PROJECT DOCUMENTATION

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Team Id:-PNT2022TMID49992

Submitted By

U.BHARATHI-951219104007

S.AMEENAL SAHIFTHA PARVEEN-951219104003

M.KALAIVANI-951219104020

C.NARAYANI-951219104033

Table of Contents

Source Code

GitHub & Project Demo Link

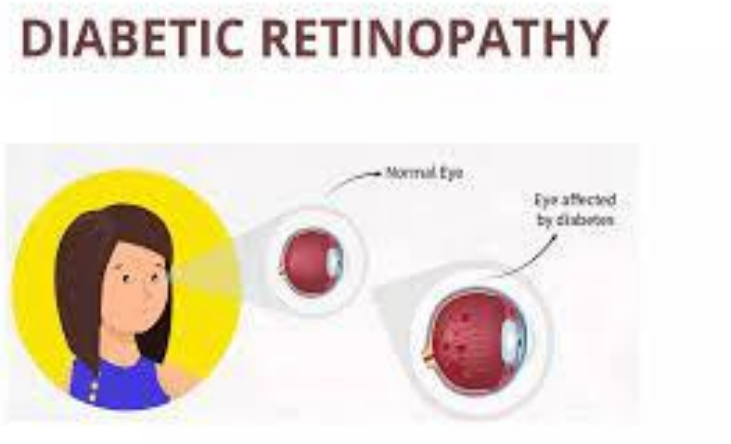
DATE	18 NOV 2022
TEAM ID	PNT2O22TMID49992
PROJECT NAME	DEEP LEARNING FUNDUS IMAGE ANALYSIS FOR EARLY DETECTION OF DIABETIC RETINOPATHY

1.INDROUCTION

Diabetic Retinopathy (DR) is a common complication of diabetes mellitus, which causes lesions on the retina that affect vision. If it is not detected early, it can lead to blindness. Unfortunately, DR is not a reversible process, and treatment only sustains vision. DR early detection and treatment can significantly reduce the risk of vision loss. The manual diagnosis process of DR retina fundus images by ophthalmologists is time, effort and cost-consuming and prone to misdiagnosis unlike computer-aided diagnosis systems.

1.1 PROJECT OVERVIEW

Diabetes mellitus refers to a group of diseases that affect how the body uses blood sugar (glucose). Glucose is an important source of energy for the cells that make up the muscles and tissues. It's also the brain's main source of fuel. The main cause of diabetes varies by type. But no matter what type of diabetes you have, it can lead to excess sugar in the blood. Too much sugar in the blood can lead to serious health problems. Chronic diabetes conditions include type 1 diabetes and type 2 diabetes. Potentially reversible diabetes conditions include pre diabetes and gestational diabetes. Pre diabetes happens when blood sugar levels are higher than normal. But the blood sugar levels aren't high enough to be called diabetes.



1.2 PURPOSE

Diabetic retinopathy is a complication of diabetes, caused by high blood sugar levels damaging the back of the eye (retina). It can cause blindness if left undiagnosed and untreated. However, it usually takes several years for diabetic retinopathy to reach a stage where it could threaten your sight. Diabetic eye screening is important as it helps to prevent sight loss. As someone with diabetes, your eyes are at risk of damage from diabetic retinopathy. Screening can detect the condition early before you notice any changes to your vision.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

The abnormal blood vessels associated with diabetic retinopathy stimulate the growth of scar tissue, which can pull the retina away from the back of the eye. This can cause spots floating in your vision, flashes of light or severe vision loss.

2.2 REFERENCES

● https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611#:~:text=Diabetic%20retinopathy%20(die%2Duh%2D,or%20only%20mild%20vision%20problems.

