

Project Report

TEAM ID:-PNT2022TMID34743

**Deep Learning Fundus Image Analysis for Early Detection of
Diabetic Retinopathy**

Team Leader : SAKTHI PRIYA E K

Team member : PAULIN SHEFIA C P

Team member : SUSHMI J

Team member : VINISHA V

Industry Mentor Name:-Shanthi

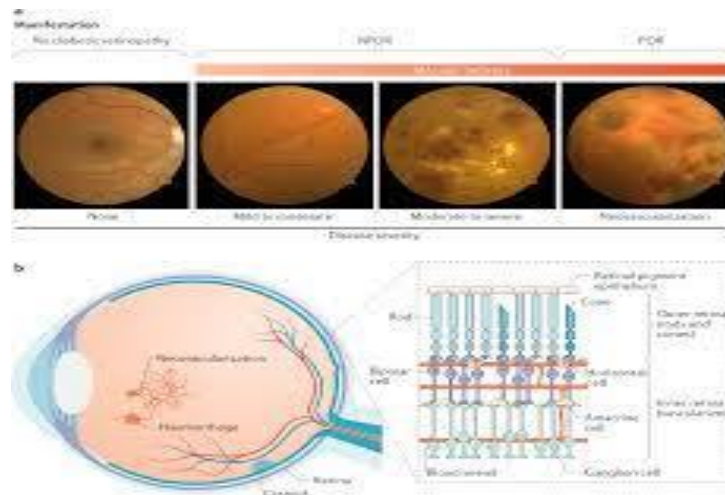
Faculty Mentor Name:-J.ARUL KING

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

INTRODUCTION:-

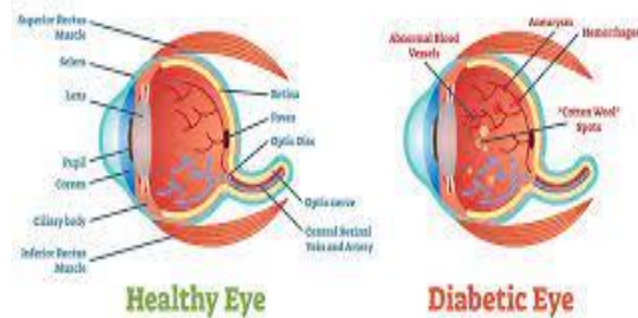
Project Overview:-

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.



Purpose:-

An eye exam is a quick, simple, and painless way for an ophthalmologist to check a person's eye health. The test allows them to check for eye problems early on, when they are easier to treat and before they cause vision complications. Eye screening is an important part of diabetes care. Untreated diabetic retinopathy is one of the most common causes of sight loss. When the condition is caught early, treatment is effective at reducing or preventing damage to the sight.



LITERATURE SURVEY:-

AUTHOR	TECHNOLOGY	DESCRIPTION	ADVANTAGES	DISADVANTAGES
Shin ES, SorensonCM, Sheibani N. Diabetes and retinal vascular dysfunction. J Ophthalmic Vis Res. 2014; 9: 362–373	Fractal Dimensional Analysis of Optical Coherence Tomography Angiography in Eyes With Diabetic Retinopathy	It used fractal dimensional analysis to analyze retinal vascular disease burden in eyes with diabetic retinopathy using spectral-domain optical coherence tomography angiography (OCTA).	The advent of OCTA analysis provide high resolution images that allow the visualization of microvascular abnormalities, and also are well-suited to mathematical analysis of each vascular layer.	A small sample size and persistence of imaging artifacts, which may reduce the precision of fractal analysis.
Wenbo Zhang 1 , Hua Liu, Mohamed Al-Shabrawey, Robert W Caldwell, Ruth B Caldwell.	Comparing retinopathy lesions in scanning laser ophthalmoscopy and colour fundus photography.	It evaluate the detection of different lesions of diabetic retinopathy in scanning laser ophthalmoscopy compared to color fundus with retina.	Features of DR including haemorrhages, microaneurysms, intraretinal microvascular abnormalities, and neo vascularization were analysed.	Due to high efficiency it's not applicable for advanced level.Lower response to the application model.
Lucy Q. Shen, Angie Child, Griffin M. Weber, Judah	Rosiglitazone and Delayed Onset of Proliferative Diabetic Retinopathy.	Evaluate whether rosiglitazone maleate, an oral peroxisome-proliferating activated receptor	All visual acuity measurements were bestcorrected and conducted on Early Treatment Diabetic	Rosiglitazone may delay the onset of PDR,possibly because of its antiangiogenic activity

