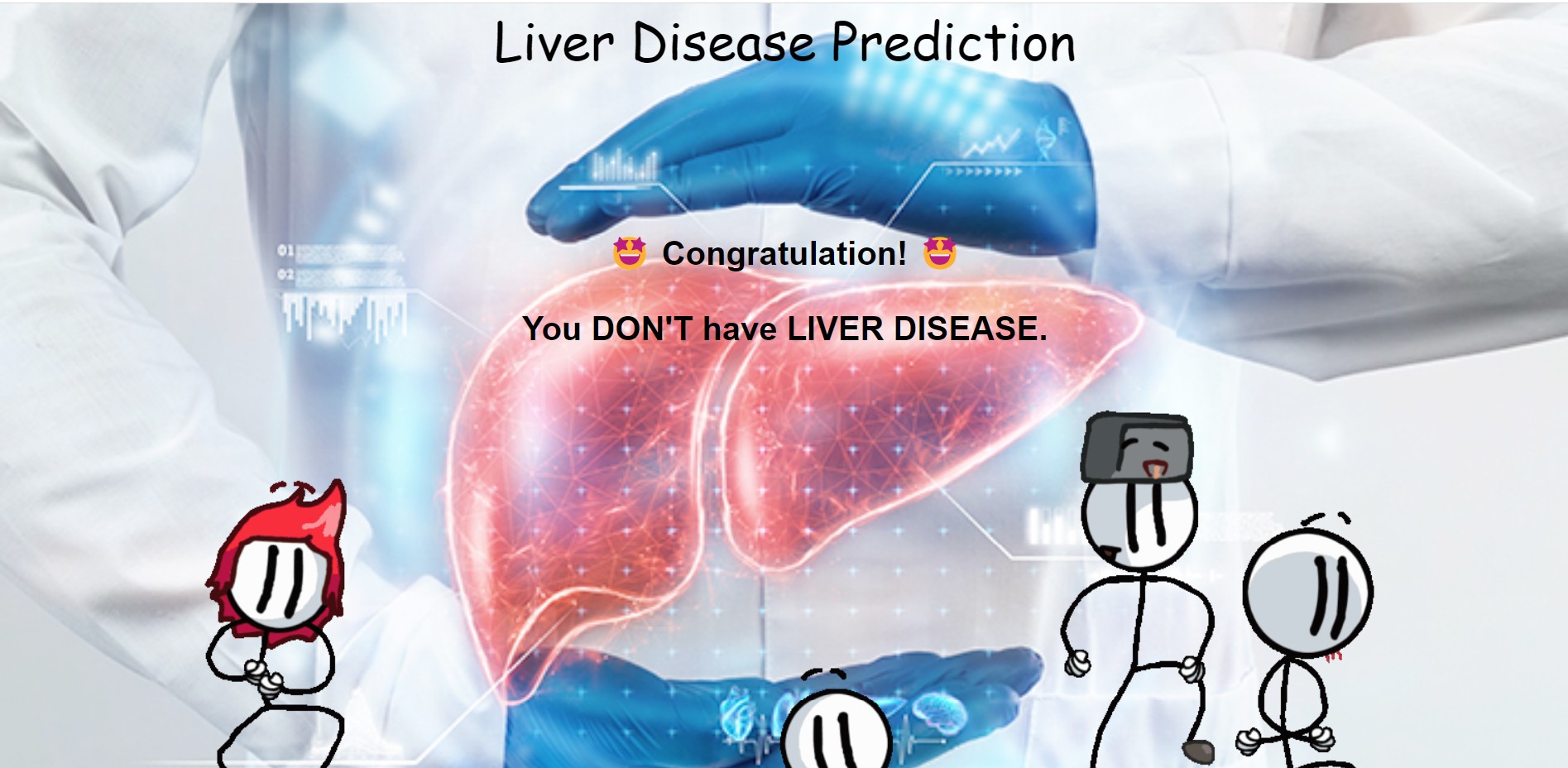
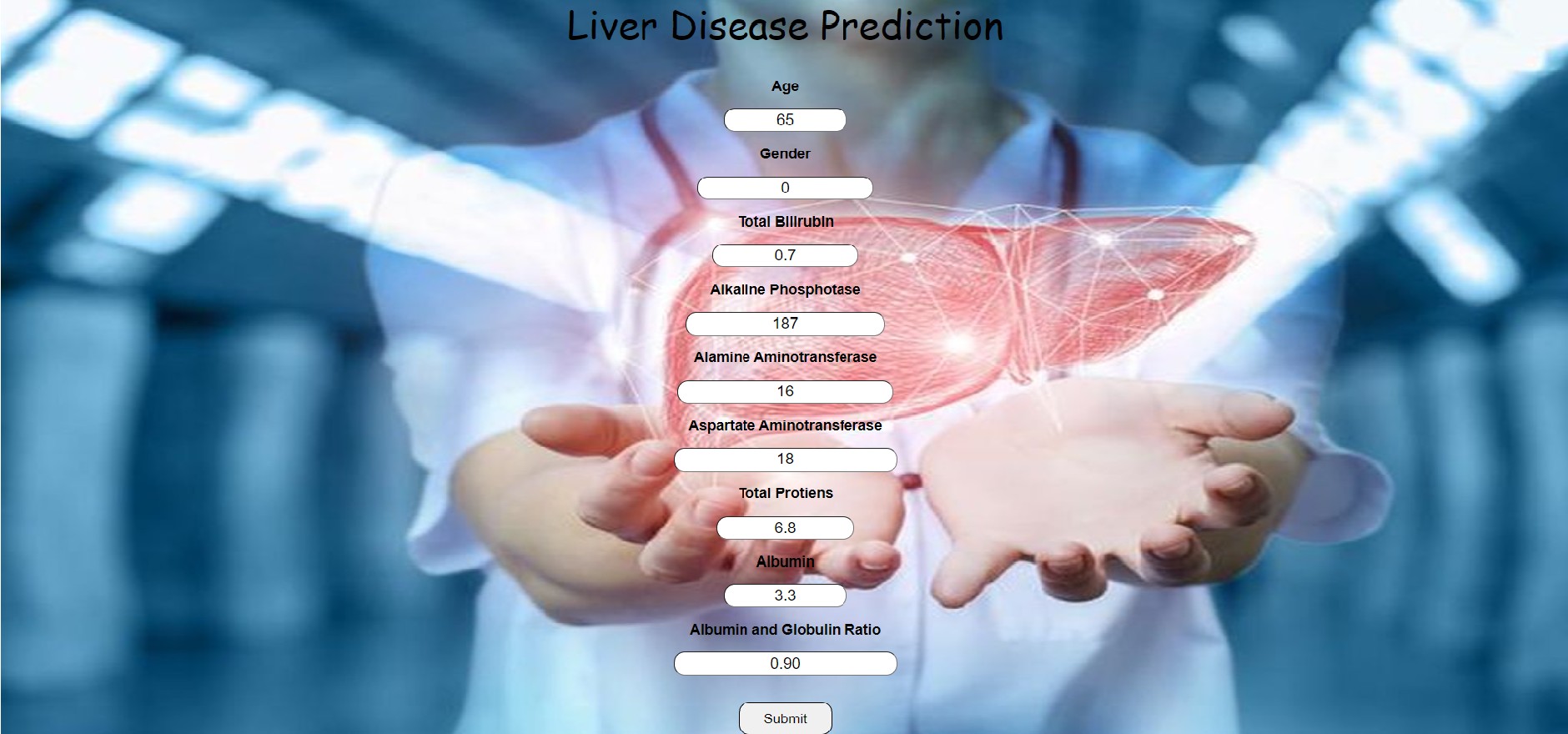
|  |  |  |
| --- | --- | --- |
| **S.No.** | **PREREQUISITES** | **TEST DATA** |
| 1 | Access to Chrome Browser | By clicking the website link |
| 2 | Entering the details required | Details should be in a integer format |
| 3 | Check for correct values | Data sholud be filled |
| 4 | Application to train the model | Provide the datasets for model training |