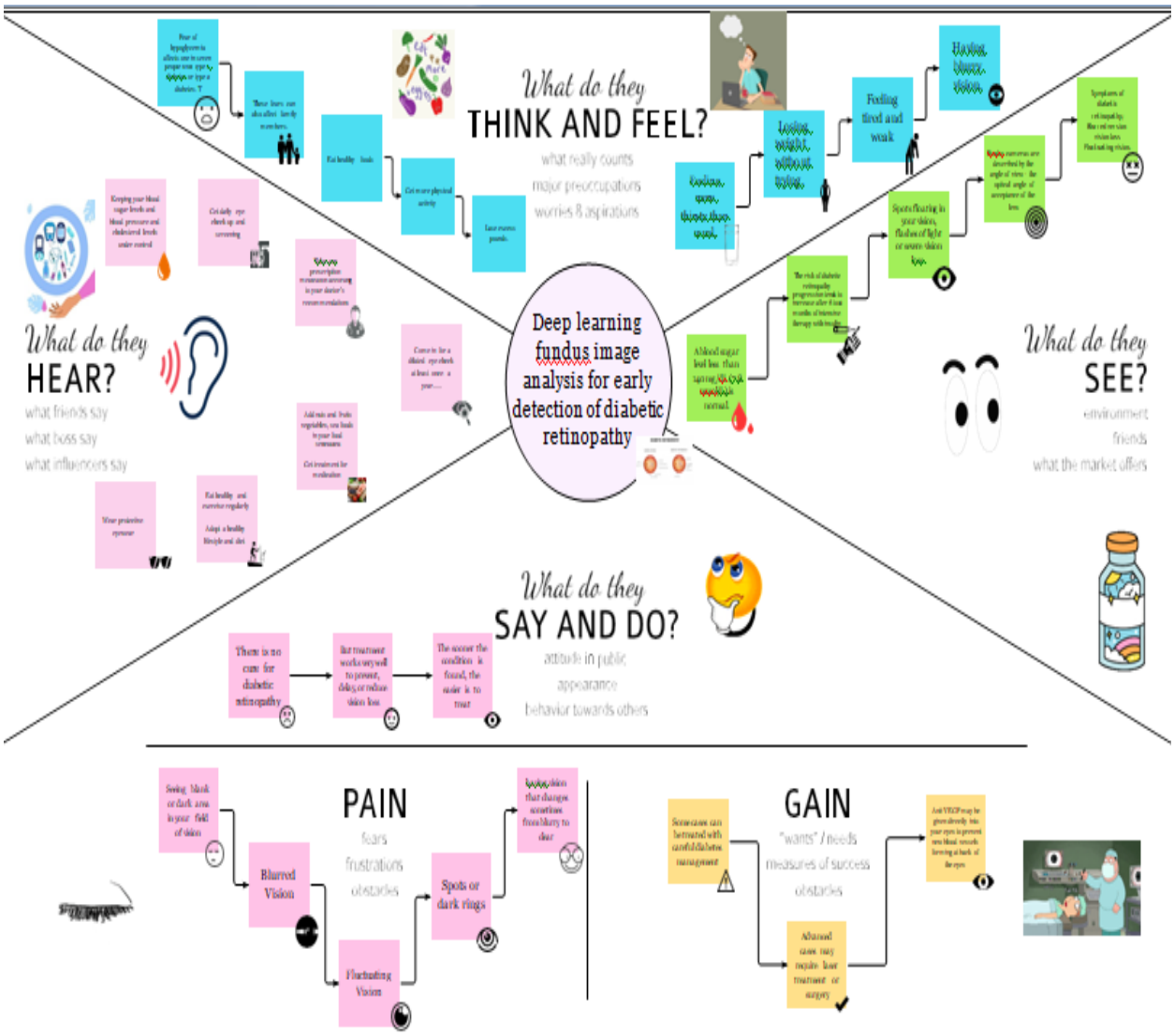
● https://www.hindawi.com/journals/joph/2018/1694187/

2.3 PROBLEM STATEMENT DEFINITION

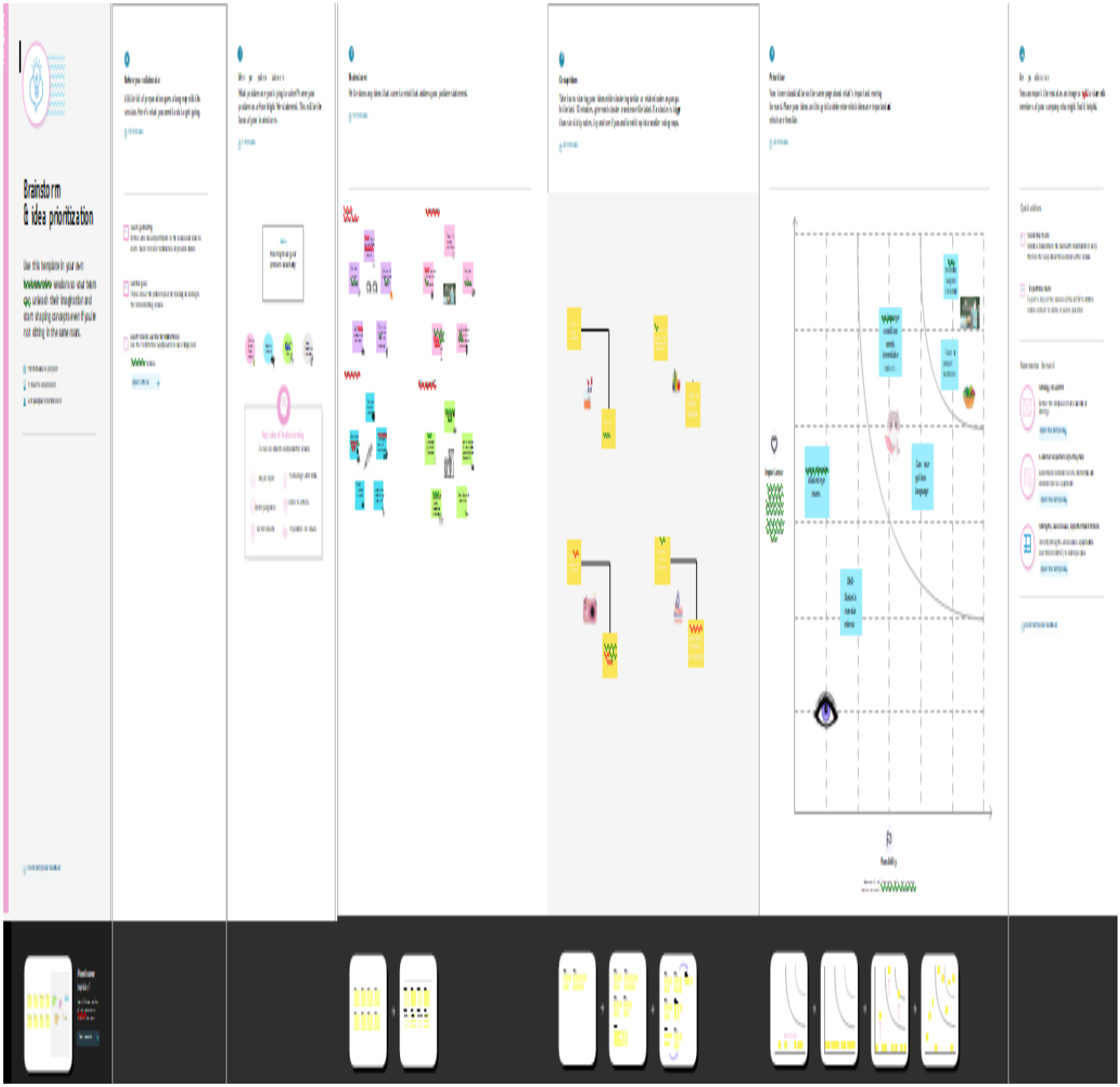
Diabetic Retinopathy(DR) is a condition that can develop in your eye if you are diagnosed with type 1 or type 2 diabetes. It damages a part of one's eye called the retina, which is the tissue that lines the back of one's eye. Signals transmitted from the retina to the brain allow one to see. High blood sugar levels due to diabetes cause damage to its blood vessels. This initially induces partial vision loss and over time one can lose their eyesight completely. The good news is that, when the symptoms of the patient are discovered in the early stages, with proper treatment one can prevent further vision loss. So it is very important for us to detect diabetic retinopathy in its early stages and prevent the patient from further damaging their eyesight. Hence, we propose a solution wherein if the fundus image, an image which displays the interior surface of the eye, is available, this problem can be easily detected by running a machine learning or deep learning model which can predict the presence of Diabetic Retinopathy in the early stages.

