

**Deep Learning Fundus image analysis for early detection of Diabetic
Retinopathy(PNT2022TMID06905)**

Project Report

Table of content

1. INTRODUCTION

1. Project Overview
2. Purpose

2. LITERATURE SURVEY

1. Existing problem
2. References
3. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

4. REQUIREMENT ANALYSIS

1. Functional requirement
2. Non-Functional requirements

5. PROJECT DESIGN

1. Data Flow Diagrams
2. Solution & Technical Architecture
3. User Stories

6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule
3. Sprint Reports

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

1. Feature 1
2. Feature 2
3. Database Schema (if Applicable)

8. TESTING

1. Test Cases
2. User Acceptance Testing

9. RESULTS

1. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Link

Deep Learning Fundus image analysis for early detection of Diabetic Retinopathy(PNT2022TMID06905)

Project Report

1. INTRODUCTION

1.1 Project Overview:

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.

1.2 Purpose:

Transfer learning has become one of the most common techniques that has achieved better performance in many areas, especially in medical image analysis and classification. We used Transfer Learning techniques like Inception V3, Resnet50, Xception V3 that are more widely used as a transfer learning method in medical image analysis and they are highly effective.

2. LITERATURE SURVEY:

2.1 Existing problem:

The conventional approach for detection of Diabetic Retinopathy is manual diagnosis by specialists. This manual diagnosis process of Diabetic Retinopathy retina fundus images by ophthalmologists is time, effort, and cost-consuming and prone to misdiagnosis unlike computer-aided diagnostic systems. Recently deep learning has become one of the most common techniques that has achieved better performance in many areas, especially in the medical image analysis and classification.

2.2 References:

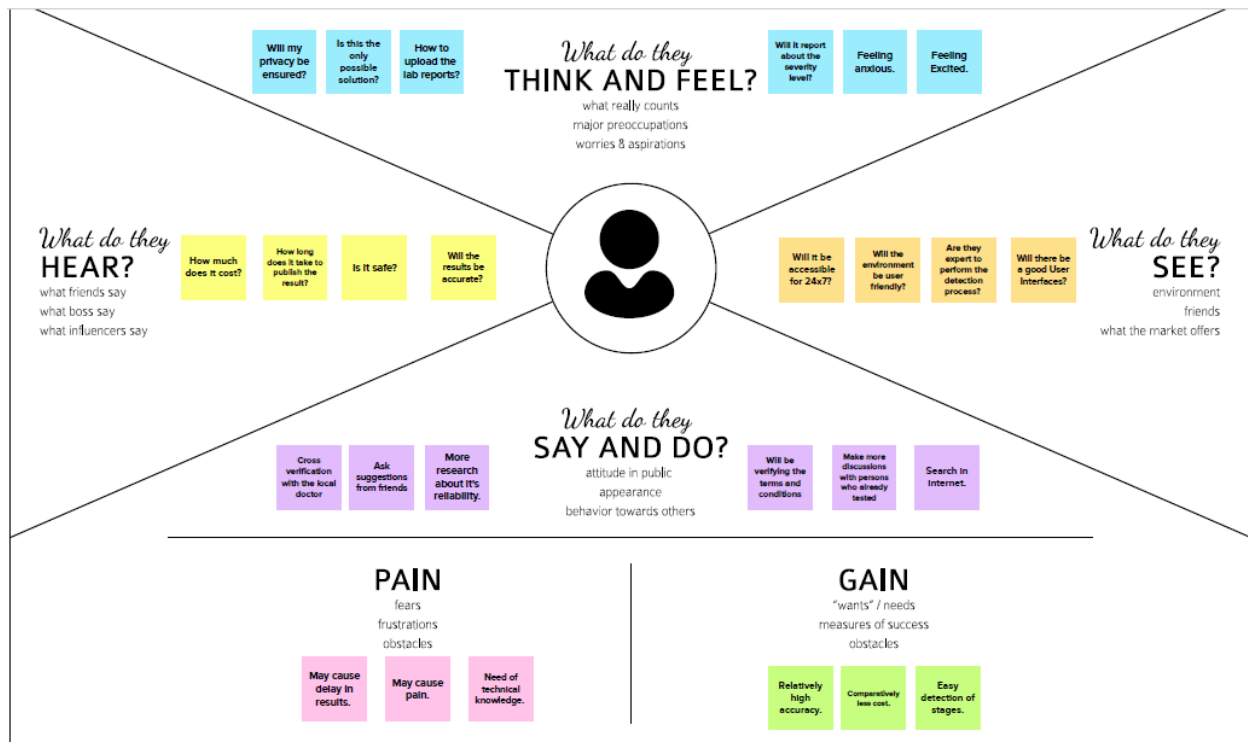
- [https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611#:~:text=Diabetic%20retinopathy%20\(die%20Duh%2D,or%20only%20mild%20vision%20problems.](https://www.mayoclinic.org/diseases-conditions/diabetic-retinopathy/symptoms-causes/syc-20371611#:~:text=Diabetic%20retinopathy%20(die%20Duh%2D,or%20only%20mild%20vision%20problems.)
- <https://www.hindawi.com/journals/joph/2018/1694187/>
- <https://www.digitaldiagnostics.com/>

2.3 Problem Statement Definition:

Approximately four hundred and twenty million people worldwide have been diagnosed with diabetes mellitus. The prevalence of this disease has doubled in the past 30 years and is only expected to increase, particularly in Asia. Of those with diabetes approximately one-third are expected to be diagnosed with diabetic retinopathy (DR), a chronic eye disease that can progress to irreversible vision loss. Early detection which is critical for good prognosis, release on skilled readers and is both labour and time-intensive. Automated techniques for diabetic retinopathy diagnoses are essential to solving these problems.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

PROBLEM 1

Time efficient and cost effective means to detect Diabetic Retinopathy

PROBLEM 2

Remote detection and Instant results of diagnosis

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

🕒 10 minutes

TIP



You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

SUDHA R

Humphrey Matrix Threshold testing	CT scans can be used to detect early stages	Optomap retinal exam provide image with 80% covering of retina
Detecting and Diagnosis using Deep Learning	Physical self detection of visual defects	

GHAYATHRI DEVI V

Using autonomous systems to detect DR	Using AI powered Robots to detect early stages of DR	CAD systems are employed to differentiate a normal eye and DR affected eye
MRI scans can be used	Using Deep learning to detect DR	

VINETHA R

Comprehensive Dilated eye exam	Longer Dilatation duration and high light levels required at any regular interval time	Fluorescein Angiography
Frequency Doubling Technology	Diagnosis using AI	

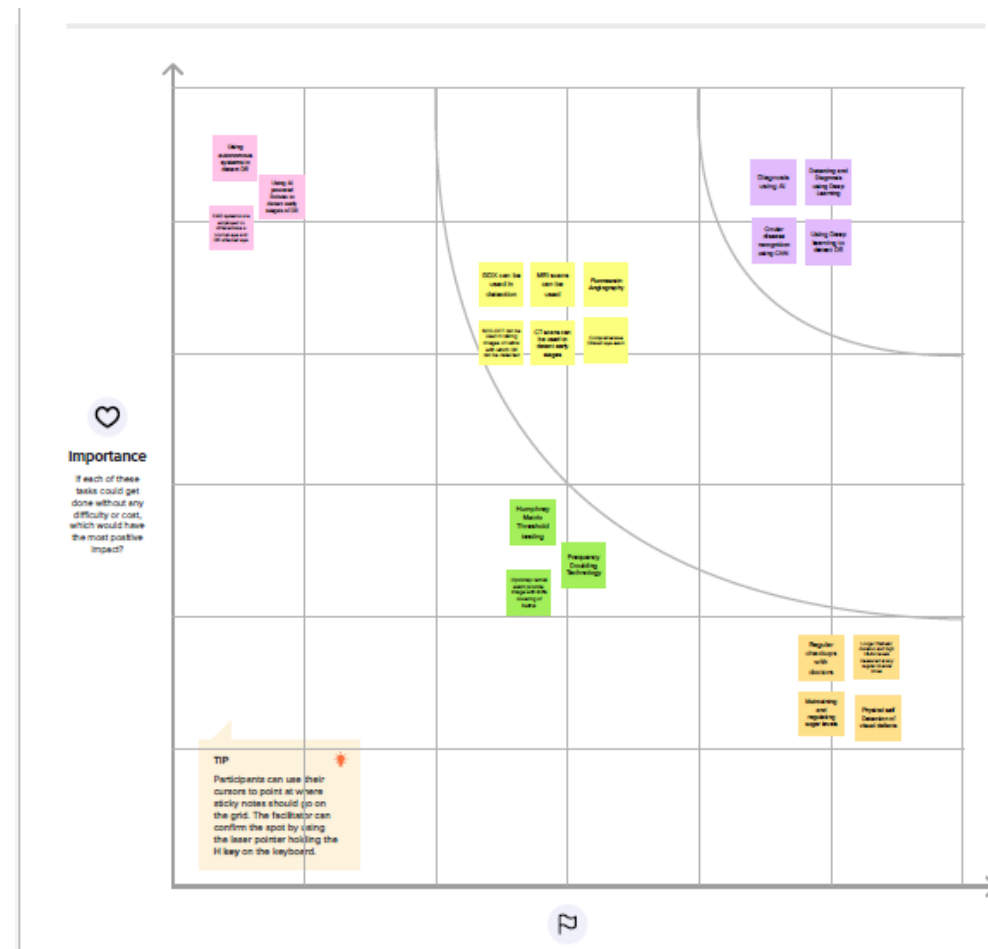
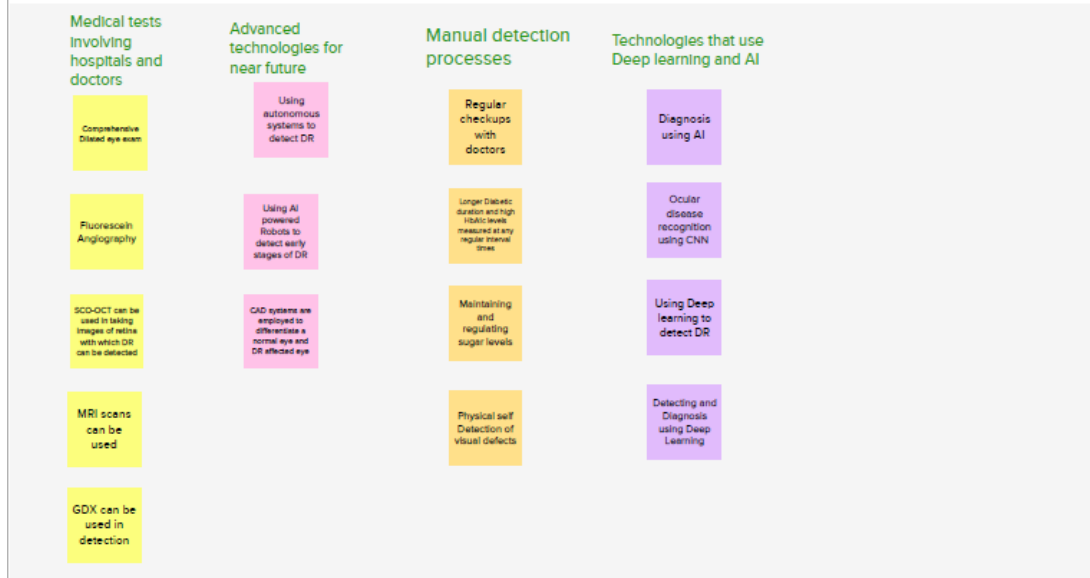
NANDHINI M V

Regular checkups with doctors	Ocular disease recognition using CNN	Maintaining and regulating sugar levels
SOD-ICT can be used in taking images of retina with which DR can be detected	GDX can be used in detection	

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

🕒 20 minutes



3.3.Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Diabetic retinopathy is one of the most threatening complications of diabetes that leads to permanent blindness if left untreated. One of the essential challenges is early detection, which is very important for treatment success. Unfortunately, the exact identification of the diabetic retinopathy stage is notoriously tricky and requires expert human interpretation of fundus images. Simplification of the detection step is crucial and can help millions of people. Convolutional neural networks (CNN) have been successfully applied in many adjacent subjects, and for diagnosis of diabetic retinopathy itself. However, the high cost of big labeled datasets, as well as inconsistency between different doctors, impeded the performance of these methods. In this paper, we propose an automatic deep-learning-based method for stage detection of diabetic retinopathy by single photography of the human fundus. Additionally, we propose the multistage approach to transfer learning, which makes use of similar datasets with different labeling. The presented method can be used as a screening method for early detection of diabetic retinopathy with sensitivity.
2.	Idea / Solution description	Our aim is to identify retinopathy using five different diabetic retinopathy. Moreover, we present what pre-processing and regularization steps to the images needs to be done for the good functionality of the deep learning system and investigate systematically how the size with much smaller number of images used in training affects its performance.

3.	Novelty / Uniqueness	We will be using transfer learning. This has become one of the most common techniques that has achieved better performance in many areas, especially in medical image analysis and classification. We used Transfer Learning techniques like Inception V3, Resnet50, Xception V3 that are more widely used as a transfer learning method in medical image analysis and they are highly effective.
4.	Social Impact / Customer Satisfaction	Using this system, we can easily detect the stages early, the doctors can able to treat the patient effectively and also reduce the complications. Therefore, this review identifies the need for improved outcome measures to provide valid, meaningful measurement of the social and emotional impact of diabetic retinopathy and discusses potential directions for future research such as item banking and computer adaptive testing.
5.	Business Model (Revenue Model)	This system would be used by both the doctors as well as the patients. It would reduce the time for the doctors to detect the stages. Demand for diabetic retinopathy is increasing with the complication related eye which is likely to treat with various eye injections and laser treatment for the disease are anticipated to increase the diabetic retinopathy market share in the forecast period from 2022 to 2032". The systems will be valuable in improving both screening of individuals with diabetes and communication and discussion among individuals caring for these patients.
6.	Scalability of the Solution	The system, being dynamically and modularly developed, allows for much modification and large scalable operations. More data when made available can be processed and produce efficient results. This system is easily and efficiently scalable.