Folkman, Lloyd Paul AielloPublish Year: 2008		agonist and oral insulin sensitizing agent with potential antiangiogenic activity,delays onset of proliferative diabetic retinopathy (PDR).	Retinopathy Study (ETDRS) visual acuity charts	
Osakada F, Ikeda H, Sasai Y, Takahashi M (2009)	Early retinal differentiation of human pluripotent stem cells in microwell suspension cultures	It develop a microwell suspension platform for the adaption of attached stem cell differentiation protocols into mixed suspension culture	Only major capital expenditure for standard cell culture laboratories is a relatively cheap shaking platform.	Shaking microwells is that they do not include many potentially beneficial functionalities associated with suspension bioreactors such as pH,O ₂ control and medium feeding regimes.

References:-

E. S. Shin, C. M. Sorenson, and N. Sheibani, “Diabetes and retinal vascular dysfunction,” *Journal of Ophthalmic & Vision Research*, vol.9, no. 3, pp. 362–373, 2014.

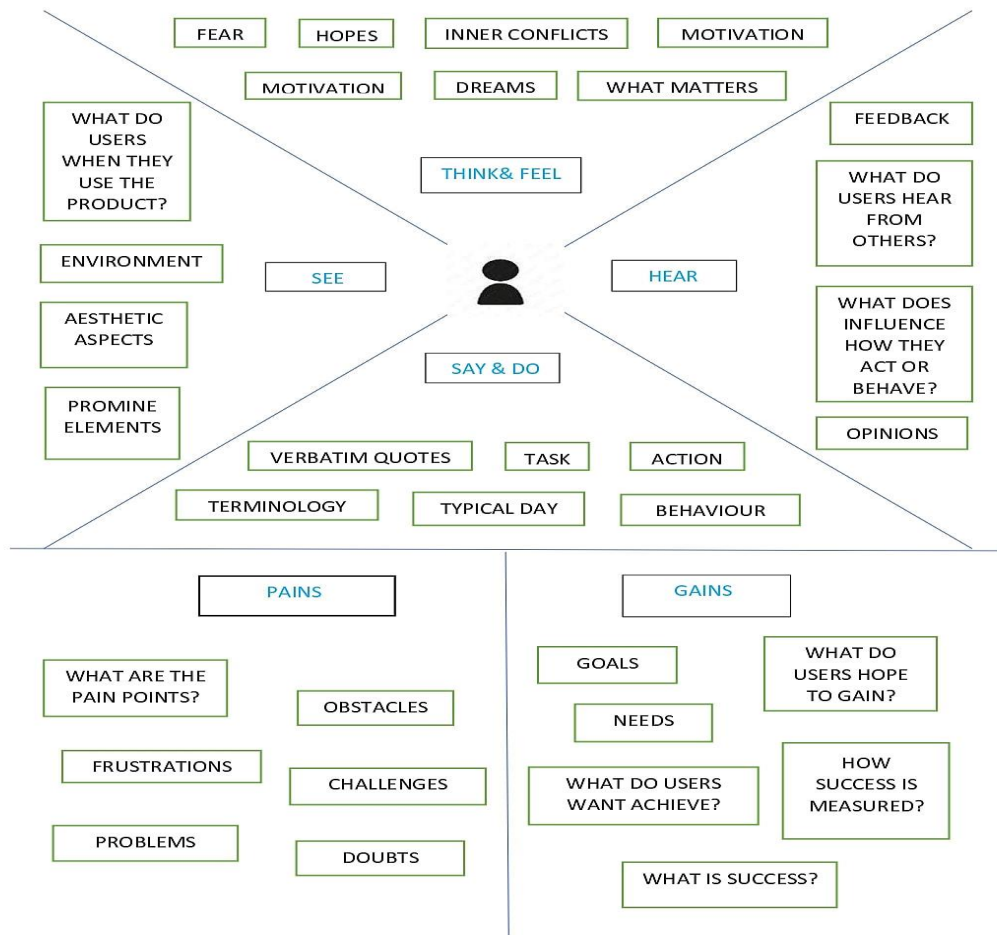
W. Zhang, H. Liu, M. Al-Shabrawey, R. W. Caldwell, and R. B. Caldwell, “Inflammation and diabetic retinal microvascular complications,” *Journal of Cardiovascular Disease Research*, vol. 2, no. 2, pp. 96–103, 2011.