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAIN STROM

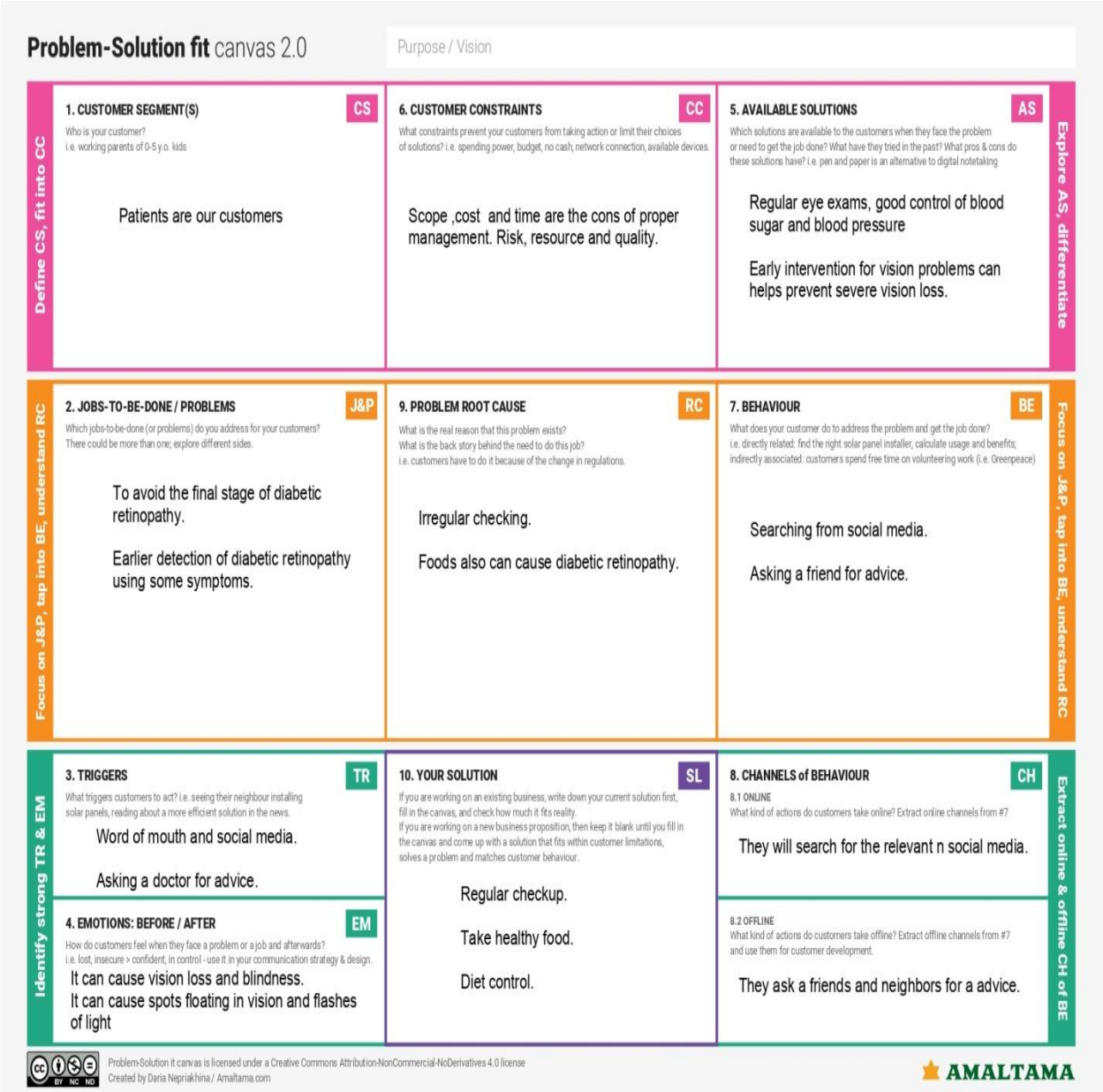


3.3 PROPOSED SOLUTION

Project team shall fill the following information in proposed solution template

S.NO	Parameter	Description
1.	Problem Statement (Problem to be solved)	Diabetic retinopathy is **a serious sight-threatening complication of diabetes**. Diabetes interferes with the body's ability to use and store sugar (glucose). The disease is characterized by too much sugar in the blood, which can cause damage throughout the body, including the eyes.
2.	Idea / Solution description	You can reduce your risk of developing diabetic retinopathy, or help stop it getting worse, by **keeping your blood sugar levels, blood pressure and cholesterol levels under control**.
3.	Novelty/Uniqueness	User can detect their disease in early stage.
4.	Social impact /customer satisfaction	Helps in preventing the loss of visibility to the needed through CSR activities or through healthcare camps.
5.	Business Model (Revenue Model)	●Can collaborate with diagnosis centers and hospitals. ●Can collaborate with government for health awareness camps.
6.	Scalability of The Solution	The aim of this research is to develop a *scalable* system to screen for a complication of diabetes that can lead to blindness: *diabetic retinopathy*.