3.3 Problem Solution Fit:

Project Title: Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID06905

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Who is your customer? I.e. working parents of 0-5 y.o. kids People with extreme hyperglycemia probably around the age of 30 and above.	6. CUSTOMER CONSTRAINTS CC What constraints prevent your customers from taking action or limit their choices of solutions? I.e. spending power, budget, no cash, network connection, available devices. The constraints may include basic technical knowledge, network issues, need of computational resources.	5. AVAILABLE SOLUTIONS AS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? I.e. pen and paper is an alternative to digital They might seek the nearby cybercafé for network instability and for technical support which might turn into great problem because of attackers.	Explore AS, differentiate
------------------------	---	---	--	---------------------------

Focus on J&P, tap into BE, understand RC	2. JOBS-TO-BE-DONE / PROBLEMS J&P Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides. -> Delayed results might be an issue -> Need of a basic technical knowledge -> Need of fundus image in-hand prior performing this test. -> People may feel that their privacy might get lost.	9. PROBLEM ROOT CAUSE RC What is the real reason that this problem exists? What is the back story behind the need to do this job? I.e. customers have to do it because of the change in regulations. -> Rapid modernization has a leading hand in this change -> Contradictions in the diagnosis of doctors also has caused this change.	7. BEHAVIOUR BE What does your customer do to address the problem and get the job done? I.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (I.e. Greenpeace) -> Seek some knowledgeable person or try contacting the technical team for guidance. -> Refer to internet in this regard. -> Try changing the network for good performance. -> May seek clinics nearby so that there will be direct consultation of doctors.	Focus on J&P, tap into BE, understand RC
--	--	---	---	--

Identify strong TR & EM	3. TRIGGERS TR What triggers customers to act? I.e. seeing their neighbors installing solar panels, reading about a more efficient solution in the news. -> It might be the cost efficient and time saving solution for this problem -> Suggestion of doctor might also be a triggering factor -> Seeing patients with similar issues getting quick and accurate response.	10. YOUR SOLUTION SL If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behavior. -> Clear explanation via documentation and user-interface will be provided. -> An exclusive comment section for resolving the queries of the users. All queries will be responded within 24 hours. -> Collaborate with hospitals for guidance of patients.	8. CHANNELS OF BEHAVIOUR CH 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7 -> Refer to internet in this regard -> Try changing the network for good performance. 8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development. -> Seek some knowledgeable person or try contacting the technical team for guidance. -> May seek clinics nearby so that there will be direct consultation of doctors.	Identify strong TR & EM
	4. EMOTIONS: BEFORE / AFTER EM How do customers feel when they face a problem or a job and afterwards? I.e. lost, insecure > confident, in control - use it in your communication strategy & design. Time and effort consuming > Time saving and effort less More dependency on doctors > Less Dependency on doctors			

4. REQUIREMENT ANALYSIS

4.1.Functional requirements:

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	Identifying the population eligible for scanning	Determine the group to be screened based on best evidence and use registers to make sure people's details are collected and uptodate.
FR-3	Testing	Conduct screening using recommended methods
FR-4	Reporting of results	Reporting the results as they are in the screening program.
FR-5	Diagnosis	Diagnose true cases and identify false positives
Fr-6	Reporting of outcomes	Collect ,analyze and report on outcomes to identify false negatives and improve effectiveness and cost effectiveness of screening program.

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	Provides novel results for five different screening and clinical grading systems for diabetic retinopathy including state-of-the-art results for accurately classifying images according to clinical five -grade Diabetic retinopathy
NFR-2	Security	Deep Learning using AI can be more precise around sensitive organs and tissues, reduce blood loss, risk Of infection, and pain during detection/screening
NFR-3	Reliability	The ability of Deep Learning is to perform pattern Recognition by creating complex relationships based on input data and then comparing it with Performance standards is a big step.

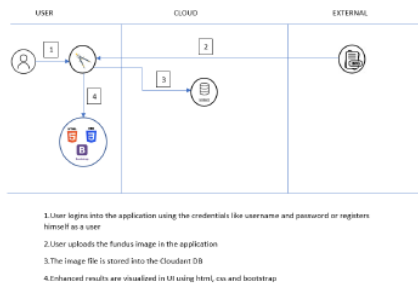
5. PROJECT DESIGN

5.1 Data Flow Diagrams:

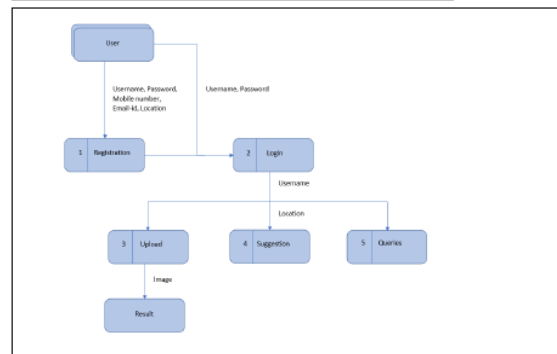
Data Flow Diagrams:

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.

Example: (Simplified)



DFD Level 0 (Industry Standard)



5.2 Solution & Technical Architecture:

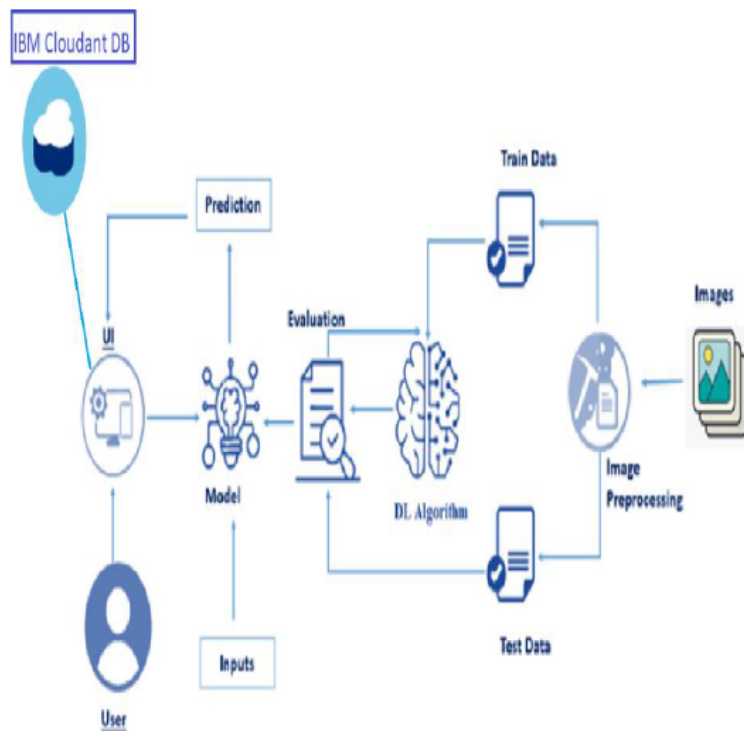


TABLE-1 : COMPONENTS & TECHNOLOGIES

S.No	Component	Description	Technology
1.	User Interface	Web UI	HTML, CSS, JS, Python.
2.	Application Logic-1	Data Preprocessing	Keras, Tensorflow, Numpy - (Importing Essential Libraries)
3.	Application Logic-2	CNN Model Creating	Keras, Tensorflow, Numpy - (Importing Essential Libraries)
4.	Application Logic-2	Web Application (UI)	Flask
5.	Database	Images (Jpeg, PNG, Jpg, etc..)	Uploads Folder !
6.	File Storage	File storage requirements (only if necessary)	IBM Block Storage / Google Drive (Depends On Preference)
7.	External API	Keras	Image Processing API.
8.	Deep Learning Model	Xception architecture	Pretrained convolution neural network model that is 18 layers deep
9.	Infrastructure (Server / Cloud)	Application Deployment on Webserver	Flask—a Python WSGI HTTPserver.

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
User	Registration	USN-1	As a user, I should be able to register myself with username, password, mobile number, email-id, location	Pop up showing registered	High	Sprint-1
	Login	USN-2	As a user, I should be able to register myself and should have forget password for recovery		High	Sprint-1
		USN-3	As a user, I can login into my application using my username and password		High	Sprint-1
		USN-4	As a user, I should be able to post my queries in the application		High	Sprint-2
	Dashboard	USN-5	As a user, I should be able to modify the credentials given by me like my location to get correct suggestions of hospitals nearby		Medium	Sprint-2
Administrator	Database	USN-6	As an administrator I should be able to update the contact details and addresses of hospitals		Medium	Sprint-2
		USN-7	As an administrator I should be able to read and respond to all the user queries from comment section		Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning and Estimation:

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I should be able to register myself with username, password, mobile number, email-id, location	5	High	Sudha R Nandhini M V
Sprint-2	Login	USN-2	As a user, I should be able to register myself and should have forget password for recovery	5	High	Sudha R Nandhini M V
Sprint-1		USN-3	As a user, I can login into my application using my username and password	5	High	Sudha R Nandhini M V
Sprint-2		USN-4	As a user, I should be able to post my queries in the application	7	High	Sudha R Nandhini M V
Sprint-4	Dashboard	USN-5	As a user, I should be able to modify the credentials given by me like my location to get correct suggestions of hospitals nearby	5	Medium	Vinetha R Ghayathri Devi V
Sprint-4	Database	USN-6	As an administrator I should be able to update the contact details and addresses of hospitals	5	Medium	Vinetha R Ghayathri Devi V
Sprint-3		USN-7	As an administrator I should be able to read and respond to all the user queries from comment section	5	Medium	Vinetha R Ghayathri Devi V
Sprint-3	User Interface (Detection)	USN-8	As a user, I should be able to upload the image of my retina and should get accurate results of the diagnosis	9	High	Vinetha R Ghayathri Devi V

6.2 Sprint Delivery Schedule:

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

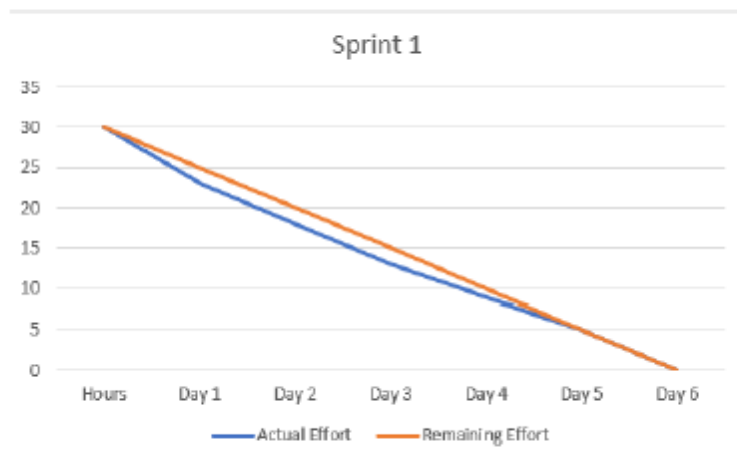
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	10	6 Days	24 Oct 2022	29 Oct 2022	10	30 Oct 2022
Sprint-2	12	6 Days	31 Oct 2022	05 Nov 2022	12	06 Nov 2022
Sprint-3	14	6 Days	07 Nov 2022	12 Nov 2022	14	15 Nov 2022
Sprint-4	10	6 Days	14 Nov 2022	19 Nov 2022	10	16 Nov 2022

6.3 Reports:

Sprint 1:

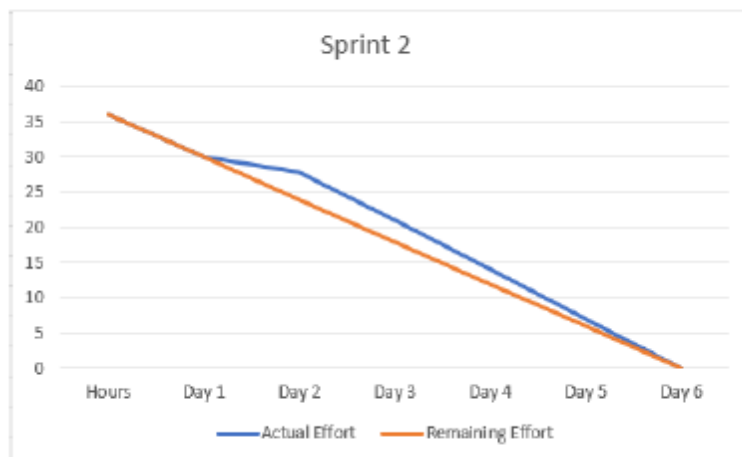
Burn Down charts:

Sprint-1



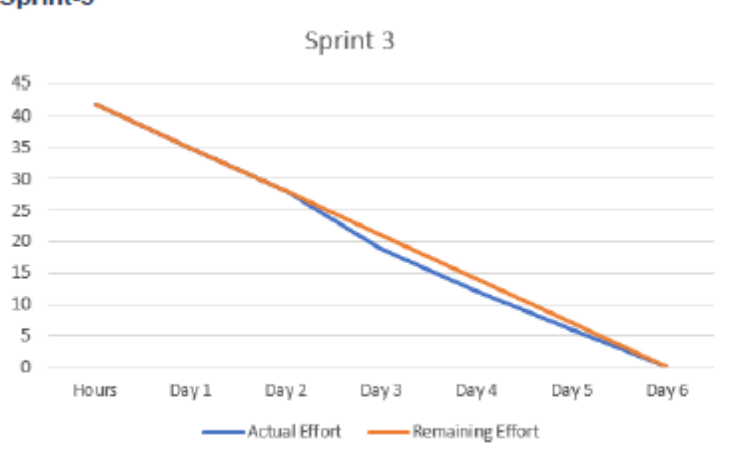
Sprint 2:

Sprint-2



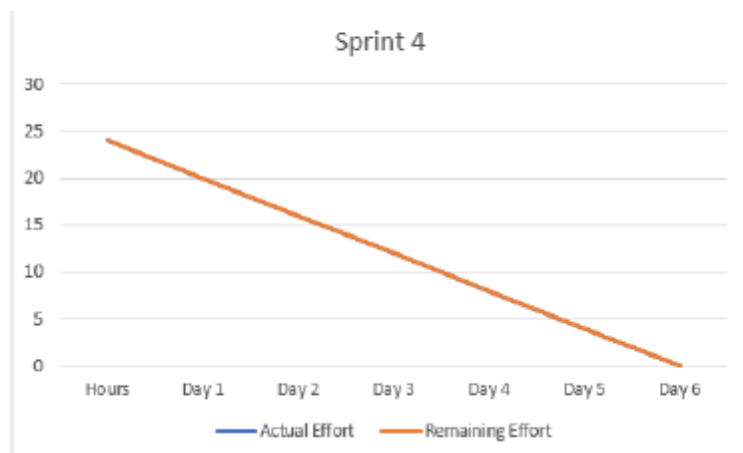
Sprint 3:

Sprint-3



Sprint_4:

Sprint-4



7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1: Predicting the stages of Diabetic retinopathy

CODE:

```

@app.route('/prediction')
def prediction():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'retina.jpg')
    full_filename1 = os.path.join(app.config['UPLOAD_FOLDER'], 'image6.png')
    return
render_template('prediction.html', image=full_filename, image2=full_filename1)

@app.route('/afterpred', methods=["GET", "POST"])
def aftepred():
    if request.method=="POST":
        full_filename2 = os.path.join(app.config['UPLOAD_FOLDER'],
'retina.jpg')
        full_filename1 = os.path.join(app.config['UPLOAD_FOLDER'],
'image6.png')
        f=request.files['pfile']
        filepath=os.path.join('static', 'uploads', f.filename)
        f.save(filepath)
        img=image.load_img(filepath, target_size=(224, 224))
        x=image.img_to_array(img)
        x=np.expand_dims(x, axis=0)
        img_data=preprocess_input(x)
        prediction=np.argmax(model.predict(img_data), axis=1)
        index=["no dr", "mild dr", "moderate dr", "severe dr", "proliferate"]
        result=str(index[prediction[0]])
        print(result)
        return
    render_template('prediction.html', prediction=result, image=full_filename2, im
age2=full_filename1)
    else:
        full_filename = os.path.join(app.config['UPLOAD_FOLDER'],
'loginimg.jpg')
        return render_template('login.html', pred="Please login using your
credentials", image=full_filename)

```

Explanation:

On entering the prediction page, the user will be directed to upload the test images. On clicking the

submit button ,the result indicating the stage of Diabetic Retinopathy will be displayed.