L. Q. Shen, A. Child, G. M. Weber, J. Folkman, and L. P. Aiello, “Rosiglitazone and delayed onset of proliferative diabetic retinopathy,” *Archives of Ophthalmology*, vol. 126, no. 6, pp. 793–799, 2008.

F. Osakada, H. Ikeda, Y. Sasai, and M. Takahashi, “Stepwise differentiation of pluripotent stem cells into retinal cells,” *Nature Protocols*, vol. 4, no. 6, pp. 811–824, 2009.

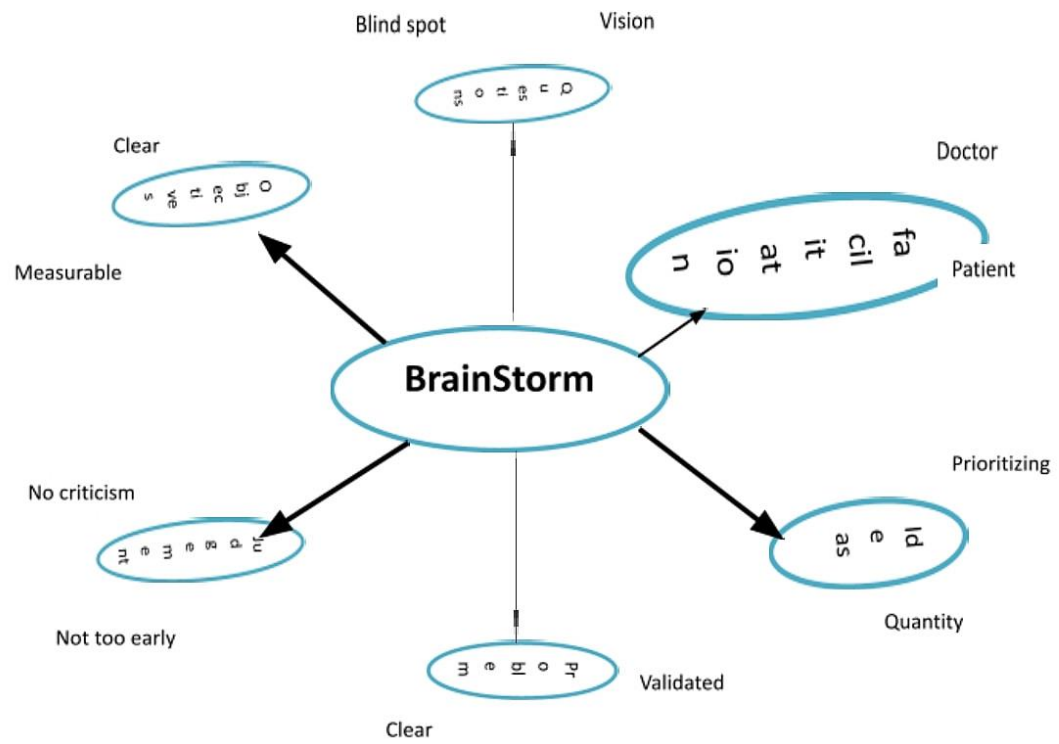
IDEATION & PROPOSED SOLUTION:-

Empathy Map Canvas:-



Brainstorming:-

Mindmapping



PROBLEM STATEMENT

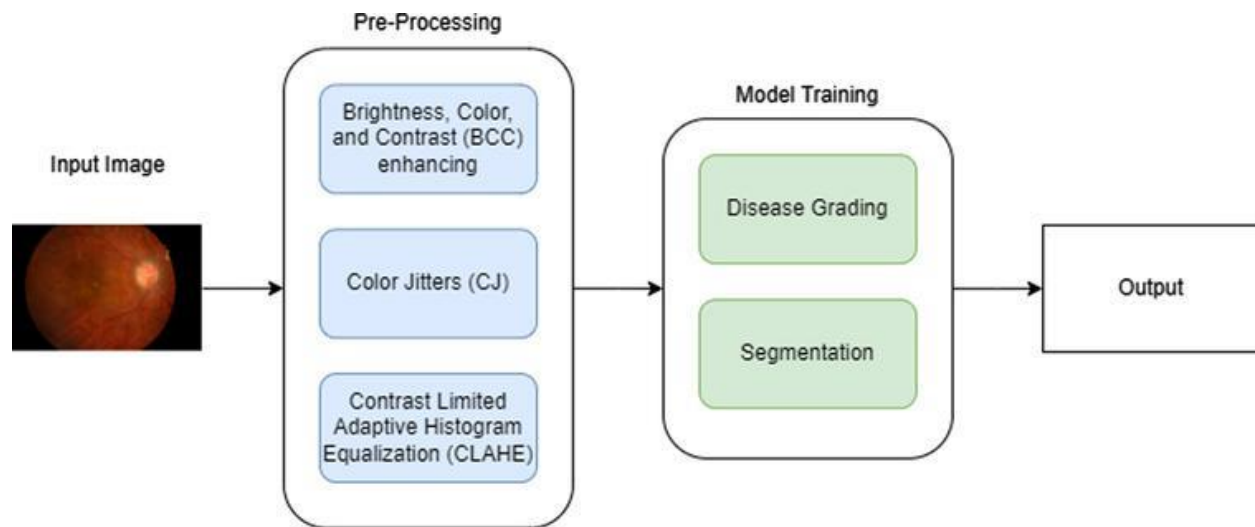
PROBLEM STATEMENT	I AM (CUSTOMER)	I AM TRYING TO	BUT	BECAUSE	WHICH MAKES ME FEEL
PS-1	PATIENT	Differentiate whether the changes in the vision of the eye is caused by diabetic retinopathy or by other health issues.	Early stages of diabetic retinopathy usually don't have any symptoms.	It is caused by damage to the blood vessels of the light sensitive tissue at the back of the eye.	Scared, Anxiety
PS-2	DOCTOR (OPHTHALMOLOGIST)	Treat the growth of new blood vessels at the back of eye.	Treating new blood vessels in eye can cause pain and a dangerous increase in pressure inside the eye.	Of NPDR the walls of the blood vessels in the retina weaken.	Guilty and degradation of career.
PS-3	GOVERNMENT	Protect and advance the delivery of high quality health care.	Most of the citizens are not aware about the schemes provided by the government.	The government's best efforts in creating awareness are through radio, TV, print or online media. Unfortunately, these communication efforts often do not reach rural areas.	Scanty

Proposed Solution:-