**Test Scenario :**

**Table 8.1 - Test Details**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Step** | **Step Details** | **Expected Results** | **Actual Results** | **Pass/Fail/Not/ Executed/ Suspended** |
| 1 | Navigate to website link | Site should open | As Expected | pass |
| 2 | Enter the details | Details should be entered | As Expected | pass |
| 3 | Click Submit | Check the result | As Expected | Pass |
| 4 | Output results | Result are generated | As Expected | Pass |

**Table 8.2 - Test Cases**



**8.2 USER ACCEPTANCE TESTING**

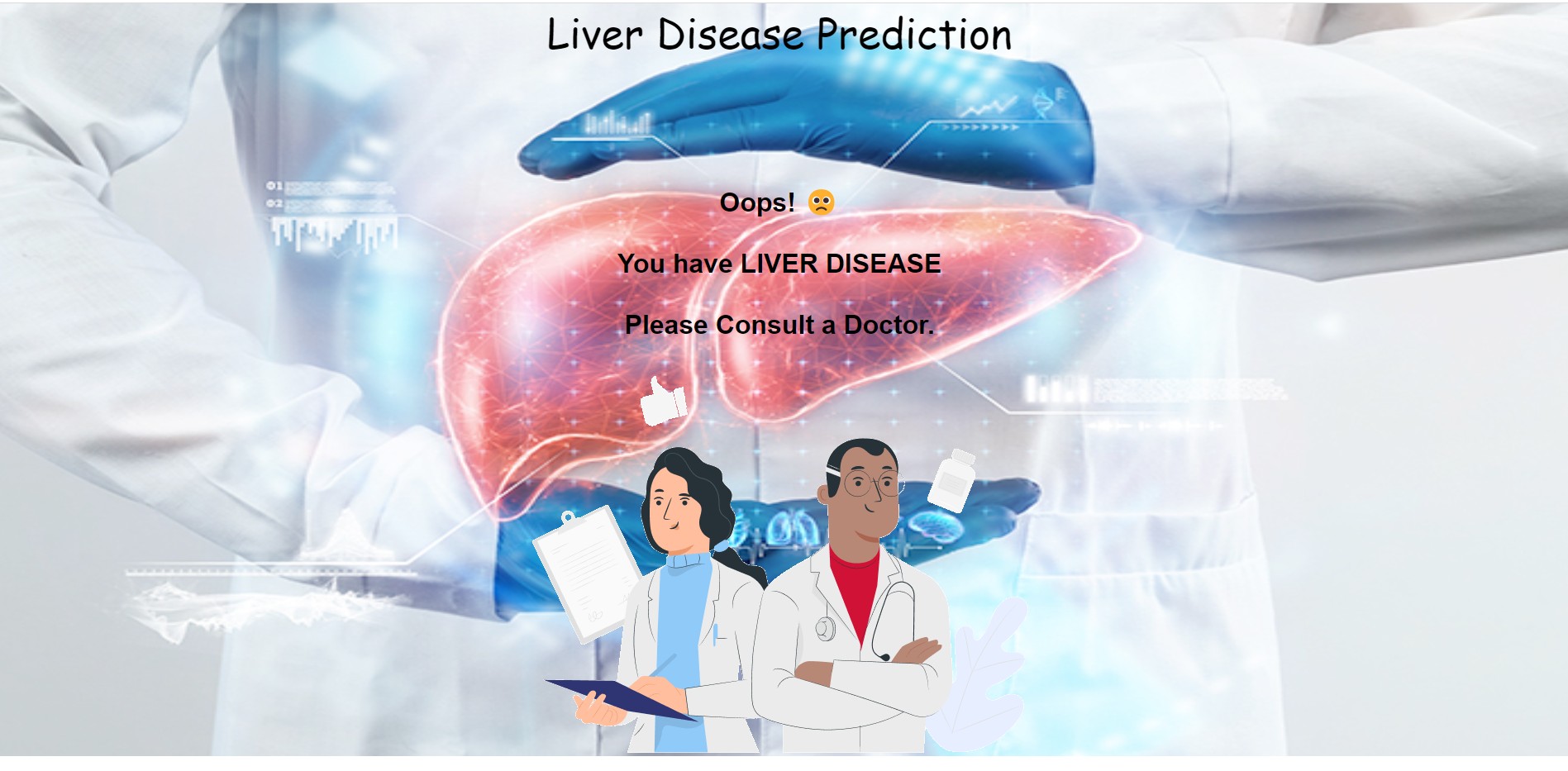
**Figure 8.1 - User Acceptance Test 1**

In this the user will be entering tested values which is moved for prediction.

**Figure 8.2 - User Acceptance Test 2**

In this the data which was entered by the user will be analyzed. This ﬁgure shows the person does not have liver disease.