7.2 Feature 2: Queries from user

CODE:

```
@app.route('/query')
def query():
    my_database_query = client['my_database_query']

    dt=[]
    for document in my_database_query:
        dt.append(document['who'])
        dt.append(document['phoneno'])
        dt.append(document['query'])
    return render_template('query1.html', data=dt)

@app.route('/afterquery', methods=['POST', 'GET'])
def afterquery():
    my_database_query = client['my_database_query']
    x=[x for x in request.form.values()]
    data1={
        '_id':x[0],
        'who':x[1],
        'phoneno':x[2],
        'query':x[3],
    }
    query={'_id':{'$eq':data1['_id']}}
    docs=my_database_query.get_query_result(query)
    if(len(docs.all())==0):
        url=my_database_query.create_document(data1)
        my_database_query = client['my_database_query']
        dt=[]
        for document in my_database_query:
            dt.append(document['who'])
            dt.append(document['phoneno'])
            dt.append(document['query'])
        return render_template('query1.html',predict="Query submitted")
```

```

Successfully", data=dt)
    else:
        url=my_database_query.create_document(data1)
        my_database_query = client['my_database_query']
        dt=[]
        for document in my_database_query:
            dt.append(document['who'])
            dt.append(document['phoneno'])
            dt.append(document['query'])
        return render_template('query1.html', predict="Query submitted
Successfully", data=dt)

```

Explanation:

On entering the home page, any queries of the user can be posted by clicking the query button. In query page there will be the list of queries and responses posted so far, the user will be asked to enter phone number and corresponding query. On hitting the submit button their query will be submitted and can be viewed by both the user and admin.

7.3 Feature 3: Response to User Queries by admin

CODE:

```

@app.route('/respond')
def respond():
    my_database_query = client['my_database_query']

    dt=[]
    for document in my_database_query:
        dt.append(document['who'])
        dt.append(document['phoneno'])
        dt.append(document['query'])
    return render_template('respond.html', data=dt)

@app.route('/afterrespond', methods=['POST', 'GET'])
def afterrespond():
    my_database_query = client['my_database_query']
    x=[x for x in request.form.values()]
    data1={
        '_id':x[0],

```

```

        'who':x[1],
        'phoneno':x[2],
        'query':x[3]
    }
    query={'_id':{'$eq':data1['_id']}}
    docs=my_database_query.get_query_result(query)
    if(len(docs.all())==0):
        url=my_database_query.create_document(data1)
        my_database_query = client['my_database_query']
        dt=[]
        for document in my_database_query:
            dt.append(document['who'])
            dt.append(document['phoneno'])
            dt.append(document['query'])
        return render_template('respond.html',predict="Response posted
Successfully", data=dt)
    else:
        url=my_database_query.create_document(data1)
        my_database_query = client['my_database_query']
        dt=[]
        for document in my_database_query:
            dt.append(document['who'])
            dt.append(document['phoneno'])
            dt.append(document['query'])
        return render_template('respond.html',predict="Response posted
Successfully", data=dt)

```

Expalnation:

After Admin logins, by clicking the Respond Queries button in admin page, an admin can view all the queries and the responses sent so far, he/she can send response to every Query by entering the Query's phone number and the response. Once the response is posted it will be reflected on both admin and user's query page.

7.4 Feature 4: Forgot password

CODE:

```
@app.route('/fp')
def fp():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'forgotpw.png')
    return render_template('fp.html', image=full_filename)

@app.route('/afterfp', methods=['POST', 'GET'])
def afterfp():
    pn=request.form['phoneno']
    securityques=request.form['secques']
    npassw=request.form['npwd']
    cpassw=request.form['cpwd']

    docs=my_database[pn]

    if(npassw==cpassw and securityques==docs['securityquestion']):
        full_filename =
os.path.join(app.config['UPLOAD_FOLDER'], 'loginimg.jpg')
        docs['pwd'] = cpassw
        docs.save()
        return render_template('login.html', predict="Successfully
updated", image=full_filename)
    if(securityques!=docs['securityquestion']):
        full_filename =
os.path.join(app.config['UPLOAD_FOLDER'], 'forgotpw.png')
        return render_template('fp.html', predict="Incorrect answer to
security question", image=full_filename)
    if(npassw!=cpassw):
        full_filename =
os.path.join(app.config['UPLOAD_FOLDER'], 'forgotpw.png')
        return render_template('fp.html', predict="New and confirm password
does not match", image=full_filename)
```

Explanation:

User or Admin on reaching the login page can use the forgot password option and can change their passwords. For the purpose of security only when the answer to the security question by the user matches with the answer given during registration and only when the new and confirm password matches the password of the particular user will be updated.

7.5 Feature 5: Modification of location by User

CODE:

```
@app.route('/modify', methods=["GET", "POST"])
def modify():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'])
    return render_template('modify.html', predict="Please enter your new
location")

@app.route('/aftermodify', methods=["GET", "POST"])
def aftermodify():
    if request.method=="POST":
        user=session['pn']
        pwd=request.form['pwd']
        nloc=request.form['location']
        docs=my_database[user]
        if (pwd==docs['pwd']):
            full_filename =
os.path.join(app.config['UPLOAD_FOLDER'], 'pimg.jpg')
            docs['location'] = nloc
            docs.save()
            return render_template('modify.html', predict="Successfully
updated")
        else:
            full_filename = os.path.join(app.config['UPLOAD_FOLDER'],
'retina.jpg')
```

```

        full_filename1 = os.path.join(app.config['UPLOAD_FOLDER'],
'image6.png')
        return
render_template('prediction.html', image=full_filename, image2=full_filename1)

```

Explanation:

A user can change their location to get appropriate hospital preference by changing their location in the modify page. The the new location will get updated when password is correct, and their phone number will be retrieved using the sessions variable and the database will be updated

7.6 Feature 6: Update hospital location by admin

CODE:

```

@app.route('/uploc')
def uploc():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'])
    return render_template('uploc.html')

@app.route('/afteruploc', methods=["GET", "POST"])
def afteruploc():
    loc=request.form['loch']
    hname=request.form['hname']
    lname=request.form['lname']
    contact=request.form['contact']
    dbl=client['db1']
    docs=dbl[loc.upper()]
    hn=hname.strip().upper()
    count=0
    for i in range(len(docs['hospitals'])):
        if docs['hospitals'][i].strip().upper()==hn:
            docs['contacts'][i]=contact
            docs['locations'][i]=lname
            count=1
            docs.save()
            break
    if(count==0):

```

```

docs['hospitals'].append(hn)
docs['contacts'].append(contact)
docs.save()
full_filename = os.path.join(app.config['UPLOAD_FOLDER'])
return render_template('uploc.html', predict="Updated or added
successfully")

```

Explanation:

An admin can update the location and contact details of the hospital by changing it in the update location page in the admin dashboard

7.7 Feature 7: Hospital suggestions

CODE:

```

@app.route('/locsugges')
def locsugges():
    print("location")
    dbl=client['db1']
    dbu=client['db']

    pn=session['pn']
    docl=dbu[pn]
    location=docl["location"]

    docs=dbl[location.upper()]
    hospital=[]
    location=[]
    for h in range(len(docs["hospitals"])):
        hospital.append(docs['hospitals'][h])
    for l in range(len(docs["locations"])):
        location.append(docs['locations'][l])
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'])
    return
render_template('locsugges.html', hospital=hospital, location=location)

```

Explanation:

All the users will be provided with a list of hospitals near them with their contact details

7.8 Database Schema:

USED:IBM cloud database

DATABASE:NoSql-JSON

Database 1:db/My_database

To store user and admin details

id(User Phone number)

rev

name

pwd

email

location

securityquestion

Database 2:my_database_query

To store User query and Admin response

id(User Phone number)

rev

who

phoneno

query

Database 3:db1

To store the hospitals details

id(Location)

rev

hospitals

locations

contacts

8. TESTING

8.1 Test cases:

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
AdminPage_TC_OO1	UI	AdminPage	Verify the UI elements in Adminpage displays	Admin credentials must be created priorly with the application	1.Enter URL and click go 2.Verify admin with below UI elements: a.RespondQueries button b. Update location button c.Logout button d.Image showing admin e.Description of the project		Application should show below UI elements: a.RespondQueries button b. Update location button c.Logout button d.Image showing admin e.Description of the project	Working as expected	Pass				Sudha R Nandhini M V
AdminPage_TC_OO2	UI	Home page	Verify whether navigations of admin page works properly	Admin credentials must be created priorly with the application	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login 6.If it is an admin they will be redirected to the admin page on successful login		User should navigate to corresponding pages when clicked	Working as expected	Pass				Sudha R Nandhini M V
CloudCreation_TC_001	Functional	Database	Verify whether database is created		1.Sign in to the IBM cloud 2.Click on the ResourceList 3.Click on cloudant_n0 database 4.Check whether the database is created with the name given		Database being created	Working as expected	Pass				Vinetha R Ghayathri Devi V
Dashboard_TC_001	Non Functional	Prediction Page	Verify User can access the modify page from changing location details	User should already registered with the application and login with their credentials	1.From home page enter the login page through " Login" button 2.Enter your credentials and login 3.This will redirect to prediction page 4.Click the Modify button to enter the modify page to modify location		Should able to enter the Modify page	Working as expected	Pass				Sudha .R Nandhini.M.V
Dashboard_TC_002	Non Functional	Prediction Page	Verify user can able to submit the form with new location updates.	User should already registered with the application	1.Enter the password and new location 2.click submit	password:bbbb newlocation:coimbatore	Should able to submit the form and update database when password is correct for the user and display "updated"	Working as expected	Pass	Only four places like Coimbatore ,Chennai,Madurai,Trichy are added for location			Sudha .R Nandhini.M.V
Dashboard_TC_003	Non Functional	Prediction Page	Verify user cannot update on wrong password	User should already registered with the application	1.Enter the password and new location 2.click submit	password:abbb newlocation:chennai	Should not update the database and redirect the modify page and display "Invalid"	Working as expected	Pass				Sudha .R Nandhini.M.V
Dashboard_TC_004	Non Functional	Prediction Page	Verify user cannot update when new location is current location	User should already registered with the application	1.Enter the password and new location 2.click submit	password:bbbb newlocation:coimbatore	Should not update and render the same page	Not working as expected	Fail				Sudha .R Nandhini.M.V
Dashboard_TC_005	Non Functional	Prediction Page	Verify user able to access location sugesstion page	User should already registered with the application and login with their credentials	1.From home page enter the login page through " Login" button 2.Enter your credentials and login 3.This will redirect to prediction page 4.Click the Hospitals button to get hospital sugesstions		User should able to redirected to location sugesstion page and find the suggested locations	Working as expected	Pass	Only four places like Coimbatore ,Chennai,Madurai,Trichy are added for location			Sudha .R Nandhini.M.V
Dashboard_TC_006	Non Functional	Prediction Page	Verify user is given the sugesstion s only according to their current location	User should already registered with the application and login with their credentials	1.From home page enter the login page through " Login" button 2.Enter your credentials and login 3.This will redirect to prediction page 4.Click the Hospitals button to get hospital sugesstions		Should get appropriate hospital sugesstion	Working as expected	Pass				Sudha .R Nandhini.M.V
Dashboard_TC_007	Non Functional	Prediction Page	Verify after modification in location user able to get the location sugesstions for the new location	User should already registered with the application and login with their credentials	1.From home page enter the login page through " Login" button 2.Enter your credentials and login 3.This will redirect to prediction page 4.Click the Hospitals button to get hospital sugesstions		Should show the hospitals according to new locations	Working as expected	Pass				Sudha .R Nandhini.M.V
Database_TC_001	Non Functional	Admin page	Verify Admin can access the update hospital details	Admin should be valid	1.From home page enter the login page through " Login" button 2.Enter Admin credentials and login 3.This will redirect to admin page 4.Click the update location button		Should render the update location page	Working as expected	Pass				Sudha .R Nandhini.M.V
Database_TC_002	Non Functional	Admin page	Verify Admin can enter the updation details	Admin should be valid	1.Enter the hospital name ,location and contacts with the location detail and click submit	location:coimbatore hospital:Eye care hospital location:VV.puram contact:9998877	Should able to submit and make update in databse	Working as expected	Pass				Sudha .R Nandhini.M.V
Database_TC_003	Non Functional	Admin page	Verify Admin can add new hospital location	Admin should be valid	1.Enter the hospital name ,location and contacts with the location detail and click submit	location:coimbatore hospital:Vasan eye hospital location:vadakovai	Should able to Add new location	Working as expected	Pass				Sudha .R Nandhini.M.V
Forgot_password_TC_001	Functional	Login Page	Verify whether user able to access forgot password page	The user should registered earlier with the application	1.Go to Home page and click login button 2.In login page Click on forgot password link		Should open a page which fetch for new credentials	Working as expected	Pass				Vinetha R Ghayathri Devi V
Forgot_password_TC_002	Functional	Forgot password page	Verify User able to submit the new credentials	The user should registered earlier with the application	1.Enter the phone number,security question,new password and confirm password 2.Click on Submit button	Phone.no:9876543210 Security question:dog newpassword:bbbb confirm password:bbbb	Should submit the details and display process sucessful	Working as expected	Pass				Vinetha R Ghayathri Devi V
Forgot_password_TC_003	Functional	Login Page	Verify whether User able to login with new credentials	The user should registered earlier with the application and the submission of new password should be successful	1.Enter the login page with Login button 2.Enter the credentials with new password 3.Submit the form	Phone.no:9876543210 password:bbbb	User should login with new password	Working as expected	Pass				Vinetha R Ghayathri Devi V
Forgot_password_TC_004	Functional	Forgot password page	Verify user able to update new password with correct phone number and correct answer to security questions and with matching new and confirm password	The user should registered earlier with the application	1.Enter the phone number,security question,new password and conform password 2.Click on Submit button	Phone.no:9876543210 Security question:dog newpassword:bbbb confirm password:bbbb	User could Update new password in databse	Working as expected	Pass				Vinetha R Ghayathri Devi V