3.4 PROBLEM SOLUTION FIT



4. REQUIRMENT ANALYSIS

4.1 FUNCTIONL REQUIRMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	User registration via form using email ID and Password
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Using the registered email ID and Password as login credentials
FR-4	Profile Dashboard	Viewing the profile, Changing Password and Previous Records
FR-5	Checking For Diabetic Retinopathy	Uploading the FUNDUS Image of the eye to predict the disease
FR-6	User Tracking	Maintaining the Record of how far the eye has been affected
FR-7	Feedback & Support	Collection of feedback whether the results were accurate enough

4.2 NON-FUNCTIONAL REQUIRMENT

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• The system should have a quality attribute that must be able to assess the ease of the usage of the UI. • The system should not expect any technical prerequisites from the user's side.
NFR-2	**Security**	• User details and login credentials should be safe and secure. • The confirmation of a valid user is required for authentication.
NFR-3	**Reliability**	• Portable and cross-platform independent. • Easy to use and flexible.
NFR-4	**Performance**	• The system should handle the traffic efficiently and service requests while consuming less bandwidth. • The accuracy of the result of a measurement, calculation, or specification should be dependent on the datasets. • The page should not take a lot of time to load the contents and display them.
NFR-5	**Availability**	• The version of the application should be available even at the time of maintenance and updating. • The system should run 24 hours a day, 7 days a week [24/7 available].
NFR-6	**Scalability**	• The application should be in the way of adding new functionalities or modules without affecting the existing functionalities.

5. PROJECT DESIGN

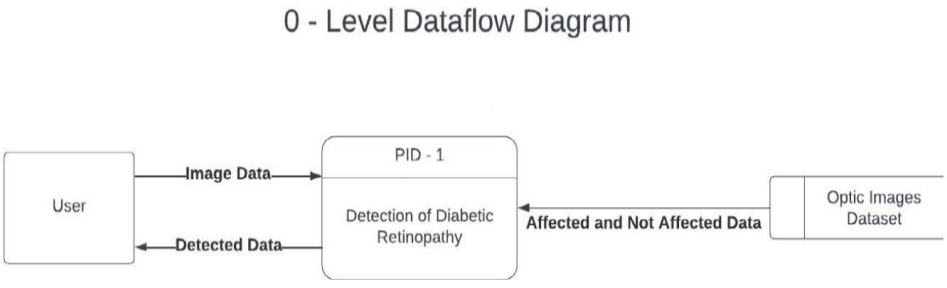
5.1 DATA FLOW DIAGRAM

Project Design Phase-II
Data Flow Diagram & User Stories

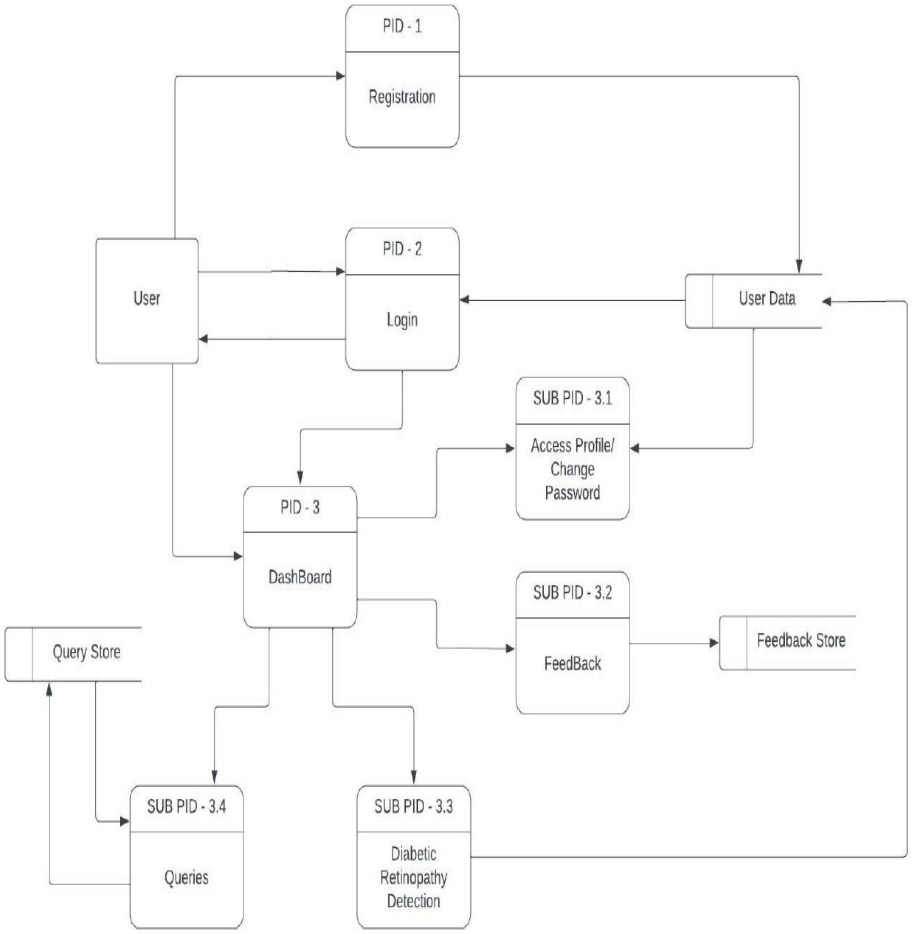
Date	29October 2022
Team ID	PNT2022TMID49992
Project Name	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Maximum Marks	4 Marks