**.**

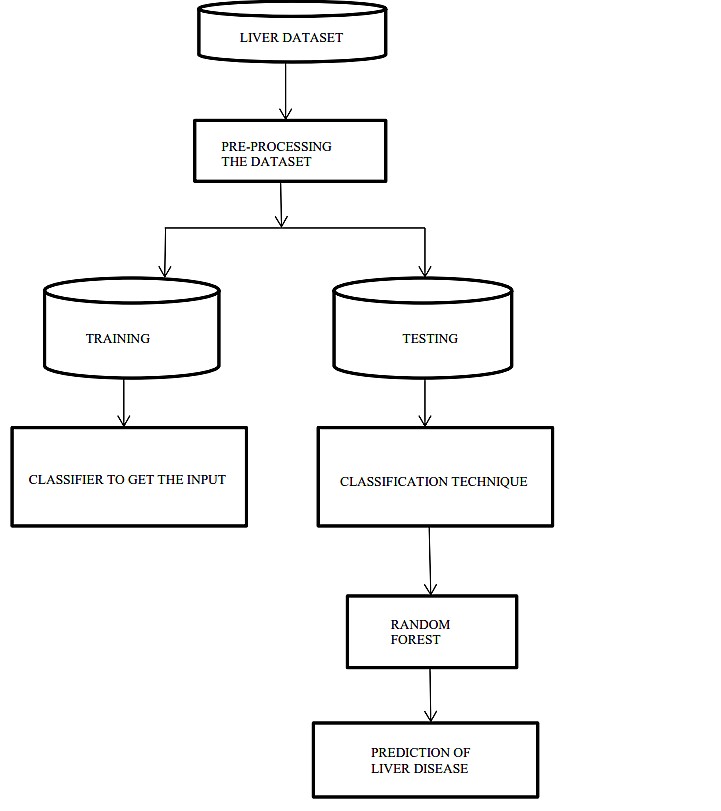


**Figure 8.3 - User Acceptance Test 3**

In this the data which was entered by the user will be analyzed. This ﬁgure shows the person have a liver disease.

**27**

**CHAPTER 9**



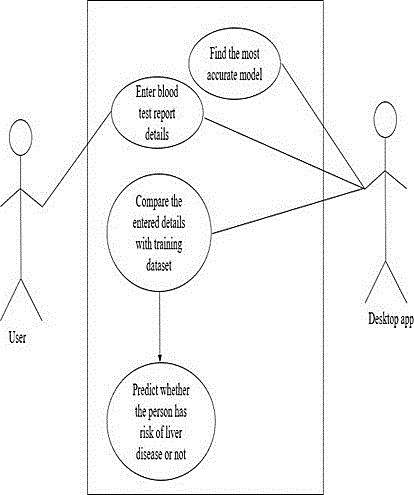
**RESULTS**

**9.1 PERFORMANCE METRICS**

**Figure 9.1 - Flow of the Performance**

Performance Metrics is performed along for all the above assigned tasks. In each and every tasks various metrics is performed inorder to provide the optimum outcome.Results gives the best outcome as expected in project.

**28**



**Figure 9.2 - Performance Matrix**

One can use following execution measures for the request and ﬁgure of imperfection slanted module as shown by his/her own need.

**Confusion Matrix:** The confusion matrix is used to measure the introduction of two class issue for the given instructive record. The right corner to corner parts TP (True positive) and TN (True Negative) adequately describe instances similarly as FP (false positive) and FN (false negative) wrongly request instances. Confusion Matrix correctly classify instance TP+TN incorrectly classify instances.

● True positives imply the positive liver tuples that were precisely named by the classiﬁer,

● True negatives are the negative liver tuples that were precisely set apart by the classiﬁer.

● False positives are the negative liver tuples that were erroneously set apart as positive tuples.

● False negatives are the positive liver tuples that were incorrectly stamped negative tuples.

**29**

**CHAPTER 10**

**ADVANTAGES AND DISADVANTAGES**

**ADVANTAGES**

In pharmaceutical industries, random forest can be used to identify the potential of a certain medicine or the composition of chemicals required for medicines. Random forest can also be used in hospitals to identify the diseases suffered by a patient ,risk of liver disease and many other diseases where early analysis and research play a crucial role.Our study has successfully developed and validated the ﬁrst risk prediction model and subsequent user-friendly scoring tool, the algorithm for Liver Function Investigations, for liver condition diagnosis in patients with no obvious liver condition at the time of incident liver function testing in primary care.