Forgot_password_TC_005	Functional	Forgot password page	Verify user able to update with Invalid credentials	The user should registered earlier with the application	1.Enter the phone number,security question,new password and conform password 2.Click on Submit button	Phone.no:9876543210 Security question:cat newpassword:abbbb confirm password:bbbb	User could not Update new password in database and display " login with Valid credentials"	Working as expected	Pass					Vinetha R Ghayathri Devi V
HomePage_TC_001	UI	Home Page	Verify the UI elements in Homepage displays		1.Enter URL and click go 2.Verify homepage with below UI elements: a.Home button b. Register button c.Login button d.Image describing Diabetic Retinopathy e.Description of the project		Application should show below UI elements: a.Home button b. Register button c.Login button d.Image describing Diabetic Retinopathy e.Description of the project	Working as expected	Pass					Vinetha R Ghayathri Devi V
HomePage_TC_002	UI	Home page	Verify whether navigations work properly		1.Enter URL and click go 2.Home page will be displayed 3.Click on any of the various navigations like		User should navigate to corresponding pages when clicked	Working as expected	Pass					Vinetha R Ghayathri Devi V
HomePage_TC_003	UI	Home page	Verify whether navigation of query works properly		1.Enter URL and click go 2.Home page will be displayed 3.Click on query.		User should navigate to corresponding query page when clicked	Working as expected	Pass					Sudha R Nandhini M V
LoginPage_TC_001	Functional	Login page	Verify user is able to log into application with valid credentials	User should already be registered with the application	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login	Phoneno:9876543210 Password:aaa	Application should redirect to Prediction page.	Working as expected	Pass					Vinetha R Ghayathri Devi V
LoginPage_TC_002	Functional	Login page	Verify user is unable to log into application with invalid credentials	User should already be registered with the application	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login	Phoneno:9876543210 Password:bbb Phoneno:9876543211 Password:aaa	Application should show 'Incorrect Password' if password is incorrect and should show 'Phone number not found' if the user logins without Registration	Working as expected	Pass					Vinetha R Ghayathri Devi V
LoginPage_TC_003	Functional	Login page	Verify whether admin is taken to corresponding admin page when logged in using valid credentials	Admin credentials must be created priorly with the application	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login	Phoneno:admin Password:bbb logintype:admin	Application should redirect to admin page	Working as expected	Pass					Sudha R Nandhini M V

LoginPage_TC_004	Functional	Login page	Verify whether admin is not taken to corresponding page when logged in using invalid credentials	Admin credentials must be created priorly with the application	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login	Phoneno:admin Password:ccc logintype:admin Phoneno:admin Password:bbb logintype:user	Application should show 'Incorrect Password' if password is incorrect and should show 'Invalid logintype if the user logins with wrong logintype	Working as expected	Pass					Sudha R Nandhini M V
PredictionPage_TC_001	UI	Prediction Page	Verify the UI elements in Adminpage displays	The user must be registered priorly with the application and logged in	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login 6.If it is an user they will be redirected to the prediction page on successful login		Application should show below UI elements: a.Home button b.Logout button c.File upload area d.Image showing different DR classification	Working as expected	Pass					Sudha R Nandhini M V
PredictionPage_TC_002	Functional	Prediction Page	Verify whether the model created predicts correct result when image of retina is given as input in file upload area	The user must be registered priorly with the application and logged in	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login 6.If it is an user they will be redirected to the prediction page on successful login 7.Select the image to be uploaded and click on submit	File chosen:c406325360b1.png	Application should predict the correct type of DR classification(Moderate DR)	Working as expected	Pass					Sudha R Nandhini M V
Query_TC_001	Non-Functional	Home page	Verify whether user able access query page		1.Go to Home page and click query button		User should get the query page to post queries	Working as expected	Pass					Vinetha R Ghayathri Devi V
Query_TC_002	Non-Functional	Query page	Verify user is able to submit the query		1.Enter the phone number and query to be posted 2.Click submit button	Phone.no:9876543210 query:Need to have image file format to take result	Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass					Vinetha R Ghayathri Devi V
QueryPage_TC_001	Functional	Query page	Verify whether all the queries posted are visible		1.Enter URL and click go 2.Home page will be displayed 3.Click on query.		User should be able to see all the queries posted by everyone and responses so far	Working as expected	Pass					Sudha R Nandhini M V
QueryPage_TC_002	Functional	Query page	Verify whether the user is able to post his query and see it being displayed on the screen		1.Enter URL and click go 2.Home page will be displayed 3.Click on query. 4.Type phone number and query and press submit	Phone number:9876543210 Query:Please tell me the format of files to upload	User should be able to see his query submitted in the screen	Working as expected	Pass					Sudha R Nandhini M V

RegistrationPage_TC_001	Functional	Registration page	Verify user is able to register with the appropriate credentials if a new user		1.Enter URL and click go 2.Home page will be displayed 3.Click on Register button 4.Will be redirected to the registration page 5.Enter the credentials and click register	Name:xxx Email id:xxx@gmail.com Phoneno:9876543210 Location:yyy Password:aaa	Application should redirect to Login page displaying "Registration successful please login using your credentials"	Working as expected	Pass				Vinetha R Ghayathri Devi V
RegistrationPage_TC_002	Functional	Registration page	Verify user is unable to register with the credentials if an old user		1.Enter URL and click go 2.Home page will be displayed 3.Click on Register button 4.Will be redirected to the registration page 5.Enter the credentials and click register	Name:xxx Email id:xxx@gmail.com Phoneno:9876543210 Location:yyy Password:aaa	Application should show "You are already a user login using your credentials"	Working as expected	Pass				Vinetha R Ghayathri Devi V
RegistrationPage_TC_003	Functional	Registration page	Verify User should be able to register with security question		1.Go to registration page and click register button 2.In register page, enter the credentials with security questions	Name:xxx Email id:xxx@gmail.com phone no:9876543210 location yyy security ques:dog password:aaaa	Should be able to register with security questions	Working as expected	Pass				Vinetha R Ghayathri Devi V
RespondQueriesPage_TC_001	Functional	Respond Query page	Verify whether all the queries and responses posted are visible	Admin credentials must be created priorly with the application and logged in	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login 6.If it is an admin they will be redirected to the admin page on successful login 7.Click on Respond Queries button		Admin should be able to see all the queries posted by everyone and responses so far	Working as expected	Pass				Sudha R Nandhini M V
RespondQueriesPage_TC_002	Functional	Respond Query page	Verify whether the admin is able to post his response and see it being displayed on the screen	Admin credentials must be created priorly with the application and logged in	1.Enter URL and click go 2.Home page will be displayed 3.Click on login button 4.Will be redirected to the login page 5.Enter the credentials and click login 6.If it is an admin they will be redirected to the admin page on successful login 7.Click on Respond Queries button	Request of 9876543210 Response:Any image format	Admin should be able to see his response posted in the screen	Working as expected	Pass				Sudha R Nandhini M V

8.2 User Acceptance Testing:

Acceptance Testing UAT Execution & Report Submission

Date	16 November 2022
Team ID	PNT2022TMID06905
Project Name	Project – Deep Learning Fundus image analysis for early detection of Diabetic Retinopathy
Maximum Marks	4 Marks

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Deep Learning fundus image analysis for early detection of Diabetic Retinopathy project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	20	0	0	0	20
Duplicate	0	2	0	0	2
External	0	0	0	0	0
Fixed	22	0	0	0	22
Not Reproduced	0	0	0	0	0
Skipped	1	0	0	0	1
Won't Fix	0	5	0	0	0
Totals	42	2	0	0	43

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	18	0	0	18
Client Application	8	0	0	8
Security	2	0	0	2

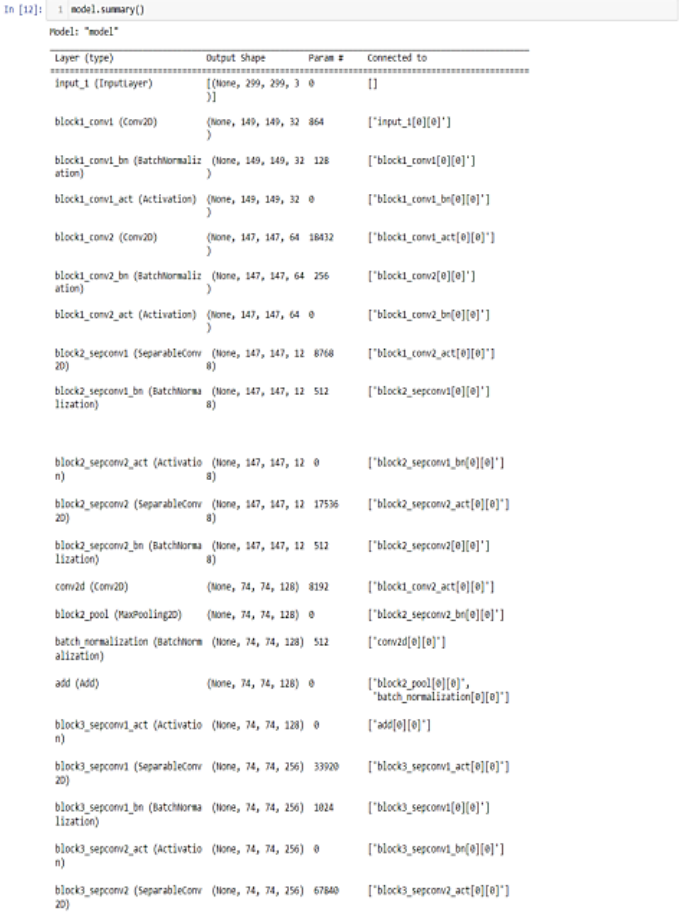
Outsource Shipping	0	0	0	0
Exception Reporting	6	0	0	6
Final Report Output	2	0	0	2
Version Control	0	0	0	0

9.RESULTS:

9.1 Performance Metrics

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	-	 <pre>In [12]: model.summary() Model: "model" Layer (type) Output Shape Param # Connected to ----- input_1 (InputLayer) [(None, 299, 299, 3 0)] block1_conv1 (Conv2D) (None, 149, 149, 32 864) block1_conv1_bn (BatchNormaliz ation) (None, 149, 149, 32 128) block1_conv1_act (Activation) (None, 149, 149, 32 0) block1_conv2 (Conv2D) (None, 147, 147, 64 18432) block1_conv2_bn (BatchNormaliz ation) (None, 147, 147, 64 256) block1_conv2_act (Activation) (None, 147, 147, 64 0) block2_seconv1 (SeparableConv 2D) (None, 147, 147, 12 8768) block2_seconv1_bn (BatchNorma lization) (None, 147, 147, 12 512) block2_seconv2_act (Activatio n) (None, 147, 147, 12 0) block2_seconv2 (SeparableConv 2D) (None, 147, 147, 12 17536) block2_seconv2_bn (BatchNorma lization) (None, 147, 147, 12 512) conv2d (Conv2D) (None, 74, 74, 128) 8192) block2_pool (MaxPooling2D) (None, 74, 74, 128) 0) batch_normalization (BatchNorm alization) (None, 74, 74, 128) 512) add (Add) (None, 74, 74, 128) 0) block3_seconv1_act (Activatio n) (None, 74, 74, 128) 0) block3_seconv1 (SeparableConv 2D) (None, 74, 74, 256) 33920) block3_seconv1_bn (BatchNorma lization) (None, 74, 74, 256) 1024) block3_seconv2_act (Activatio n) (None, 74, 74, 256) 0) block3_seconv2 (SeparableConv 2D) (None, 74, 74, 256) 67840)</pre>

			<div> <div>block3_sepconv2_bn (BatchNormalization)</div> <div>(None, 74, 74, 256) 1024</div> <div>['block3_sepconv2[0][0]']</div> </div> <div> <div>conv2d_1 (Conv2D)</div> <div>(None, 37, 37, 256) 32768</div> <div>['add[0][0]']</div> </div> <div> <div>block3_pool (MaxPooling2D)</div> <div>(None, 37, 37, 256) 0</div> <div>['block3_sepconv2_bn[0][0]']</div> </div> <div> <div>batch_normalization_1 (BatchNormalization)</div> <div>(None, 37, 37, 256) 1024</div> <div>['conv2d_1[0][0]']</div> </div> <div> <div>add_1 (Add)</div> <div>(None, 37, 37, 256) 0</div> <div>['block3_pool[0][0]', 'batch_normalization_1[0][0]']</div> </div> <div> <div>block4_sepconv1_act (Activation)</div> <div>(None, 37, 37, 256) 0</div> <div>['add_1[0][0]']</div> </div> <div> <div>block4_sepconv1 (SeparableConv2D)</div> <div>(None, 37, 37, 728) 188672</div> <div>['block4_sepconv1_act[0][0]']</div> </div> <div> <div>block4_sepconv1_bn (BatchNormalization)</div> <div>(None, 37, 37, 728) 2912</div> <div>['block4_sepconv1[0][0]']</div> </div> <div> <div>block4_sepconv2_act (Activation)</div> <div>(None, 37, 37, 728) 0</div> <div>['block4_sepconv1_bn[0][0]']</div> </div> <div> <div>block4_sepconv2 (SeparableConv2D)</div> <div>(None, 37, 37, 728) 536536</div> <div>['block4_sepconv2_act[0][0]']</div> </div> <div> <div>block4_sepconv2_bn (BatchNormalization)</div> <div>(None, 37, 37, 728) 2912</div> <div>['block4_sepconv2[0][0]']</div> </div> <div> <div>conv2d_2 (Conv2D)</div> <div>(None, 19, 19, 728) 186368</div> <div>['add_1[0][0]']</div> </div> <div> <div>block4_pool (MaxPooling2D)</div> <div>(None, 19, 19, 728) 0</div> <div>['block4_sepconv2_bn[0][0]']</div> </div>	
			<div> <div>batch_normalization_2 (BatchNormalization)</div> <div>(None, 19, 19, 728) 2912</div> <div>['conv2d_2[0][0]']</div> </div> <div> <div>add_2 (Add)</div> <div>(None, 19, 19, 728) 0</div> <div>['block4_pool[0][0]', 'batch_normalization_2[0][0]']</div> </div> <div> <div>block5_sepconv1_act (Activation)</div> <div>(None, 19, 19, 728) 0</div> <div>['add_2[0][0]']</div> </div> <div> <div>block5_sepconv1 (SeparableConv2D)</div> <div>(None, 19, 19, 728) 536536</div> <div>['block5_sepconv1_act[0][0]']</div> </div> <div> <div>block5_sepconv1_bn (BatchNormalization)</div> <div>(None, 19, 19, 728) 2912</div> <div>['block5_sepconv1[0][0]']</div> </div> <div> <div>block5_sepconv2_act (Activation)</div> <div>(None, 19, 19, 728) 0</div> <div>['block5_sepconv1_bn[0][0]']</div> </div> <div> <div>block5_sepconv2 (SeparableConv2D)</div> <div>(None, 19, 19, 728) 536536</div> <div>['block5_sepconv2_act[0][0]']</div> </div> <div> <div>block5_sepconv2_bn (BatchNormalization)</div> <div>(None, 19, 19, 728) 2912</div> <div>['block5_sepconv2[0][0]']</div> </div> <div> <div>block5_sepconv3_act (Activation)</div> <div>(None, 19, 19, 728) 0</div> <div>['block5_sepconv2_bn[0][0]']</div> </div> <div> <div>block5_sepconv3 (SeparableConv2D)</div> <div>(None, 19, 19, 728) 536536</div> <div>['block5_sepconv3_act[0][0]']</div> </div> <div> <div>block5_sepconv3_bn (BatchNormalization)</div> <div>(None, 19, 19, 728) 2912</div> <div>['block5_sepconv3[0][0]']</div> </div> <div> <div>add_3 (Add)</div> <div>(None, 19, 19, 728) 0</div> <div>['block5_sepconv3_bn[0][0]', 'add_2[0][0]']</div> </div>	