Sl.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To develop a AI based deep learning fundus image analysis and classification by following criteria, 1.To avoid the vision loss of the patient that caused by diabetic retinopathy. 2.To create awareness among people so they can get the clear clarification of this disease.
2.	Idea / Solution description	To develop deep learning approach such as Deep Convolutional Neural Network(DCCN) gives high accuracy in classification of these diseases through spatial analysis.A DCCN is more complex architecture inferred more from Human visual perspects.
3.	Novelty / Uniqueness	Deep convolutional neural network is to find a better and optimized way to classifying the fundus image with little preprocessing techniques.
4.	Social Impact / Customer Satisfaction	It will save the lives of people and minimize the vision loss by classifying the diabetic retinopathy using AI.
5.	Business Model (Revenue Model)	Due to the increasing high demand for diabetic retinopathy for damaged eye-retina patients on which the diabetic retinopathy manufactures are anticipated to generate high revenue.
6.	Scalability of the Solution	Diabetic Retinopathy is preventable through strict glycaemic control,annual dilated eye exams by an ophthalmologist and modification in life style.

Problem Solution Fit:-

Each domain is important in improving the effectiveness of a programme; focusing on one domain exclusively will not achieve maximum improvement for a given set of resources. Investing in equipment without considering the pathway of how people with diabetes might be identified or referred if necessary.



REQUIREMENT ANALYSIS:-

Functional requirement:-

Following are the functional requirements of the proposed solution.