**DISADVANTAGES**

The idea for the approach of this project arises from the current situation regarding the increase in the conﬁrmatory diagnosis of liver, and lack of treatment or the user's ignorance of its pathologies leads to irreversible kidney failure in the ﬁnal stages of disease, such as dialysis for life, ﬁnancially affecting the health system, as it is a costly treatment that generates the most signiﬁcant amount of absorption of the resources available for health. This could be reduced by using tools such as machine learning to classify from the initial stages. Although the application of machine learning in healthcare and other areas is favorable, the ﬁeld of liver disease has not yet exploited its full

potential.

**30**

**CHAPTER 11**

**CONCLUSION**

The principal part of this work is to make an effective diagnosis system for liver infection patients. The application will have the option to predict liver infection prior and advise the wellbeing condition. This application can be surprisingly gainful in low-salary nations where our absence of medicinal foundations and just as particular specialists. In our study, there are a few bearings for future work in this ﬁeld. We just explored some popular supervised machine learning algorithms, more algorithms can be picked to assemble an increasingly precise model of liver disease prediction

and performance can be progressively improved. Additionally, this work

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| likewise ready | to assume a signiﬁcant role in health | care | research | and | just |
| as restorative | focuses to anticipate liver infection. |  |  |  |  |

**31**

**CHAPTER 12**

**FUTURE SCOPE**

Diseases related to liver is becoming more and more common with time. With continuous technological advancements, these are only going to increase in the future. Although people are becoming more conscious of health nowadays and are joining yoga classes, dance classes; still the sedentary lifestyle and luxuries that are continuously being introduced and enhanced; the problem is going to last long. So, in such a scenario, our project will be extremely helpful to the society. With the dataset that we used for this project, we got 81% accuracy for Random forest model, and though it might be diﬃcult to get such accuracies with very large datasets, from this projects results, one can clearly conclude that we can predict the risk of liver diseases with accuracy of 90 % or more. Also it can be incorporated into a website and these app and

website will be highly beneﬁcial for a large section of society.

**32**

**CHAPTER 13**

**APPENDIX**

**SOURCE CODE:**

**Algorithm :**

#Importing The Libraries import pandas as pd import numpy as np

import matplotlib.pyplot as plt import seaborn as sns

import pickle

#Reading the Dataset

data=pd.read\_csv(path)

#Exploratory Data Analysis data.head()

data.tail()

data.info()

data.describe()

#Checking For Null Values And Handling Null Values data.isnull().any()

data.isnull().sum()

data['Albumin\_and\_Globulin\_Ratio']=data.ﬁllna(data['Albumin\_and\_Globulin\_Ratio']

.mode()[0])

data.isnull().sum()

#Data Visualization sns.countplot(data=data,x='Gender',lable='Count') m,f=data['Gender'].value\_counts()

print("No of Males:",m)

print("No of Females:",f)

**33**

sns.countplot(data=data, x='Dataset') LD,NLD=data['Dataset'].value\_counts() print("liver disease patients:",LD) print("Non-liver disease patients:",NLD)

#Splitting The Dataset Into Dependent And Independent Variable x=data.iloc[:,0:-1]

y=data.iloc[:,-1]

#Split The Dependent And Independent Features Into Train Set And TestSet from sklearn.model\_selction import train\_test\_split

xtrain.xtest,ytrain,ytest=train\_test\_split(x,y,test\_size=0.2)

#Check the shape of both xtrain and xtest. xtrain.shape

xtest.shape

#importing the machine learning model from sklearn.svm import svc

from sklearn.ensemble import RandomForestClassiﬁer from sklearn.neighbors import KNeighborsClassiﬁer

#initiailizing the machine learning models svm=SVC() RFmodel=RandomForestClassiﬁer() KNNmodel=KNeighborsClassiﬁers() svm=SVC()

#Train the data with SVM model svm.ﬁt(xtrain,ytrain)

#Random forest classiﬁers model

from sklearn.ensemble import RandomForestClassiﬁer

RFmodel=RandomForestClassiﬁer()

#Train the data with random forest model RFmodel.ﬁt(xtrain,ytrain) SVMaccuracy=accuracy\_score(SVMpred,ytest) SVMaccuracy