			block6_sepconv1_act (Activation (None, 19, 19, 728) 0 n) ['add_3[0][0]'] block6_sepconv1 (SeparableConv (None, 19, 19, 728) 536536 20) ['block6_sepconv1_act[0][0]'] block6_sepconv1_bn (BatchNormaliza (None, 19, 19, 728) 2912 lization) ['block6_sepconv1[0][0]'] block6_sepconv2_act (Activation (None, 19, 19, 728) 0 n) ['block6_sepconv1_bn[0][0]'] block6_sepconv2 (SeparableConv (None, 19, 19, 728) 536536 20) ['block6_sepconv2_act[0][0]'] block6_sepconv2_bn (BatchNormaliza (None, 19, 19, 728) 2912 lization) ['block6_sepconv2[0][0]'] block6_sepconv3_act (Activation (None, 19, 19, 728) 0 n) ['block6_sepconv2_bn[0][0]'] block6_sepconv3 (SeparableConv (None, 19, 19, 728) 536536 20) ['block6_sepconv3_act[0][0]'] block6_sepconv3_bn (BatchNormaliza (None, 19, 19, 728) 2912 lization) ['block6_sepconv3[0][0]'] add_4 (Add) (None, 19, 19, 728) 0 ['block6_sepconv3_bn[0][0]', 'add_3[0][0]'] block7_sepconv1_act (Activation (None, 19, 19, 728) 0 n) ['add_4[0][0]'] block7_sepconv1 (SeparableConv (None, 19, 19, 728) 536536 20) ['block7_sepconv1_act[0][0]'] block7_sepconv1_bn (BatchNormaliza (None, 19, 19, 728) 2912 lization) ['block7_sepconv1[0][0]'] block7_sepconv2_act (Activation (None, 19, 19, 728) 0 n) ['block7_sepconv1_bn[0][0]'] block7_sepconv2 (SeparableConv (None, 19, 19, 728) 536536 20) ['block7_sepconv2_act[0][0]'] block7_sepconv2_bn (BatchNormaliza (None, 19, 19, 728) 2912 lization) ['block7_sepconv2[0][0]'] block7_sepconv3_act (Activation (None, 19, 19, 728) 0 n) ['block7_sepconv2_bn[0][0]'] block7_sepconv3 (SeparableConv (None, 19, 19, 728) 536536 20) ['block7_sepconv3_act[0][0]'] block7_sepconv3_bn (BatchNormaliza (None, 19, 19, 728) 2912 lization) ['block7_sepconv3[0][0]'] add_5 (Add) (None, 19, 19, 728) 0 ['block7_sepconv3_bn[0][0]', 'add_4[0][0]'] block8_sepconv1_act (Activation (None, 19, 19, 728) 0 n) ['add_5[0][0]'] block8_sepconv1 (SeparableConv (None, 19, 19, 728) 536536 20) ['block8_sepconv1_act[0][0]'] block8_sepconv1_bn (BatchNormaliza (None, 19, 19, 728) 2912 lization) ['block8_sepconv1[0][0]'] block8_sepconv2_act (Activation (None, 19, 19, 728) 0 n) ['block8_sepconv1_bn[0][0]']
--	--	--	---

			<div> <div>block8_sepconv2 (SeparableConv (None, 19, 19, 728) 536536</div> <div>2D)</div> <div>['block8_sepconv2_act[0][0]']</div> </div> <div> <div>block8_sepconv2_bn (BatchNorm (None, 19, 19, 728) 2912</div> <div>lization)</div> <div>['block8_sepconv2[0][0]']</div> </div> <div> <div>block8_sepconv3_act (Activatio (None, 19, 19, 728) 0</div> <div>n)</div> <div>['block8_sepconv3_bn[0][0]']</div> </div> <div> <div>block8_sepconv3 (SeparableConv (None, 19, 19, 728) 536536</div> <div>2D)</div> <div>['block8_sepconv3_act[0][0]']</div> </div> <div> <div>block8_sepconv3_bn (BatchNorm (None, 19, 19, 728) 2912</div> <div>lization)</div> <div>['block8_sepconv3[0][0]']</div> </div> <div> <div>add_6 (Add) (None, 19, 19, 728) 0</div> <div></div> <div>['block8_sepconv3_bn[0][0]', 'add_5[0][0]']</div> </div> <div> <div>block8_sepconv1_act (Activatio (None, 19, 19, 728) 0</div> <div>n)</div> <div>['add_6[0][0]']</div> </div> <div> <div>block9_sepconv1 (SeparableConv (None, 19, 19, 728) 536536</div> <div>2D)</div> <div>['block9_sepconv1_act[0][0]']</div> </div> <div> <div>block9_sepconv1_bn (BatchNorm (None, 19, 19, 728) 2912</div> <div>lization)</div> <div>['block9_sepconv1[0][0]']</div> </div> <div> <div>block9_sepconv2_act (Activatio (None, 19, 19, 728) 0</div> <div>n)</div> <div>['block9_sepconv1_bn[0][0]']</div> </div> <div> <div>block9_sepconv2 (SeparableConv (None, 19, 19, 728) 536536</div> <div>2D)</div> <div>['block9_sepconv2_act[0][0]']</div> </div> <div> <div>block9_sepconv2_bn (BatchNorm (None, 19, 19, 728) 2912</div> <div>lization)</div> <div>['block9_sepconv2[0][0]']</div> </div> <div> <div>block9_sepconv3_act (Activatio (None, 19, 19, 728) 0</div> <div>n)</div> <div>['block9_sepconv2_bn[0][0]']</div> </div> <div> <div>block9_sepconv3 (SeparableConv (None, 19, 19, 728) 536536</div> <div>2D)</div> <div>['block9_sepconv3_act[0][0]']</div> </div> <div> <div>block9_sepconv3_bn (BatchNorm (None, 19, 19, 728) 2912</div> <div>lization)</div> <div>['block9_sepconv3[0][0]']</div> </div> <div> <div>add_7 (Add) (None, 19, 19, 728) 0</div> <div></div> <div>['block9_sepconv3_bn[0][0]', 'add_6[0][0]']</div> </div> <div> <div>block10_sepconv1_act (Activati (None, 19, 19, 728) 0</div> <div>on)</div> <div>['add_7[0][0]']</div> </div> <div> <div>block10_sepconv1 (SeparableCon (None, 19, 19, 728) 536536</div> <div>v2D)</div> <div>['block10_sepconv1_act[0][0]']</div> </div> <div> <div>block10_sepconv1_bn (BatchNorm (None, 19, 19, 728) 2912</div> <div>alization)</div> <div>['block10_sepconv1[0][0]']</div> </div> <div> <div>block10_sepconv2_act (Activati (None, 19, 19, 728) 0</div> <div>on)</div> <div>['block10_sepconv1_bn[0][0]']</div> </div> <div> <div>block10_sepconv2 (SeparableCon (None, 19, 19, 728) 536536</div> <div>v2D)</div> <div>['block10_sepconv2_act[0][0]']</div> </div> <div> <div>block10_sepconv2_bn (Batchnorm (None, 19, 19, 728) 2912</div> <div>alization)</div> <div>['block10_sepconv2[0][0]']</div> </div> <div> <div>block10_sepconv3_act (Activati (None, 19, 19, 728) 0</div> <div>on)</div> <div>['block10_sepconv2_bn[0][0]']</div> </div>
--	--	--	--

			<div> <div>block10_sepconv3 (SeparableCon</div> <div>v2D)</div> <div>(None, 19, 19, 728)</div> <div>536536</div> <div>['block10_sepconv3_act[0][0]']</div> </div> <div> <div>block10_sepconv3_bn (BatchNorm</div> <div>alization)</div> <div>(None, 19, 19, 728)</div> <div>2912</div> <div>['block10_sepconv3[0][0]']</div> </div> <div> <div>add_8 (Add)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['block10_sepconv3_bn[0][0]', 'add_7[0][0]']</div> </div> <div> <div>block11_sepconv1_act (Activati</div> <div>on)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['add_8[0][0]']</div> </div> <div> <div>block11_sepconv1 (SeparableCon</div> <div>v2D)</div> <div>(None, 19, 19, 728)</div> <div>536536</div> <div>['block11_sepconv1_act[0][0]']</div> </div> <div> <div>block11_sepconv1_bn (BatchNorm</div> <div>alization)</div> <div>(None, 19, 19, 728)</div> <div>2912</div> <div>['block11_sepconv1[0][0]']</div> </div> <div> <div>block11_sepconv2_act (Activati</div> <div>on)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['block11_sepconv1_bn[0][0]']</div> </div> <div> <div>block11_sepconv2 (SeparableCon</div> <div>v2D)</div> <div>(None, 19, 19, 728)</div> <div>536536</div> <div>['block11_sepconv2_act[0][0]']</div> </div> <div> <div>block11_sepconv2_bn (BatchNorm</div> <div>alization)</div> <div>(None, 19, 19, 728)</div> <div>2912</div> <div>['block11_sepconv2[0][0]']</div> </div> <div> <div>block11_sepconv3_act (Activati</div> <div>on)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['block11_sepconv2_bn[0][0]']</div> </div> <div> <div>block11_sepconv3 (SeparableCon</div> <div>v2D)</div> <div>(None, 19, 19, 728)</div> <div>536536</div> <div>['block11_sepconv3_act[0][0]']</div> </div> <div> <div>block11_sepconv3_bn (BatchNorm</div> <div>alization)</div> <div>(None, 19, 19, 728)</div> <div>2912</div> <div>['block11_sepconv3[0][0]']</div> </div> <div> <div>add_9 (Add)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['block11_sepconv3_bn[0][0]', 'add_8[0][0]']</div> </div> <div> <div>block12_sepconv1_act (Activati</div> <div>on)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['add_9[0][0]']</div> </div> <div> <div>block12_sepconv1 (SeparableCon</div> <div>v2D)</div> <div>(None, 19, 19, 728)</div> <div>536536</div> <div>['block12_sepconv1_act[0][0]']</div> </div> <div> <div>block12_sepconv1_bn (BatchNorm</div> <div>alization)</div> <div>(None, 19, 19, 728)</div> <div>2912</div> <div>['block12_sepconv1[0][0]']</div> </div> <div> <div>block12_sepconv2_act (Activati</div> <div>on)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['block12_sepconv1_bn[0][0]']</div> </div> <div> <div>block12_sepconv2 (SeparableCon</div> <div>v2D)</div> <div>(None, 19, 19, 728)</div> <div>536536</div> <div>['block12_sepconv2_act[0][0]']</div> </div> <div> <div>block12_sepconv2_bn (BatchNorm</div> <div>alization)</div> <div>(None, 19, 19, 728)</div> <div>2912</div> <div>['block12_sepconv2[0][0]']</div> </div> <div> <div>block12_sepconv3_act (Activati</div> <div>on)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['block12_sepconv2_bn[0][0]']</div> </div> <div> <div>block12_sepconv3 (SeparableCon</div> <div>v2D)</div> <div>(None, 19, 19, 728)</div> <div>536536</div> <div>['block12_sepconv3_act[0][0]']</div> </div> <div> <div>block12_sepconv3_bn (BatchNorm</div> <div>alization)</div> <div>(None, 19, 19, 728)</div> <div>2912</div> <div>['block12_sepconv3[0][0]']</div> </div> <div> <div>add_10 (Add)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['block12_sepconv3_bn[0][0]', 'add_9[0][0]']</div> </div> <div> <div>block13_sepconv1_act (Activati</div> <div>on)</div> <div>(None, 19, 19, 728)</div> <div>0</div> <div>['add_10[0][0]']</div> </div>
--	--	--	--