FR.No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Identify and selecting dataset	It is necessary to select the appropriate dataset to enhance the model's performance.
FR-2	Diagnosis	The training should ensure proper diagnosis and make sure to identify the true and false of the medical condition [Diabetic retinopathy].
FR-3	Analysis	Based on the training the model should analyse the medical condition [DR] in order to protect/detect the disease accurately.
FR-4	Testing	The trained model is tested with different data to ensure it has trained well to predict/detect the medical condition[DR]

FR-5	Treatment	The testing of the model gives us the level of the medical condition so that we can go for the required treatment.
------	-----------	--

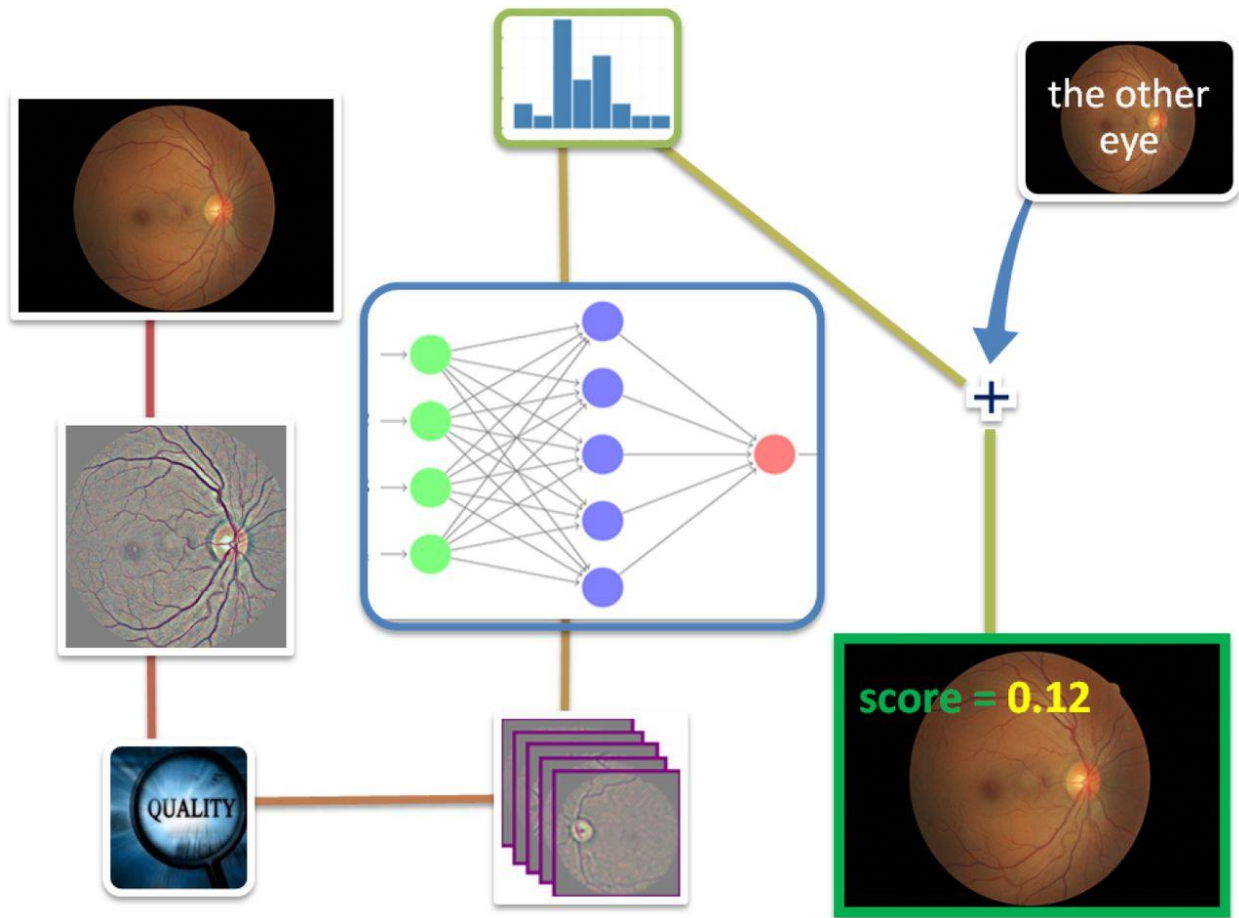
Non-Functional requirements:-

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

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	User with basic understanding of the medical condition and computer knowledge can operate the system.
NFR-2	Reliability	There is a chance of hardware failure or false positives when the testing data is more of different than the training dataset.
NFR-3	Performance	The performance of the model is meant to give speedy results for the patients.
NFR-4	Availability	The model is made to be available a anytime and anywhere.
NFR-5	Scalability	The scalability of the model can be enhanced with future technologies so that the performance of the model can be improved.