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

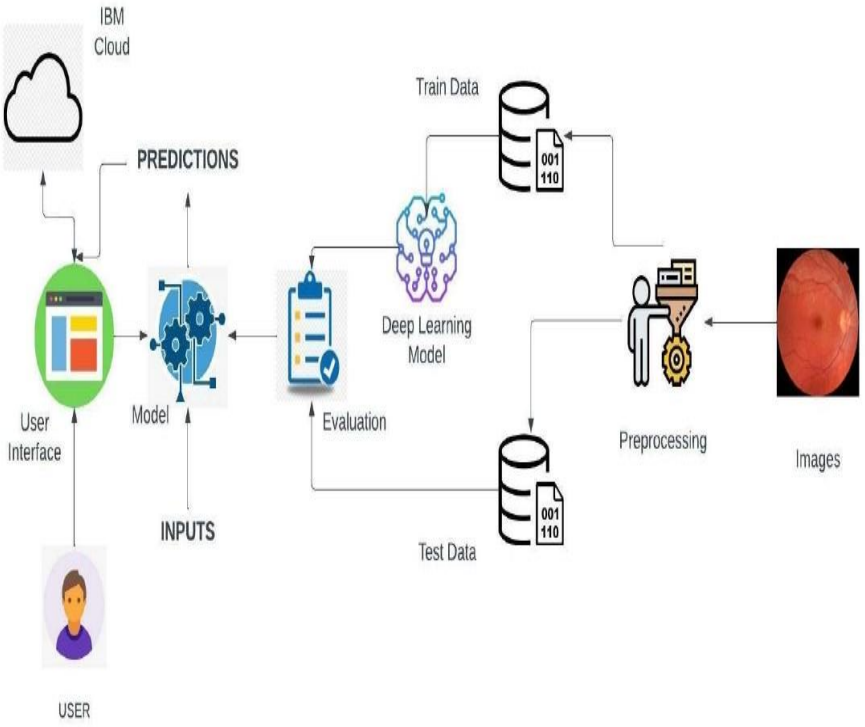


1 - Level Dataflow Diagram



5.2 SOLUTION & TECHNOLOGICAL ARCHITECTURE

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



Design Phase II

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	MUSN-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
		MUSN-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
		MUSN-3	Validation of the user can be done directly using email or OTP.	Account validated and got access to profile dashboard.	Medium	Sprint-1
	Login	MUSN-4	Enter the username and password to login to the application.	Right account credentials should be entered.	High	Sprint-1
Customer (Web user)	Dashboard	WUSN-1	Users can access their information like age, gender, previous history etc..	Monitoring Diabetes level	High	Sprint-1
		WUSN-2	Users can upload the FUNDUS image and obtain the results.	Prediction of DR	High	Sprint-2
Customer Care Executive	Overview	CCE-1	As a customer, I can view the data in the graphical representation.	The level of diabetes as well as the DR are analysed	High	Sprint-2
	Database Analysis	CCE-2	As an operator, I should be provided with the image.	Manual detection of DR	High	Sprint-2
Support Team	Support	ST-1	Responds to user queries via telephone or email.	Queries can be raised in certain situations.	Medium	Sprint-3
Administrator	User Control and help	AN-1	Helping users in case of any technical issues faced by them like forgetting the password, collecting feedback and improving the UI.	Maintaining the application's functionality and business	Medium	Sprint-4

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 applicationby entering my email or phone number and password, and confirming my password.	10	High	Narayani
Sprint-1	Dashboard	USN-2	As a user, I will Redirect to the dashboardafter registration which shows the importance of DR.	10	Medium	Narayani& Ameenal
Sprint-2	Login	USN-3	As a user, I can log into the application byentering Login credentials	5	High	Ameenal
Sprint-2	Upload Images	USN-4	As a user, I should be able to upload the imageof eye Retina.	10	High	Ameenal& Narayani
Sprint-2	Upload	USN-5	As a user, based on my requirement I cannavigate through the dashboard.	5	Medium	Ameenal& Narayani
Sprint-3	Train the model	Task 1	As a developer, the dataset will be uploadedand trained by developed algorithm.	20	High	Bharathi& Kalaivani
Sprint-4	Testing & Evaluation	Task 2	As a developer, we tested the trained modelusing the provided dataset and model will beevaluated for accurate results.	10	High	Kalaivani
Sprint-4	Display predicted result	USN-6	As a user, I can view the predicted result inthe dashboard.	10	High	Bharathi& Kalaivani