2.	Accuracy	<p>Training Accuracy – 0.8229</p> <p>Validation Accuracy - 0.8542</p>	<pre>In [19]: 1 r = model.fit_generator(training_set, 2 validation_data=test_set, 3 epochs=30, 4 steps_per_epoch=len(training_set)//32, 5 validation_steps=len(test_set)//32 6)</pre> <p>C:\Users\SRI SAJ\AppData\Local\Temp\ipykernel_7336\2721303376.py:1: UserWarning: 'Model.fit_generator' is deprecated and will be removed in a future version. Please use 'Model.fit', which supports generators. r = model.fit_generator(training_set,</p> <p>Epoch 1/30 3/3 [=====] - 75s 29s/step - loss: 2.6565 - accuracy: 0.8021 - val_loss: 3.1720 - val_accuracy: 0.7500 Epoch 2/30 3/3 [=====] - 50s 23s/step - loss: 3.3913 - accuracy: 0.7708 - val_loss: 2.9065 - val_accuracy: 0.7500 Epoch 3/30 3/3 [=====] - 55s 22s/step - loss: 3.5876 - accuracy: 0.7396 - val_loss: 2.6472 - val_accuracy: 0.7917 Epoch 4/30 3/3 [=====] - 55s 22s/step - loss: 2.8447 - accuracy: 0.7604 - val_loss: 4.0966 - val_accuracy: 0.7708 Epoch 5/30 3/3 [=====] - 54s 22s/step - loss: 3.9040 - accuracy: 0.7604 - val_loss: 2.7981 - val_accuracy: 0.7188 Epoch 6/30 3/3 [=====] - 55s 22s/step - loss: 2.1252 - accuracy: 0.8333 - val_loss: 2.0358 - val_accuracy: 0.8021 Epoch 7/30 3/3 [=====] - 49s 22s/step - loss: 3.1932 - accuracy: 0.7692 - val_loss: 2.9658 - val_accuracy: 0.7708 Epoch 8/30 3/3 [=====] - 53s 21s/step - loss: 5.5181 - accuracy: 0.7396 - val_loss: 2.7913 - val_accuracy: 0.7812 Epoch 9/30 3/3 [=====] - 52s 21s/step - loss: 2.6200 - accuracy: 0.8125 - val_loss: 2.4544 - val_accuracy: 0.6875 Epoch 10/30 3/3 [=====] - 52s 20s/step - loss: 2.3717 - accuracy: 0.7917 - val_loss: 3.0723 - val_accuracy: 0.6979 Epoch 11/30 3/3 [=====] - 53s 21s/step - loss: 3.2614 - accuracy: 0.7500 - val_loss: 2.3165 - val_accuracy: 0.8125 Epoch 12/30 3/3 [=====] - 52s 20s/step - loss: 4.0855 - accuracy: 0.6875 - val_loss: 3.2525 - val_accuracy: 0.7812</p>
			<p>Epoch 13/30 3/3 [=====] - 51s 21s/step - loss: 3.0122 - accuracy: 0.8229 - val_loss: 2.4108 - val_accuracy: 0.7500 Epoch 14/30 3/3 [=====] - 50s 20s/step - loss: 2.5197 - accuracy: 0.7604 - val_loss: 1.2118 - val_accuracy: 0.8229 Epoch 15/30 3/3 [=====] - 52s 20s/step - loss: 3.1242 - accuracy: 0.7917 - val_loss: 3.3027 - val_accuracy: 0.7292 Epoch 16/30 3/3 [=====] - 50s 20s/step - loss: 3.3930 - accuracy: 0.7396 - val_loss: 1.5712 - val_accuracy: 0.8750 Epoch 17/30 3/3 [=====] - 53s 21s/step - loss: 2.5512 - accuracy: 0.7812 - val_loss: 1.9767 - val_accuracy: 0.7812 Epoch 18/30 3/3 [=====] - 50s 20s/step - loss: 2.0621 - accuracy: 0.8229 - val_loss: 2.6166 - val_accuracy: 0.6875 Epoch 19/30 3/3 [=====] - 51s 20s/step - loss: 4.3960 - accuracy: 0.7083 - val_loss: 2.5433 - val_accuracy: 0.7396 Epoch 20/30 3/3 [=====] - 52s 21s/step - loss: 1.6552 - accuracy: 0.8438 - val_loss: 3.4086 - val_accuracy: 0.7083 Epoch 21/30 3/3 [=====] - 52s 20s/step - loss: 3.8576 - accuracy: 0.7917 - val_loss: 3.9289 - val_accuracy: 0.8021 Epoch 22/30 3/3 [=====] - 49s 19s/step - loss: 3.0081 - accuracy: 0.7292 - val_loss: 2.4708 - val_accuracy: 0.7396 Epoch 23/30 3/3 [=====] - 49s 20s/step - loss: 2.1175 - accuracy: 0.7812 - val_loss: 1.4821 - val_accuracy: 0.8125 Epoch 24/30 3/3 [=====] - 54s 22s/step - loss: 3.0934 - accuracy: 0.7500 - val_loss: 2.5336 - val_accuracy: 0.7917 Epoch 25/30 3/3 [=====] - 51s 23s/step - loss: 3.4418 - accuracy: 0.7821 - val_loss: 3.1132 - val_accuracy: 0.7917 Epoch 26/30 3/3 [=====] - 54s 21s/step - loss: 1.6937 - accuracy: 0.8229 - val_loss: 1.6791 - val_accuracy: 0.8438 Epoch 27/30 3/3 [=====] - 51s 20s/step - loss: 3.0819 - accuracy: 0.7917 - val_loss: 2.6662 - val_accuracy: 0.7604 Epoch 28/30 3/3 [=====] - 53s 21s/step - loss: 3.1182 - accuracy: 0.7500 - val_loss: 1.7582 - val_accuracy: 0.7604 Epoch 29/30 3/3 [=====] - 53s 20s/step - loss: 2.0389 - accuracy: 0.7917 - val_loss: 2.2502 - val_accuracy: 0.7500 Epoch 30/30 3/3 [=====] - 53s 21s/step - loss: 2.8220 - accuracy: 0.8229 - val_loss: 1.1712 - val_accuracy: 0.8542</p>

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- 1.The Application can able to detect the Stages of Diabetic Retinopathy easily from the remote location.
- 2.The Application can saves time and money.
- 3.The User can able to post queries and get their doubts clarified .
- 4.The user can see through nearby Eye hospitals.
- 5.User can able to change their password when the forgot.
- 6.User can change their location .

DISADVANTAGES:

- 1.The location suggesstion is limited to four locations.
- 2.The security question is too weak.
- 3.The password is not encryted .
- 4.The user should have some Technical knowledge.

11. CONCLUSION:

Automated diagnostic systems significantly reduce the time required to determine diagnoses, saving effort and costs for ophthalmologists and result in the timely treatment of patients. Automated systems for DR detection play an important role in detecting DR at an early stage. Most researchers have used the CNN for the classification and the detection of the DR images due to its efficiency. This project is one of the implementations of automated systems for Diabetic Retinopathy detection and classification that used deep learning techniques which can be extended for high-level diagnosis in future.

12. FUTURE SCOPE:

This project is the first step in diagnosing Diabetic Retinopathy with automated systems. It can be taken as a forerunner and a lot of diagnostic systems can be developed to ease the time and efforts of the ophthalmologists and the patients. In the future modernized era, automated systems will have a great command in the day-to-day activities of people. This project in future can be extended with great accuracy and can be implemeted as an autonomous system to detect Diabetic Retinopathy by screening the eyes of the patients directly.

13.APPENDIX:

SOURCE CODE:

HTML FILES:

1. index.html

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of  
Diabetic Retinopathy</title>
```

```
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
```

```
  </head>
```

```
  <style>
```

```
    html,body{
```

```
      margin:0;
```

```
      width:100%;
```

```
      height:100%;
```

```
      font-family:Roboto;
```

```
      display: flex;
```

```
      flex-direction: column;
```

```
    }
```

```
    .header{
```

```
      position: fixed;
```

```
      top: 0px;
```

```
      width:100%;
```

```
      height:100px;
```

```
      background-color:dimgrey;
```

```
      display: flex;
```

```
      flex-direction: row;
```

```
      align-items: center;
```

```
      justify-content: space-between;
```

```
    }
```

```
    #heading{
```

```
      padding:10px;
```

```
      color:white;
```

```
      font-size:25px;
```

```
      font-style: italic
```

```
    }
```

```
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.container{
display:flex;
justify-content:center;
align-items: center
flex-direction:row;
}
#imgs{
margin-top:100px;
height: 300px;
width: 60vw;
}
#subhead{
font-size: 30px;
color: coral;
text-align: center;
text-decoration-line: underline;
}
th, td {
width:40vw;
padding: 15px;
}
```

```

th{
color:darkgreen;
text-decoration-line: underline;
}
td{
color:brown;
text-align:justify;
}

</style>
<body>
  <div class="header">
    <p id="heading">Deep learning fundus image analysis for early detection of
diabetic retinopathy</p>
    <div class="options">
      <p id="option1"><a href="index">Home</a></p>
      <p id="option1"><a href="register">Register</a></p>
      <p id="option1"><a href="login">Login</a></p>
      <p id="option1"><a href="query">Query</a></p>
    </div>
  </div>
  <div class="container">
    <br>
  </div>
  <div>
    <p id="subhead">About Project</p>
  </div>
  <table>
    <tr><th>Problem</th>
      <th>Solution</th>
    <tr><td><br>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.</td>
                                <td>Transfer learning has become one of the most common techniques
that has achieved better
                                performance in many areas, especially in medical image analysis and
classification. We used Transfer
                                Learning techniques like Inception V3,Resnet50,Xception V3 that are
more widely used as a transfer
                                learning method in medical image analysis and they are highly
effective.</td>
                                </table>
```

```
</body>
</html>
```

2.register.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of
Diabetic Retinopathy</title>
        <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
    </head>
    <style>
        html,body{
            margin:0;
            width:100%;
            height:100%;
            font-family:Roboto;
            display: flex;
            flex-direction: column;
        }
        .header{
            position: fixed;
            top: 0px;
            width:100%;
            height:100px;
            background-color:dimgrey;
```



```
display: flex;
flex-direction: row;
align-items: center;
justify-content: space-between;
}
#heading{
padding:10px;
color:white;
font-size:25px;
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.content{
margin:auto;
}
.frm{
margin-top: 100px;
border:2px solid black;
padding:30px;
border-radius:5px;
}
.inputs{
```

```

padding:10px;
border-radius:5px;
outline:0;
width:250px;
}
.btn{
cursor: pointer;
width:100%;
}
.si:hover{
color:white;
background-color:dimgrey;
}
img{
height:60px;
width:60px;
border-radius:50%;
}
</style>
<body>
<div class="header">
<p id="heading">Registration</p>
<div class="options">
<p id="option1"><a href="index">Home</a></p>
</div>
</div>
<div class="content">
<form class="frm" action="http://127.0.0.1:5000/afterreg" method="POST">
<h3><center></center></h3>
<input type="text" class="inputs" placeholder="Name"
name="name" required>
<input type="text" class="inputs" placeholder="Email id"
name="email" required><br><br>
<input type="text" class="inputs" placeholder="Phone Number"
name="phoneno" required>
<input type="text" class="inputs" placeholder="Location of
Residence" name="location" required><br><br>
<input type="password" class="inputs" placeholder="Password"
name="pwd" required>

```

```

        <input type="text" class="inputs" placeholder="Your favourite
Pet(Security Question)" name="secques" required><br><br>
        <select name="loginas" class="inputs btn" style="border:2px
solid";>

        <option value="user">User</option>
        </select><br><br>
        <button type="submit" class="inputs btn si" name="submit"
value="submit" >Register</button>

        <br><br>
        Already a user?<a href="login" style="color:blue;"> Login</a>

    </form>
</div>
<br>
<center><p style="color: blue;">{{pred}}</p></center>

</body>
</html>

```

3.login.html

```

<!DOCTYPE html>
<html>
    <head>
        <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of
Diabetic Retinopathy</title>
        <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
    </head>
    <style>
        html,body{
            margin:0;
            width:100%;
            height:100%;
            font-family:Roboto;
            display: flex;
            flex-direction: column;
        }
        .header{
            position: fixed;
            top: 0px;
            width:100%;

```

```
height:100px;
background-color:dimgrey;
display: flex;
flex-direction: row;
align-items: center;
justify-content: space-between;
}
#heading{
padding:10px;
color:white;
font-size:25px;
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.content{
margin:auto;
}
.frm{
margin-top: 30px;
border:2px solid black;
padding:30px;
border-radius:5px;
```

```

    }
    .inputs{
        padding:10px;
        border-radius:5px;
        outline:0;
        width:250px;
    }
    .btn{
        cursor: pointer;
        width:-webkit-fill-available;
    }
    .si:hover{
        color:white;
        background-color:dimgrey;
    }
    img{
        height:60px;
        width:60px;
        border-radius:50%;
    }
</style>
<body>
    <div class="header">
        <p id="heading">Login</p>
        <div class="options">
            <p id="option1"><a href="index">Home</a></p>
        </div>
    </div>

    <div class="content">
        <center><p style="color: blue;width:250px;margin-
top:100px">{{predict}}</p></center><br>
        <form class="frm" action="http://127.0.0.1:5000/afterlogin" method="POST">
            <h3><center></center></h3>
            <input type="text" class="inputs" placeholder="Phoneno"
name="phoneno" required><br><br>
            <input type="password" class="inputs" placeholder="Password"
name="pwd" required><br><br>
            <select name="loginas" class="inputs btn" style="border:2px

```

solid";>

<option value="admin">Admin</option>

<option value="user">User</option>

</select>

 Forgot

password?

<button type="submit" class="inputs btn si" >Login</button>

New user?

Register

</form>

</div>

</body>

</html>

4.fp.html

<!DOCTYPE html>

<html>

<head>

<title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of Diabetic Retinopathy</title>

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">

</head>

<style>

html,body{
margin:0;
width:100%;
height:100%;
font-family:Roboto;
display: flex;
flex-direction: column;
}

.header{
position: fixed;
top: 0px;
width:100%;
height:100px;

```
background-color:dimgrey;
display: flex;
flex-direction: row;
align-items: center;
justify-content: space-between;
}
#heading{
padding:10px;
color:white;
font-size:25px;
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.content{
margin:auto;
}
.frm{
margin-top: 100px;
border:2px solid black;
padding:30px;
border-radius:5px;
}
```

```

.inputs{
padding:10px;
border-radius:5px;
outline:0;
width:250px;
}
.btn{
cursor: pointer;
width:-webkit-fill-available;
}
.si:hover{
color:white;
background-color:dimgrey;
}
img{
height:90px;
width:90px;
border-radius:50%;
}
</style>
<body>
<div class="header">
<p id="heading">FORGOT PASSWORD</p>
<div class="options">
<p id="option1"><a href="index">Home</a></p>
</div>
</div>

<div class="content">
<center><p style="color: blue;margin-top: 100px;">{{predict}}</p></center><br>
<form class="frm" action="http://127.0.0.1:5000/afterfp" method="POST">
<h3><center></center></h3>
<input type="text" class="inputs" placeholder="Phone number"
name="phoneno" required><br><br>
<input type="text" class="inputs" placeholder="Your favourite
pet(Security Question)" name="secques" required><br><br>
<input type="password" class="inputs" placeholder="New
Password" name="npwd" required><br><br>
<input type="password" class="inputs" placeholder="Confirm

```



```
Password" name="cpwd" required><br><br>
                                <button type="submit" class="inputs btn si" >Submit</button>
                                <br><br>

                                </form>
                            </div>

                        </body>
</html>
```

5.prediction.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of
Diabetic Retinopathy</title>
        <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
    </head>
    <style>
        html,body{
            margin:0;
            width:100%;
            height:100%;
            font-family:Roboto;
            display: flex;
            flex-direction: column;
        }
        .header{
            position: fixed;
            top: 0px;
            width:100%;
            height:100px;
            background-color:dimgrey;
            display: flex;
            flex-direction: row;
            align-items: center;
            justify-content: space-between;
        }
    </style>
```

```
#heading{
padding:10px;
color:white;
font-size:25px;
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.container{
margin-top: 100px;
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
border:2px solid black;
}
.retinaimg{
margin-top:25px;
height:100px;
width: 100px;
border-radius:50%;
padding:20px;
}
```

```

        .stages{
            height:500px;
            width:600px
        }
        hr {
            width:100%;
            border:1px solid black;
            background:black;
        }
    </style>
<body>
    <div class="header">
        <p id="heading">Prediction</p>
        <div class="options">
            <p id="option1"><a href="index">Home</a></p>
            <p id="option1"><a href="locsugges">Hospitals</a></p>
            <p id="option1"><a href="modify">Modify</a></p>
            <p id="option1"><a href="logout">Logout</a></p>
        </div>
    </div>
    <div class="container">
        