**34**

from sklearn.ensemble import RandomForestClassiﬁer RFmodel=RandomForestClassiﬁer() RFmodel.ﬁt(xtrain,ytrain) RFpred=RFmodel.predict(xtest) RFaccuracy=accuracy\_score(RFpred,ytest)

RFaccuracy RFcm=confusion\_matrix(RFpred,ytest) RFcm

from sklearn.neighbors import KNeighborsClassiﬁer

KNN=KNeighborsClassiﬁer() KNN.ﬁt(xtrain,ytrain) KNNpred=KNN.preduct(xtest) KNNaccuracy=accuracy\_score(KNNpred,ytest) KNNaccuracy KNNcm=confusion\_matrix(KNNpred,ytest) KNNcm

# saving the model import pickle

pickle.dump(svm,open('liver\_analysis.pkl','wb'))

**Flask Connection**

from ﬂask import Flask, render\_template, request import numpy as np

import pickle

app = Flask(\_\_name )

model = pickle.load(open('Liver2.pkl', 'rb'))

[@app.route](mailto:@app.route)('/',methods=['GET'])

def Home():

return render\_template('index.html')

[@app.route](mailto:@app.route)("/predict", methods=['POST'])

def predict():

if request.method == 'POST': Age = int(request.form['Age'])

**35**

Gender = int(request.form['Gender'])

Total\_Bilirubin = ﬂoat(request.form['Total\_Bilirubin']) Alkaline\_Phosphotase = int(request.form['Alkaline\_Phosphotase']) Alamine\_Aminotransferase =

int(request.form['Alamine\_Aminotransferase']) Aspartate\_Aminotransferase =

int(request.form['Aspartate\_Aminotransferase']) Total\_Protiens = ﬂoat(request.form['Total\_Protiens']) Albumin = ﬂoat(request.form['Albumin']) Albumin\_and\_Globulin\_Ratio =

ﬂoat(request.form['Albumin\_and\_Globulin\_Ratio'])

values = np.array([[Age,Gender,Total\_Bilirubin,Alkaline\_Phosphotase,Alamine\_Amino transferase,Aspartate\_Aminotransferase,Total\_Protiens,Albumin,Albumin\_a nd\_Globulin\_Ratio]])

prediction = model.predict(values)

return render\_template('result.html', prediction=prediction)

if name == " main ":

app.run(debug=True)

**HTML Templates index.htm**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Liver Prediction Model</title>

</head>

<body>

<div class="container">

<h2 class='container-heading'><span class="heading\_font">Liver

Disease Prediction</span></h2>

**36**

</div>

<div class="ml-container">

<form action="{{ url\_for('predict') }}" method="POST">

<br>

<h3>Age</h3>

<input id="ﬁrst" name="Age" placeholder="in Year" required="required">

<br>

<h3>Gender</h3>

<input id="second" name="Gender" placeholder="Male = 1, Female=0" required="required">

<br>

<h3>Total Bilirubin</h3>

<input id="third" name="Total\_Bilirubin" placeholder="Total Bilirubin" required="required">

<br>

<h3>Alkaline Phosphotase</h3>

<input id="fourth" name="Alkaline\_Phosphotase" placeholder="Alkaline Phosphotase" required="required">

<br>

<h3>Alamine Aminotransferase</h3>

<input id="ﬁfth" name="Alamine\_Aminotransferase" placeholder="Alamine Aminotransferase" required="required">

<br>

<h3>Aspartate Aminotransferase</h3>

<input id="sixth" name="Aspartate\_Aminotransferase" placeholder="Aspartate Aminotransferase" required="required">

<br>

<h3>Total Protiens</h3>

**37**

<input id="seventh" name="Total\_Protiens" placeholder="Total

Protiens" required="required">

<br>

<h3>Albumin</h3>

<input id="eight" name="Albumin" placeholder="Albumin" required="required">

<br>

<h3>Albumin and Globulin Ratio</h3>

<input id="ninth" name="Albumin\_and\_Globulin\_Ratio" placeholder="Albumin and Globulin Ratio" required="required">

<br>

<br>

<br>

<button id="sub" type="submit ">Submit</button>

<br>

<br>

<br>

<br>

</form>

</div>

<style>

/\* Background Image \*/

body

{

background- image:url("https://akcdn.detik.net.id/community/media/visual/2020/12/17