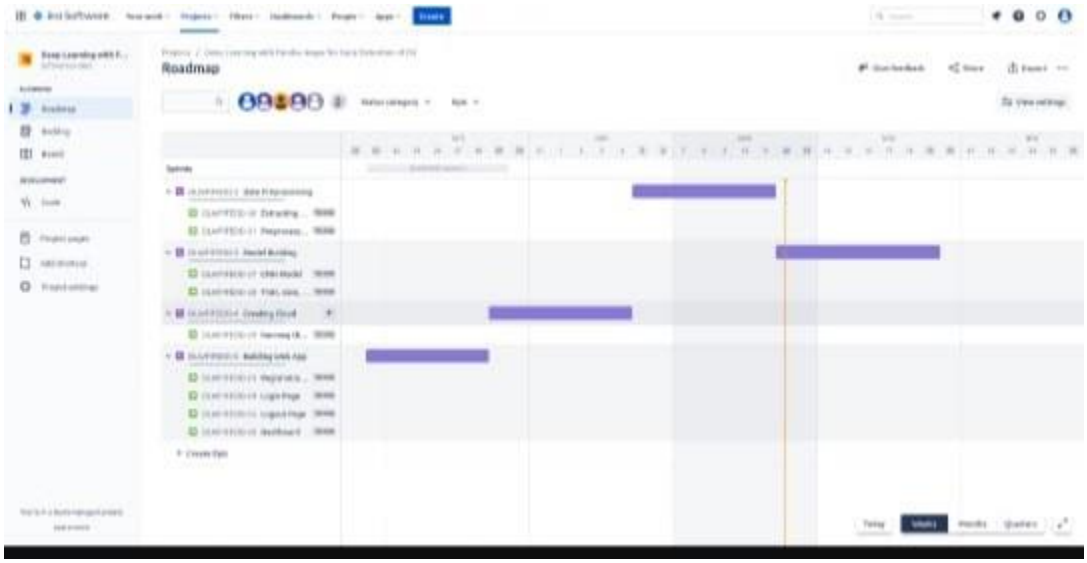
6.2 SPRINT DELIVERY SCHEDULE

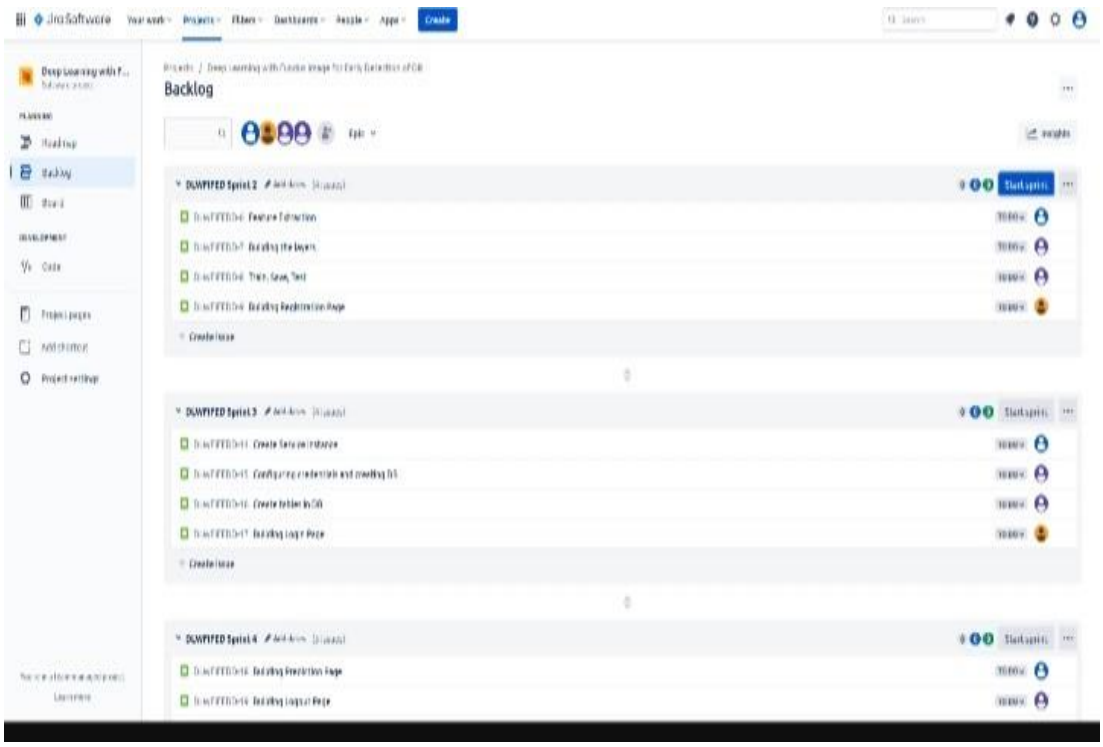
Sprint	Total story point	Duration	Sprint StartDate	Sprint End Date (Planned)	Story Points Completed (ason Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct2022	29Oct 2022	20	29Oct 2022
Sprint-2	20	6 Days	5 Oct 2022	5 Nov2022	20	5 Nov2022
Sprint-3	20	6 Days	12 Nov2022	12 Nov2022	20	12 Nov2022
Sprint-4	20	6 Days	19 Nov2022	19 Nov2022	20	19 Nov2022

AV = SPRINT DURATIO/VELOCITY = 20/10 = 2

AV=20/6=3.33points per day.

6.3 REPORT FROM JIRA





7. CODING AND SOLUTIONING

7.1 Feature

We have developed a website which authenticates users and help them upload and check the seriousness of the diabetics. We have developed a multilayer deep convolutional nueral network that classifies the user image of a eye to which extense has the disease diabetics has been affected. The model will classify the images into 5 categories of diabetics and report them on asking for prediction. We have also developed a messaging service for receiving message for the type of diabetics.