        <form class="frm" action="/afterpred" method="POST" enctype="multipart/form-
data">
            <input type="file" name="pfile" placeholder="Choose file"
accept="image/*" required>
            <button type="submit" name="submit" value="submit" >Submit</button>
        </form>
    </div>
    <br><br>
    <center><p>Diabetic Retinopathy Classification is: {{prediction}}</p></center>
    <hr>
    <center></center>

    </body>
</html>

```

6.query1.html

```
<!DOCTYPE html>
```

```
<html>
  <head>
    <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of
Diabetic Retinopathy</title>
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
    <script>
      function calc()
      {
        var d = new Date();
        var n = d.getTime();
        document.getElementById('timestmp').value=n;
      }
    </script>
  </head>
  <style>
    html,body{
    margin:0;
    width:100%;
    height:100%;
    font-family:Roboto;
    display: flex;
    flex-direction: column;
    }
    .header{
    position: fixed;
    top: 0px;
    width:100%;
    height:100px;
    background-color:dimgrey;
    display: flex;
    flex-direction: row;
    align-items: center;
    justify-content: space-between;
    }
    #heading{
    padding:10px;
    color:white;
    font-size:25px;
    font-style: italic
```

```
}  
#option1{  
margin-right:2px;  
padding:10px;  
color:white;  
font-size:20px;  
}  
.options{  
display: flex;  
}  
a{  
text-decoration:none;  
color:white;  
}  
#option1:hover{  
border:1px solid white;  
border-radius:10px;  
background-color:black;  
}  
.container{  
display:flex;  
justify-content:center;  
align-items: center  
flex-direction:row;  
}  
#imgs{  
margin-top:100px;  
height: 300px;  
width: 60vw;  
}  
.frm{  
position: fixed;  
left: 0;  
bottom: 0;  
width: 100%;  
border:2px solid black;  
padding:30px;  
border-radius:5px;  
}
```

```

        .inputs{
        padding:10px;
        border-radius:5px;
        outline:0;
        width:300px;
        }
        .btn{
        cursor: pointer;
        width:300px;
        }
        .si:hover{
                color:white;
                background-color:dimgrey;
        }
        #subhead{
        font-size: 30px;
        color: coral;
        text-align: center;
        text-decoration-line: underline;
        }
        th, td {
        width:40vw;
        padding: 15px;
        }
        th{
        color:darkgreen;
        text-decoration-line: underline;
        }
        td{
        color:brown;
        text-align:justify;
        }
        .disp{
        height:62vh;
        }

</style>
<body>
    <div class="header">

```

```

        <p id="heading">Queries and Responses</p>
        <div class="options">
            <p id="option1"><a href="index">Home</a></p>
        </div>
    </div>
    <div class="container">
        <center><p style="color: blue;margin-top: 100px;">{{predict}}</p></center><br>
    </div>
    <div class="disp" style="margin-top:-7px; overflow-y:scroll;">
        {% set i=0 %}
        {% for i in range(data|length)%}
            {% if data[i]=="user" :%}
                <p>User:{{data[i+1] }}</p>
                <p>Query:{{data[i+2] }}</p>
                {% set i=i+1 %}
                <hr>
            {% endif %}
            {% if data[i]=="admin": %}
                <p style="text-align:right; margin-
right:10px;color:deeppink;">User:{{data[i+1]}}</p>
                <p style="text-align:right;margin-
right:10px;color:deeppink;">Response:{{data[i+2]}}</p>
                {% set i=i+1 %}
                <hr>
            {% endif %}
            {% endfor %}
        </div>
    <div>
        <form class="frm" action="http://127.0.0.1:5000/afterquery" method="POST"
onsubmit="calc()">
            <input type="hidden" id="timstmp" name="timestamp" value="0">
            <input type="hidden" name="who" value="user">
            <input type="text" class="inputs" placeholder="Phone number"
name="phoneno" required>
            <input type="text" class="inputs" placeholder="Query"
name="query" required>
            <button type="submit" class="inputs btn si" >Submit</button>
            <br><br>
        </form>
    </div>

```

</div>

</body>

</html>

7.modify.html

<!DOCTYPE html>

<html>

<head>

<title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of Diabetic Retinopathy</title>

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">

</head>

<style>

html,body{

margin:0;

width:100%;

height:100%;

font-family:Roboto;

display: flex;

flex-direction: column;

}

.header{

position: fixed;

top: 0px;

width:100%;

height:100px;

background-color:dimgrey;

display: flex;

flex-direction: row;

align-items: center;

justify-content: space-between;

}

#heading{

padding:10px;

color:white;

font-size:25px;


```
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.content{
margin:auto;
}
.frm{
margin-top: 100px;
border:2px solid black;
padding:30px;
border-radius:5px;
}
.inputs{
padding:10px;
border-radius:5px;
outline:0;
width:250px;
}
.btn{
cursor: pointer;
width:-webkit-fill-available;
}
```

```

        .si:hover{
            color:white;
            background-color:dimgrey;
        }
        img{
            height:60px;
            width:60px;
            border-radius:50%;
        }
    </style>
<body>
    <div class="header">
        <p id="heading">Location change</p>
        <div class="options">
            <p id="option1"><a href="index">Home</a></p>
            <p id="option1"><a href="prediction">Back</a></p>
            <p id="option1"><a href="logout">Logout</a></p>
        </div>
    </div>

    <div class="content">
        <center><p style="color: blue;width:250px;">{{predict}}</p></center><br>
        <form class="frm" action="http://127.0.0.1:5000/aftermodify" method="POST">

            <input type="password" class="inputs" placeholder="password"
name="pwd" required><br><br>
            <input type="text" class="inputs" placeholder="Enter new
location" name="location" required><br><br>

            <button type="submit" class="inputs btn si" >Modify</button>
            <br><br>

        </form>
    </div>

</body>
</html>

```

8.locsugges.html

<!DOCTYPE html>

<html>

 <head>

 <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of Diabetic Retinopathy</title>

 <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">

 </head>

 <style>

 html,body{
 margin:0;
 width:100%;
 height:100%;
 font-family:Roboto;
 display: flex;
 flex-direction: column;
 }

 th{

padding-top: 10px;
padding-bottom: 20px;
padding-left: 30px;
padding-right: 40px;
color:black;

}

td {

padding-top: 10px;
padding-bottom: 20px;
padding-left: 30px;
padding-right: 40px;
color:teal;

}

 .header{
 position: fixed;
 top: 0px;
 width:100%;
 height:100px;
 background-color:dimgrey;

```
display: flex;
flex-direction: row;
align-items: center;
justify-content: space-between;
}
#heading{
padding:10px;
color:white;
font-size:25px;
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.content{
margin:auto;
}
.frm{
margin-top: 100px;
border:2px solid black;
padding:30px;
border-radius:5px;
}
.inputs{
```

```
padding:10px;
border-radius:5px;
outline:0;
width:250px;
}
.btn{
cursor: pointer;
width:-webkit-fill-available;
}
.si:hover{
color:white;
background-color:dimgrey;
}
```

```
</style>
```

```
<body>
```

```
<div class="header">
```

```
<p id="heading">Hospitals Near You</p>
```

```
<div class="options">
```

```
<p id="option1"><a href="index">Home</a></p>
```

```
<p id="option1"><a href="prediction">Back</a></p>
```

```
</div>
```

```
</div>
```

```
<div class="content">
```

```
<table>
```

```
{% set i=0 %}
```

```
<tr>
```

```
<th>HOSPITALS</th>
```

```
<th>LOCATIONS</th>
```

```
<th>CONTACTS</th>
```

```
</tr>
```

```
{% for i in range(hospital|length)%}
```

```
<tr><td>{{hospital[i]}}</td>
```

```
<td>{{location[i] }}</td>
```

```
<td>{{contact[i] }}</td></tr>
```

```
{% set i=i+1 %}
```

```
{% endfor %}
```

```
</table>
```

</div>

</body>

</html>

9.admin.html

<!DOCTYPE html>

<html>

<head>

<title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of Diabetic Retinopathy</title>

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">

</head>

<style>

html,body{

margin:0;

width:100%;

height:100%;

font-family:Roboto;

display: flex;

flex-direction: column;

}

.header{

position: fixed;

top: 0px;

width:100%;

height:100px;

background-color:dimgrey;

display: flex;

flex-direction: row;

align-items: center;

justify-content: space-between;

}

#heading{

padding:10px;

color:white;

font-size:25px;

```
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.container{
margin-top: 100px;
display: flex;
flex-direction: row;
align-items: center;
justify-content: center;
border:2px solid black;
}
#imgs{
margin-top:50px;
height: 300px;
width: 300px;
}
#subhead{
font-size: 30px;
color: coral;
text-align: center;
text-decoration-line: underline;
}
```

```

th, td {
width:40vw;
padding: 15px;
}
th{
color:darkgreen;
text-decoration-line: underline;
}
td{
color:brown;
text-align:justify;
}
</style>
<body>
<div class="header">
<p id="heading">Welcome Admin</p>
<div class="options">
<p id="option1"><a href="respond">Respond Queries</a></p>
<p id="option1"><a href="uploc">Update locations</a></p>
<p id="option1"><a href="logout">Logout</a></p>
</div>
</div>
<div class="container">
<br>
</div>
<div>
<p id="subhead">About Project</p>
</div>
<table>
<tr><th>Problem</th>
<th>Solution</th>
<tr><td><br>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.</td> |
| <td>Transfer learning has become one of the most common techniques that has achieved better performance in many areas, especially in medical image analysis and classification. We used Transfer Learning techniques like Inception V3, Resnet50, Xception V3 that are more widely used as a transfer learning method in medical image analysis and they are highly effective.</td> |

</body>
</html>

10.respond.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of Diabetic Retinopathy</title>
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
    <script>
      function calc()
      {
        var d = new Date();
        var n = d.getTime();
        document.getElementById('timestmp').value=n;
      }
    </script>
  </head>
  <style>
    html,body{
      margin:0;
      width:100%;
      height:100%;
      font-family:Roboto;
```

```
display: flex;
flex-direction: column;
}
.header{
position: fixed;
top: 0px;
width:100%;
height:100px;
background-color:dimgrey;
display: flex;
flex-direction: row;
align-items: center;
justify-content: space-between;
}
#heading{
padding:10px;
color:white;
font-size:25px;
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
}
.container{
```

```
display:flex;
justify-content:center;
align-items: center
flex-direction:row;
}
#imgs{
margin-top:100px;
height: 300px;
width: 60vw;
}
.frm{
position: fixed;
left: 0;
bottom: 0;
width: 100%;
border:2px solid black;
padding:30px;
border-radius:5px;
}
.inputs{
padding:10px;
border-radius:5px;
outline:0;
width:300px;
}
.btn{
cursor: pointer;
width:300px;
}
.si:hover{
        color:white;
        background-color:dimgrey;
}
#subhead{
font-size: 30px;
color: coral;
text-align: center;
text-decoration-line: underline;
}
```

```

th, td {
width:40vw;
padding: 15px;
}
th{
color:darkgreen;
text-decoration-line: underline;
}
td{
color:brown;
text-align:justify;
}
.disp{
height:62vh;
}

```

</style>

<body>

<div class="header">

<p id="heading">Queries and Responses</p>

<div class="options">

<p id="option1">Back</p>

<p id="option1">Logout</p>

</div>

</div>

<div class="container">

<center><p style="color: blue;margin-top: 100px;">{{predict}}</p></center>

</div>

<div class="disp" style="margin-top:-7px; overflow-y:scroll;">

{% set i=0 %}

{% for i in range(data|length)%}

{% if data[i]=="user" :%}

<p>User:{{data[i+1] }}</p>

<p>Query:{{data[i+2] }}</p>

{% set i=i+1 %}

<hr>

{% endif %}

{% if data[i]=="admin": %}

<p style="text-align:right; margin-

```

right:10px;color:deeppink;">User:{{data[i+1]}}</p>
        <p style="text-align:right;margin-
right:10px;color:deeppink;">Response:{{data[i+2]}}</p>
        {% set i=i+1 %}
        <hr>
        {% endif %}
        {% endfor %}
    </div>
    <div>
        <form class="frm" action="http://127.0.0.1:5000/afterrespond" method="POST"
onsubmit="calc()">
            <input type="hidden" id="timstmp" name="timestamp" value="0">
            <input type="hidden" name="who" value="admin">
            <input type="text" class="inputs" placeholder="Requestof"
name="phoneno" required>
            <input type="text" class="inputs" placeholder="Response"
name="query" required>
            <button type="submit" class="inputs btn si" >Post</button>
            <br><br>
        </form>
    </div>

</body>
</html>

```

11.uploc.html

```

<!DOCTYPE html>
<html>
    <head>
        <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of
Diabetic Retinopathy</title>
        <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
    </head>
    <style>
        html,body{
            margin:0;
            width:100%;

```

```
height:100%;
font-family:Roboto;
display: flex;
flex-direction: column;
}
.header{
position: fixed;
top: 0px;
width:100%;
height:100px;
background-color:dimgrey;
display: flex;
flex-direction: row;
align-items: center;
justify-content: space-between;
}
#heading{
padding:10px;
color:white;
font-size:25px;
font-style: italic
}
#option1{
margin-right:2px;
padding:10px;
color:white;
font-size:20px;
}
.options{
display: flex;
}
a{
text-decoration:none;
color:white;
}
#option1:hover{
border:1px solid white;
border-radius:10px;
background-color:black;
```

```

    }
    .content{
    margin:auto;
    }
    .frm{
    margin-top: 100px;
    border:2px solid black;
    padding:30px;
    border-radius:5px;
    }
    .inputs{
    padding:10px;
    border-radius:5px;
    outline:0;
    width:250px;
    }
    .btn{
    cursor: pointer;
    width:-webkit-fill-available;
    }
    .si:hover{
        color:white;
        background-color:dimgrey;
    }
    img{
    height:60px;
    width:60px;
    border-radius:50%;
    }
</style>
<body>
    <div class="header">
        <p id="heading">Admin update</p>
        <div class="options">
            <p id="option1"><a href="index">Home</a></p>
            <p id="option1"><a href="admin">Back</a><p>

        </div>
    </div>
</div>

```

```

<div class="content">
  <center><p style="color: blue;width:250px;">{{predict}}</p></center><br>
    <form class="frm" action="http://127.0.0.1:5000/afteruploc" method="POST">

      <input type="text" class="inputs" placeholder="Enter the location
of hospital" name="loch" required><br><br>
      <input type="text" class="inputs" placeholder="Enter the hospital
name" name="hname" required><br><br>
      <input type="text" class="inputs" placeholder="location"
name="lname" required><br><br>
      <input type="text" class="inputs" placeholder="phone no"
name="contact" required><br><br>
      <button type="submit" class="inputs btn si" >UPDATE</button>
      <br><br>

    </form>
  </div>

</body>
</html>

```

12.logout.html

```

<!DOCTYPE html>
<html>
  <head>
    <title>PNT2022TMID06905 Deep learning fundus image analysis for early detection of
Diabetic Retinopathy</title>
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto">
  </head>
  <style>
    html,body{
      margin:0;
      width:100%;
      height:100%;
      font-family:Roboto;
      display: flex;
      flex-direction: column;

```