/6-makanan-untuk-mencegah-penyakit-liver-salah-satunya-kopi-

5\_43.jpeg?w=250&q=");

height: 100%;

/\* Center and scale the image nicely \*/

**38**

background-position: center; background-repeat: no-repeat; background-size: 100% 100%;

}

/\* Color \*/

body{

font-family: Arial, Helvetica,sans-serif;

text-align: center; margin: 0; padding: 0;

width: 100%; height: 100%; display: ﬂex;

ﬂex-direction: column;

}

/\* Heading Font \*/

.container-heading{

margin: 0;

}

.heading\_font{

color: #black;

font-family: 'Paciﬁco', cursive;

font-size: 50px;

font-weight: normal;

}

/\* Box \*/

#ﬁrst {

border-radius: 14px;

height: 25px;

width: 150px;

**39**

font-size: 20px;

text-align: center;

}

#second {

border-radius: 14px;

height: 25px; width: 220px; font-size: 20px;

text-align: center;

}

#third {

border-radius: 14px;

height: 25px; width: 180px; font-size: 20px;

text-align: center;

}

#fourth {

border-radius: 14px;

height: 25px; width: 250px; font-size: 20px;

text-align: center;

}

#ﬁfth {

border-radius: 14px;

height: 25px; width: 270px; font-size: 20px;

text-align: center;

**40**

}

#sixth {

border-radius: 14px;

height: 25px; width: 280px; font-size: 20px;

text-align: center;

}

#seventh {

border-radius: 14px;

height: 25px; width: 170px; font-size: 20px;

text-align: center;

}

#eight {

border-radius: 14px;

height: 25px; width: 150px; font-size: 20px;

text-align: center;

}

#ninth {

border-radius: 14px;

height: 25px; width: 280px; font-size: 20px;

text-align: center;

}

/\* Submit Button \*/

**41**

#sub {

}

width: 120px;

height: 43px;

text-align: center; border-radius: 14px; font-size: 18px;

</style>

</body>

</html>

**result.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Liver Disease Result</title>

</head>

<body>

<div class="container">

<form action="{{ url\_for('predict')}}" method="post">

<h2 class='container-heading'><span class="heading\_font">Liver

Disease Prediction</span></h2>

<br><br><br><br><br><br><br>

<!-- Result -->

<div class="results">

{% if prediction==2 %}

<h1><span class='danger'>Oops! 🙁<br><br>You have LIVER DISEASE <br><br>Please Consult a Doctor.</span></h1>

<img class="gif" src="{{ url\_for('static', ﬁlename='dr.gif')}}"

**42**

alt="LIVER Image">

{% elif prediction==1 %}

<h1><span class='safe'>🤩 Congratulation! 🤩<br><br>You DON'T

have LIVER DISEASE.</span></h1>

<img class="gif1" src="{{ url\_for('static', ﬁlename='yes.webp')}}" alt="Not LIVER Image">

{% endif %}

</div>

</form>

</div>

<style>

/\* Background Image \*/

body

{

background-image:url("https://d1vbn70lmn1nqe.cloudfront.net/prod/wp- content/uploads/2022/05/11050117/hepatitis-kronis-halodoc.jpg"); height: 100%;

/\* Center and scale the image nicely \*/ background-position: center; background-repeat: no-repeat; background-size: 100% 100%;

}

/\* Color \*/

body{

font-family: Arial, Helvetica,sans-serif;

text-align: center; margin: 0; padding: 0;

width: 100%;

height: 100%;

**43**

display: ﬂex;

ﬂex-direction: column;

}

/\* Heading Font \*/

.container-heading{

margin: 0;

}

.heading\_font{

color: #black;

font-family: 'Paciﬁco', cursive;

font-size: 50px;

font-weight: normal;

}

</style>

</body>

</html>

**Github and Project Video Demo Link**

**Github Link:**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | |  | | | |

<https://github.com/IBM-EPBL/IBM-Project-20170-1659714069>

**Project Video Demo Link:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | |  | |  |  | https://mail.google.com/mail/u/0/images/cleardot.gif  https://mail.google.com/mail/u/0/images/cleardot.gif |
| |  | | --- | |  | | | |

<https://youtube.com/shorts/nR_zep9hfdM?feature=share>

**44**