8. TESTING

8.1 TEST CASES

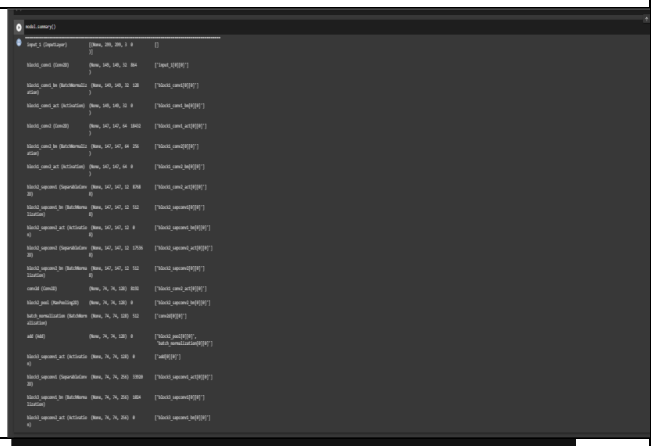
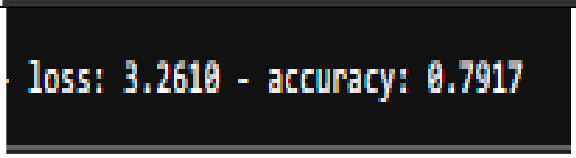
Resolution	Severity 1	Severity 2	Severity 3	Severity4	Subtotal
By Design	5	4	2	3	14
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	9	2	4	15	30
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	17	14	13	21	65

8.2 USER ACCEPTANCE TESTING

Section	Total Cases	Not Tested	FAIL	Pass
Print Engine	9	0	0	9
Client Application	45	0	0	45
Security	2	0	0	2
Out-source Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 Performance Metrics

S.NO	PARAMETER	VALUES	SCREENSHOT
1	Model Summary	Total params:21,885,485 Trainable params:1,024,005 Non-trainable params:20,861,480	
	Accuracy	Training Accuracy– 0.7917 Validation Accuracy– loss3.2610	
	Confidence Score(Only Yolo Projects)	Class Detected Confidence score	- - - -

10. ADVANTAGES AND DISADVANTAGES

10.1 ADVANTAGES

- Deep learning is well-suited for image analysis tasks. This is because deep learning algorithms can automatically learn features from images, which is essential for accurate image analysis.

- Deep learning is efficient at handling large amounts of data. This is important for medical image analysis, as medical images are often very large.

- Deep learning is scalable. This means that it can be used to train models on very large datasets, which is important for medical image analysis tasks where data is often limited.

- Deep learning is able to learn from data with little supervision. This is important for medical image analysis, as often there is limited labeled data available.

- Deep learning is robust. This means that it is less likely to over fit to the data, which is important for medical image analysis where data is often limited.

10.2 DISADVANTAGES

- There are several disadvantages of deep learning for early detection of diabetic retinopathy.
- One disadvantage is that deep learning requires a large amount of data to train the models.
- This can be a challenge for researchers who do not have access to a large dataset.
- Another challenge is that deep learning models can be very complex, which can make them difficult to interpret.
- Finally, deep learning models can be computationally intensive, which can make them difficult to deploy in resource-limited settings.

11. CONCLUSIONS

Diabetic retinopathy (DR) is a leading cause of blindness in the United States. Early detection and treatment of DR is critical to preventing vision loss. However, DR is often asymptomatic in its early stages, making it difficult to detect. Deep learning (DL) is a type of artificial intelligence that can be used to automatically detect patterns in data. DL has been shown to be effective for detecting DR in images of the retina. In this study, a DL algorithm was used to automatically detect DR in fundus images. The algorithm was able to accurately detect DR in early stages, before it is symptomatic. This could potentially lead to earlier diagnosis and treatment of DR, which could help to prevent vision loss.

12. FUTURE SCOPE

There is a great potential for deep learning in fundus image analysis for early detection of diabetic retinopathy. However, there are a few challenges that need to be addressed. First, the current data sets are small and lack diversity. Second, the images are often low quality and need to be pre-processed before they can be used for deep learning. Third, the ground truth labels for the images are often not available. Finally, the current deep learning models are not able to generalize well to real-world data.