```
}  
.header{  
position: fixed;  
top: 0px;  
width:100%;  
height:100px;  
background-color:dimgrey;  
display: flex;  
flex-direction: row;  
align-items: center;  
justify-content: space-between;  
  
}  
.log  
{  
color:black;  
font-size:50px;  
  
}  
#heading{  
padding:10px;  
color:white;  
font-size:25px;  
font-style: italic  
}  
#option1{  
margin-right:2px;  
padding:10px;  
color:white;  
font-size:20px;  
}  
.options{  
display: flex;  
}  
a{  
text-decoration:none;  
color:white;  
}  
#option1:hover{
```

```
border:1px solid white;
border-radius:10px;
background-color:black;
}
.content{
margin:auto;
}
.frm{
margin-top: 100px;
border:2px solid black;
padding:30px;
border-radius:5px;
}
.inputs{
padding:10px;
border-radius:5px;
outline:0;
width:250px;
}
.btn{
cursor: pointer;
width:50%;

}
.si:hover{
        color:white;
        background-color:dimgrey;
    }
img{
height:60px;
width:60px;
border-radius:50%;
}
</style>
```

```
<body>
```

```
<div class="header">
```

```
<p id="heading">Diabetic retinopathy</p>
```

```
<div class="options">
```

```
<p id="option1"><a href="index">Home</a></p>
```

```

        <p id="option1"><a href="login">Login </a></p>
    </div>
</div>
<div class="content">
    <p class="log">Successfully logout</p>
    <p> Login for more information</p><br>

</div>

</body>
</html>

```

13.cloudcreation.ipynb

```

from cloudant.client import Cloudant
pip install cloudant
client=Cloudant.iam('username','apikey',connect=True)
my_database=client.create_database('my_database')

```

14.querydb.ipynb

```

from cloudant.client import Cloudant
client=Cloudant.iam('username','apikey',connect=True)
my_database=client.create_database('my_database_query')

```

15.trainingmodel.ipynb

```

imageSize = [299,299]
trainPath = r"E:\GCT\SEM 7\nalaiyathiran\dataset\preprocessed dataset\preprocessed dataset\training"
testPath = r"E:\GCT\SEM 7\nalaiyathiran\dataset\preprocessed dataset\preprocessed dataset\testing"
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img
from tensorflow.keras.applications.xception import Xception,preprocess_input
from glob import glob
import numpy as np
import matplotlib.pyplot as plt
train_datagen = ImageDataGenerator(rescale=1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip
= True)
test_datagen = ImageDataGenerator(rescale = 1./255)

```

```

training_set = train_datagen.flow_from_directory(r"E:\GCT\SEM 7\nalaiyathiran\dataset\preprocessed
dataset\preprocessed dataset\training",
                                                target_size = (299,299),
                                                batch_size = 32,
                                                class_mode = 'categorical')
test_set = test_datagen.flow_from_directory(r"E:\GCT\SEM 7\nalaiyathiran\dataset\preprocessed
dataset\preprocessed dataset\training",
                                            target_size = (299,299),
                                            batch_size = 32,
                                            class_mode = 'categorical')
xception=Xception(input_shape = imageSize + [3], weights = 'imagenet', include_top= False)
for layer in xception.layers:
    layer.trainable = False
x = Flatten()(xception.output)
prediction = Dense(5, activation = 'softmax')(x)
model = Model(inputs=xception.input, outputs=prediction)
model.summary()
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
r = model.fit_generator(training_set,
                        validation_data=test_set,
                        epochs=30,
                        steps_per_epoch=len(training_set)//32,
                        validation_steps=len(test_set)//32
                        )
model.save('Updated-Xception-diabetic-retinopathy.h5')

```

16.app.ipynb

```

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 cloudant.client import Cloudant
from werkzeug.utils import secure_filename
from flask import Flask, request, render_template, redirect, url_for, session
app=Flask(__name__)

client=Cloudant.iam('username','apikey',connect=True)
my_database = client['my_database']

```

```
app.secret_key="SECRET_KEY"
model=load_model(r"Updated-Xception-diabetic-retinopathy.h5")
```

```
image_folder=os.path.join('static','images')
app.config['UPLOAD_FOLDER'] = image_folder
```

```
@app.route('/')
def index():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'drimage.jpg')
    return render_template('index.html',image=full_filename)
```

```
@app.route('/index')
def home():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'drimage.jpg')
    return render_template('index.html',image=full_filename)
```

```
@app.route('/register')
def register():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'registerimg.jpg')
    return render_template('register.html',image=full_filename)
```

```
@app.route('/afterreg',methods=['POST','GET'])
def afterreg():
    x=[x for x in request.form.values()]
    data={
        '_id':x[2],
        'name':x[0],
        'pwd':x[4],
        'email':x[1],
        'location':x[3],
        'securityquestion':x[5],
        'logintype':x[6]
    }
    query={'_id':{'$eq':data['_id']}}
    docs=my_database.get_query_result(query)
    if(len(docs.all())==0):
        url=my_database.create_document(data)
```

```
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'loginimg.jpg')
    return render_template('login.html',predict="Registration successfull please login using your
credentials",image=full_filename)
```

```
else:
```

```
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'registerimg.jpg')
    return render_template('register.html',pred="You are already a member login using your
credentials",image=full_filename)
```

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

```
def login():
```

```
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'loginimg.jpg')
    return render_template('login.html',image=full_filename)
```

```
@app.route('/afterlogin', methods=['POST','GET'])
```

```
def afterlogin():
```

```
    user=request.form['phoneno']
    session['pn']=user
    passw=request.form['pwd']
    lgnas=request.form['loginas']
```

```
    query={'_id':{'$eq':user}}
    docs=my_database.get_query_result(query)
```

```
if(len(docs.all())==0):
```

```
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'loginimg.jpg')
    return render_template('login.html',predict="Phone number/id not found",image=full_filename)
```

```
else:
```

```
    if((user==docs[0][0]['_id'] and passw==docs[0][0]['pwd'] and lgnas==docs[0][0]['logintype'] )):
```

```
        if(docs[0][0]['logintype']=='user'):
```

```
            full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'retina.jpg')
            full_filename1 = os.path.join(app.config['UPLOAD_FOLDER'], 'image6.png')
            return render_template('prediction.html',image=full_filename,image2=full_filename1)
```

```
        if(docs[0][0]['logintype']=='admin'):
```

```
            full_filename2 = os.path.join(app.config['UPLOAD_FOLDER'], 'adminimg.png')
            return render_template('admin.html',image=full_filename2)
```

```
    if(lgnas!=docs[0][0]['logintype']):
```

```
        full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'loginimg.jpg')
        return render_template('login.html',image=full_filename,predict="Incorrect Logintype")
```

```
    if(passw!=docs[0][0]['pwd']):
```

```
full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'loginimg.jpg')
return render_template('login.html',image=full_filename,predict="Incorrect password")
```

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

```
def respond():
```

```
    my_database_query = client['my_database_query']
```

```
    dt=[]
```

```
    for document in my_database_query:
```

```
        dt.append(document['who'])
```

```
        dt.append(document['phoneno'])
```

```
        dt.append(document['query'])
```

```
    return render_template('respond.html',data=dt)
```

```
@app.route('/afterrespond', methods=['POST','GET'])
```

```
def afterrespond():
```

```
    my_database_query = client['my_database_query']
```

```
    x=[x for x in request.form.values()]
```

```
    data1={
```

```
        '_id':x[0],
```

```
        'who':x[1],
```

```
        'phoneno':x[2],
```

```
        'query':x[3]
```

```
    }
```

```
    query={'_id':{'$eq':data1['_id']}}
```

```
    docs=my_database_query.get_query_result(query)
```

```
    if(len(docs.all())==0):
```

```
        url=my_database_query.create_document(data1)
```

```
        my_database_query = client['my_database_query']
```

```
        dt=[]
```

```
        for document in my_database_query:
```

```
            dt.append(document['who'])
```

```
            dt.append(document['phoneno'])
```

```
            dt.append(document['query'])
```

```
        return render_template('respond.html',predict="Response posted Successfully",data=dt)
```

```
    else:
```

```
        url=my_database_query.create_document(data1)
```

```
        my_database_query = client['my_database_query']
```

```

dt=[]
for document in my_database_query:
    dt.append(document['who'])
    dt.append(document['phoneno'])
    dt.append(document['query'])
return render_template('respond.html',predict="Response posted Successfully",data=dt)

```

```

@app.route('/fp')
def fp():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'],'forgotpw.png')
    return render_template('fp.html',image=full_filename)

```

```

@app.route('/afterfp', methods=['POST','GET'])
def afterfp():
    pn=request.form['phoneno']
    securityques=request.form['secques']
    npassw=request.form['npwd']
    cpassw=request.form['cpwd']

```

```

docs=my_database[pn]

```

```

if(npassw==cpassw and securityques==docs['securityquestion']):
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'],'loginimg.jpg')
    docs['pwd'] = cpassw
    docs.save()
    return render_template('login.html',predict="Successfully updated",image=full_filename)
if(securityques!=docs['securityquestion']):
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'],'forgotpw.png')
    return render_template('fp.html',predict="Incorrect answer to security question",image=full_filename)
if(npassw!=cpassw):
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'],'forgotpw.png')
    return render_template('fp.html',predict="New and confirm password does not
match",image=full_filename)

```



```

@app.route('/prediction')
def prediction():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'retina.jpg')
    full_filename1 = os.path.join(app.config['UPLOAD_FOLDER'], 'image6.png')
    return render_template('prediction.html',image=full_filename,image2=full_filename1)

@app.route('/afterpred',methods=["GET","POST"])
def afterpred():
    if request.method=="POST":
        full_filename2 = os.path.join(app.config['UPLOAD_FOLDER'], 'retina.jpg')
        full_filename1 = os.path.join(app.config['UPLOAD_FOLDER'], 'image6.png')
        f=request.files['pfile']
        filepath=os.path.join('static',uploads,f.filename)
        f.save(filepath)
        img=image.load_img(filepath,target_size=(299,299))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)
        img_data=preprocess_input(x)
        prediction=np.argmax(model.predict(img_data),axis=1)
        index=["No DR","Mild DR","Moderate DR","Severe DR","Proliferate DR"]
        result=str(index[prediction[0]])
        return
    render_template('prediction.html',prediction=result,image=full_filename2,image2=full_filename1)
    else:
        full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'loginimg.jpg')
        return render_template('login.html',pred="Please login using your credentials",image=full_filename)

@app.route('/query')
def query():
    my_database_query = client['my_database_query']

    dt=[]
    for document in my_database_query:
        dt.append(document['who'])
        dt.append(document['phoneno'])
        dt.append(document['query'])
    return render_template('query1.html',data=dt)

```

```

@app.route('/afterquery', methods=['POST','GET'])
def afterquery():
    my_database_query = client['my_database_query']
    x=[x for x in request.form.values()]
    data1={
        '_id':x[0],
        'who':x[1],
        'phoneno':x[2],
        'query':x[3],
    }
    query={'_id':{'$eq':data1['_id']}}
    docs=my_database_query.get_query_result(query)
    if(len(docs.all())==0):
        url=my_database_query.create_document(data1)
        my_database_query = client['my_database_query']
        dt=[]
        for document in my_database_query:
            dt.append(document['who'])
            dt.append(document['phoneno'])
            dt.append(document['query'])
        return render_template('query1.html',predict="Query submitted Successfully",data=dt)
    else:
        url=my_database_query.create_document(data1)
        my_database_query = client['my_database_query']
        dt=[]
        for document in my_database_query:
            dt.append(document['who'])
            dt.append(document['phoneno'])
            dt.append(document['query'])
        return render_template('query1.html',predict="Query submitted Successfully",data=dt)

@app.route('/admin')
def admin():
    full_filename2 = os.path.join(app.config['UPLOAD_FOLDER'], 'adminimg.png')
    return render_template('admin.html',image=full_filename2)

@app.route('/locsugges')
def locsugess():

```

```

dbl=client['db1']
dbu=client['my_database']

pn=session['pn']
doc1=dbu[pn]
location=doc1["location"]

docs=dbl[location.upper()]
hospital=[]
location=[]
contact=[]
for h in range(len(docs["hospitals"])):
    hospital.append(docs['hospitals'][h])
for c in range(len(docs["hospitals"])):
    contact.append(docs['contacts'][c])
for l in range(len(docs["hospitals"])):
    location.append(docs['locations'][l])
full_filename = os.path.join(app.config['UPLOAD_FOLDER'])
return render_template('locsugges.html',hospital=hospital,location=location,contact=contact)

@app.route('/uploc')
def uploc():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'])
    return render_template('uploc.html')

@app.route('/afteruploc',methods=["GET","POST"])
def afteruploc():
    loc=request.form['loch']
    hname=request.form['hname']
    lname=request.form['lname']
    contact=request.form['contact']
    dbl=client['db1']
    docs=dbl[loc.upper()]
    hn=hname.strip().upper()
    count=0
    for i in range(len(docs['hospitals'])):
        if docs['hospitals'][i].strip().upper()==hn:
            docs['contacts'][i]=contact
            docs['locations'][i]=lname
            count=1

```

```

        docs.save()
        break
    if(count==0):
        docs['hospitals'].append(hname)
        docs['locations'].append(lname)
        docs['contacts'].append(contact)
        docs.save()
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'])
    return render_template('uploc.html',predict="Updated or added successfully")

@app.route('/modify',methods=["GET","POST"])
def modify():
    full_filename = os.path.join(app.config['UPLOAD_FOLDER'])
    return render_template('modify.html',predict="Please enter your new location")

@app.route('/aftermodify',methods=["GET","POST"])
def aftermodify():
    if request.method=="POST":
        user=session['pn']
        pwd=request.form['pwd']
        nloc=request.form['location']
        docs=my_database[user]
        if (pwd==docs['pwd']):
            full_filename = os.path.join(app.config['UPLOAD_FOLDER'],'pimg.jpg')
            docs['location'] = nloc
            docs.save()
            return render_template('modify.html',predict="Successfully updated")
        else:
            full_filename = os.path.join(app.config['UPLOAD_FOLDER'], 'retina.jpg')
            full_filename1 = os.path.join(app.config['UPLOAD_FOLDER'], 'image6.png')
            return render_template('prediction.html',image=full_filename,image2=full_filename1)

@app.route('/logout')
def logout():
    session.pop('pn', None)
    return render_template('logout.html',)

```

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

GITHUB LINK:

<https://github.com/IBM-EPBL/IBM-Project-3912-1658671063>

PROJECT DEMO LINK:

<https://drive.google.com/file/d/1Zu-Xt7MyZ0LKrCcnTeSzZuto2jSC0Dox/view?usp=sharing>