13. APPENDIX

```
import numpy as np
import os
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.inception_v3 import preprocess_input
from flask import Flask, request,flash, render_template, redirect,url_for
from cloudant.client import Cloudant
from twilio.rest import Client
model = load_model(r"Updated-xception-diabetic-retinopathy.h5")
app = Flask(__name__)
app.secret_key="abc"
```

```
app.config['UPLOAD_FOLDER'] = "User_Images"
# Authenticate using an IAM API key

client = Cloudant.iam('08bcbaf0-260b-48e0-abdb-08db348afcf2-bluemix',
                      'yhZfUubpS3vS1vEKZSS37teD6IAUi8oLynOCQLIwnQsa',
connect=True)
# Create a database using an initialized client
my_database = client.create_database('my_database')
if my_database.exists():
print("Database '{0}' successfully created.".format('my_db'))
# default home page or route

user = ""

@app.route('/')
def index():
    return render_template('index.html', pred="Login", vis ="visible")

@ app.route('/index')
def home():
    return render_template("index.html", pred="Login", vis ="visible")


# registration page
@ app.route('/register',methods=["GET","POST"])
def register():
```

```
    if request.method == "POST":
        name =  request.form.get("name")
        mail = request.form.get("emailid")
        mobile = request.form.get("num")
pswd = request.form.get("pass")
        data = {
            'name': name,
            'mail': mail,
            'mobile': mobile,
            'psw': pswd
        }
        print(data)
        query = {'mail': {'$eq': data['mail']}}
        docs = my_database.get_query_result(query)
        print(docs)
        print(len(docs.all()))
        if (len(docs.all()) == 0):
url = my_database.create_document(data)
            return render_template("register.html", pred=" Registration Successful , please login using your details ")
        else:
            return render_template('register.html', pred=" You are already a member , please login using your details ")
    else:
        return render_template('register.html')
```

```
@ app.route('/login', methods=['GET','POST'])
def login():
    if request.method == "GET":
        user = request.args.get('mail')
passw = request.args.get('pass')
print(user, passw)
 query = {'mail': {'$eq': user}}
        docs = my_database.get_query_result(query)
        print(docs)
        print(len(docs.all()))
        if (len(docs.all()) == 0):
            return render_template('login.html', pred="")
        else:
            if ((user == docs[0][0]['mail'] and passw == docs[0][0]['psw'])):
flash("Logged in as " + str(user))
                return render_template('index.html', pred="Logged in as "+str(user), vis
="hidden", vis2="visible")
            else:
                return render_template('login.html', pred="The password is wrong.")
    else:
        return render_template('login.html')


@ app.route('/logout')
```

```
def logout():
    return render_template('logout.html')


@app.route("/predict",methods=["GET", "POST"])
def predict():
    if request.method == "POST":
        f = request.files['file']
        # getting the current path 1.e where app.py is present
        basepath = os.path.dirname(__file__)
        #print ( " current path " , basepath )
        # from anywhere in the system we can give image but we want that
filepath = os.path.join(str(basepath), 'User_Images', str(f.filename))
        #print ( " upload folder is " , filepath )
f.save(filepath)
img = image.load_img(filepath, target_size=(299, 299))
        x = image.img_to_array(img)  #ing to array
        x = np.expand_dims(x, axis=0)  # used for adding one more dimension
        #print ( x )
img_data = preprocess_input(x)
        prediction = np.argmax(model.predict(img_data), axis=1)
        index = [' No Diabetic Retinopathy ', ' Mild NPDR ',
                 ' Moderate NPDR ', ' Severe NPDR ', ' Proliferative DR ']
        result = str(index[prediction[0]])
        print(result)
account_sid = 'AC8e0f2f5263d71c8f630a6486779cf08b'
```

```
auth_token = '30b489873afb3c47340070eabd6bfb15'

        client = Client(account_sid, auth_token)

        '" Change the value of 'from' with the number
        received from Twilio and the value of 'to'
        with the number in which you want to send message."'
        message = client.messages.create(
                                from_='+16075363206',
                                body ='Results: '+ result,
                                to ='+919445979800'
                            )
        return render_template('prediction.html', prediction=result, fname = filepath)
    else:
        return render_template("prediction.html")

if __name__ == "__main__":
app.debug = True
app.run()
```

cloud.ipynb:-

```
from cloudant.client import Cloudant
client=Cloudant.iam(a89ef88f-e4c3-4d63-b816-bdc4cb267519-bluemix',
p5Hd4TxZ4Ab22qD5ldcAllNTLx4Ons3nZZJj0U4YgBCi ',connect=True)
my_database=client.create_database('my-database')
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<!-- CSS only -->
<link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fz
T"
crossorigin="anonymous"
  />
<!-- JavaScript Bundle with Popper -->
<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi
466C8"
crossorigin="anonymous"
></script>
<style>
      #navbarRight {
          margin-left: auto;
```

```
        padding-right:10px;
    }
.navbar-brand{
        padding-left:15px;
    }
</style>
<title>DR Predcition</title>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-dark">
<div>
<a class="navbar-brand" href="#" style="color:aliceblue">Diabetic Retinopathy Classification</a>
</div>
    {{msg}}
<div class="navbar-collapse collapse w-100 order-3 dual-collapse2" id="navbarNav">
<ul class="navbar-nav mr-auto text-center" id="navbarRight">
<li class="nav-item active">
<a class="nav-link" href="index" style="color: aliceblue;">Home </a>
</li>
<li class="nav-item" style="visibility:{{ vis2 }}">
<a class="nav-link" href="predict" style="color: aliceblue;">Prediction</a>
</li>
<li class="nav-item">
<a class="nav-link" href="login" style="color: aliceblue;">{{pred}}</a>
```

```
</li>
<li class="nav-item" style="visibility:{{ vis }}">
<a class="nav-link" href="register" style="color: aliceblue;">Register</a>
</li>
</ul>
</div>
</nav>
<br><br>
<div class="jumbotron container">
<h1 class="display-4">Diabetic Retinopathy</h1>
<p class="lead">Diabetic retinopathy is a diabetes complication that affects eyes. It's caused by damage to the blood vessels of the light-sensitive tissue at the back of the eye (retina).

At first, diabetic retinopathy might cause no symptoms or only mild vision problems. But it can lead to blindness.

The condition can develop in anyone who has type 1 or type 2 diabetes. The longer you have diabetes and the less controlled your blood sugar is, the more likely you are to develop this eye complication.</p>
<hr class="my-4">
<div class="d-flex justify-content-center">
<img style="width:70vw;" src="static/diabetic-retinopathy-home.jpg">
</div>
</div>
</body>
</html>
```

login.html:-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<!-- CSS only -->
<link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fz
T"
crossorigin="anonymous"
  />
<!-- JavaScript Bundle with Popper -->
<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi
466C8"
crossorigin="anonymous"
></script>
<style>
     #navbarRight {
```

```
        margin-left: auto;
        padding-right:10px;

 }
.navbar-brand{
        padding-left:15px;
      }
</style>
<title>DR Predcition</title>
</head>
<form action="",method='POST'>
</body>
</html>
```

Github link demo vdo

https://github.com/IBM-EPBL/IBM-Project-41124-1660639592/blob/main/Final%20Deliverables/Demo%20video/Demo%20Video.mp4

Github link

https://github.com/IBM-EPBL/IBM-Project-41124-1660639592