PROJECT DESIGN:-

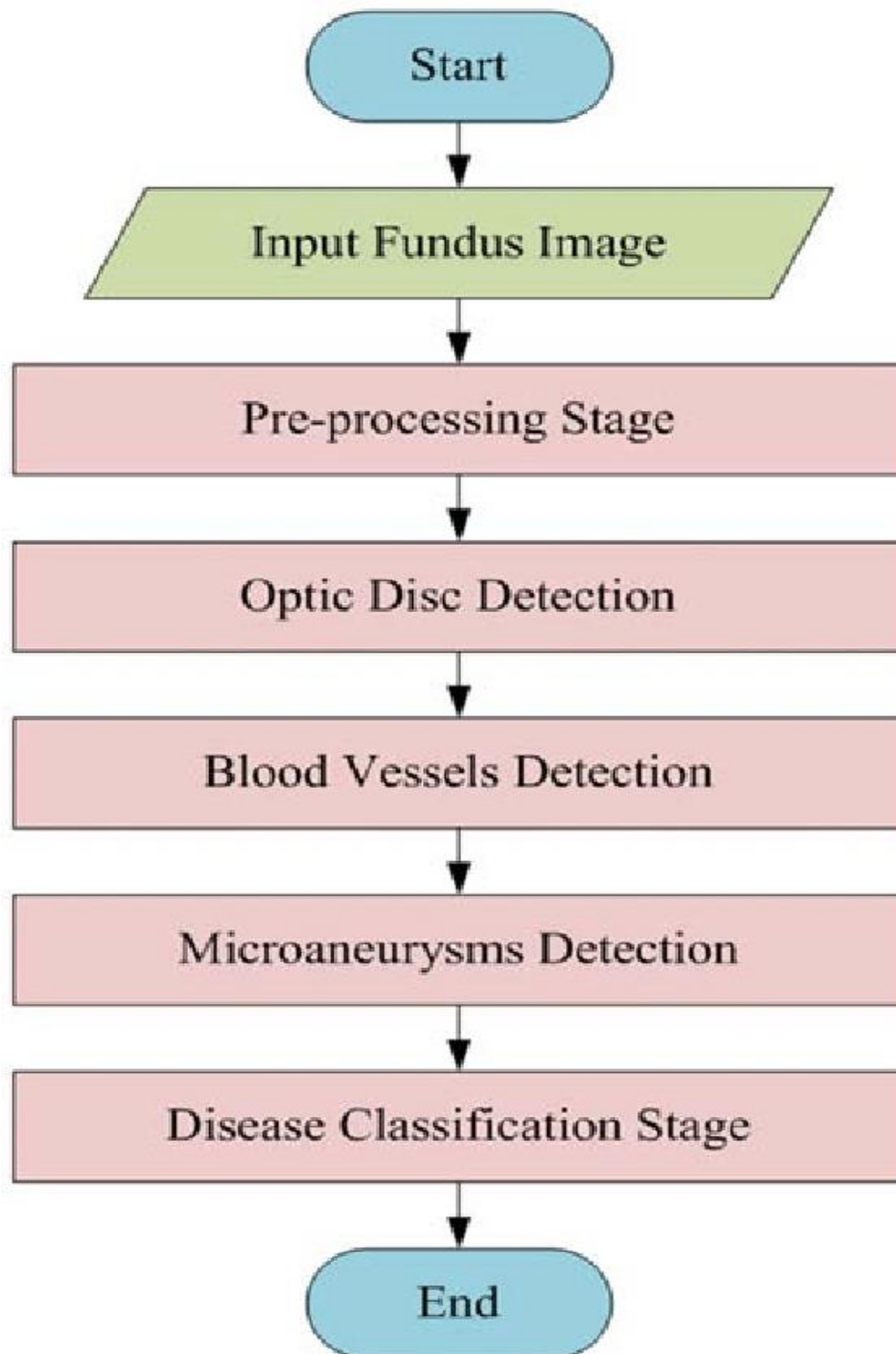
Data Flow Diagrams:-



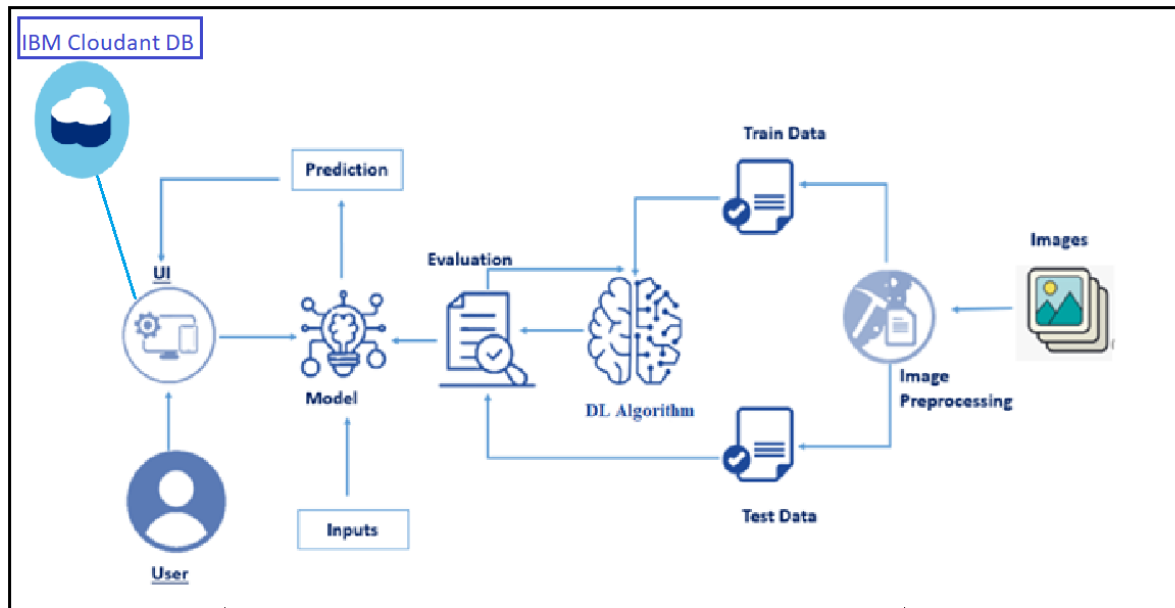
User type	Functional Requirement(Epic)	User story number	User story /task	Acceptance Criteria	Priority	Release
Common user	Dashboard	USN-1	As a user, I can able to	I can upload or take image	High	Sprint-1

			upload image of my eyes			
		USN-2	As a user, I will receive the diagnosis as to whether I have retinopathy or not	I can receive the diagnosis	High	Sprint-1
		USN-3	As a user , I receive the severity of the retinopathy	I can receive the severity of the retinopathy	Medium	Sprint-2
		USN-4	As a user , I can receive the suggested remedy	I can receive the suggested remedy	Medium	Sprint-2

Solution Architecture:-



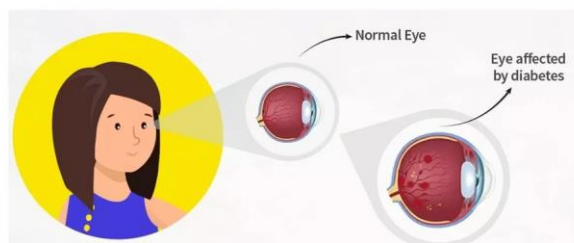
Technical Architecture:-



User Stories:-

- As a user, I can able to upload image of my eyes.
- As a user, I will receive the diagnosis as to whether I have retinopathy or not.
- As a user , I receive the severity of the retinopaahy.
- As a user , I can receive the suggestedremedy.

DIABETIC RETINOPATHY



PROJECT PLANNING & SCHEDULING:-

Sprint Planning & Estimation:-

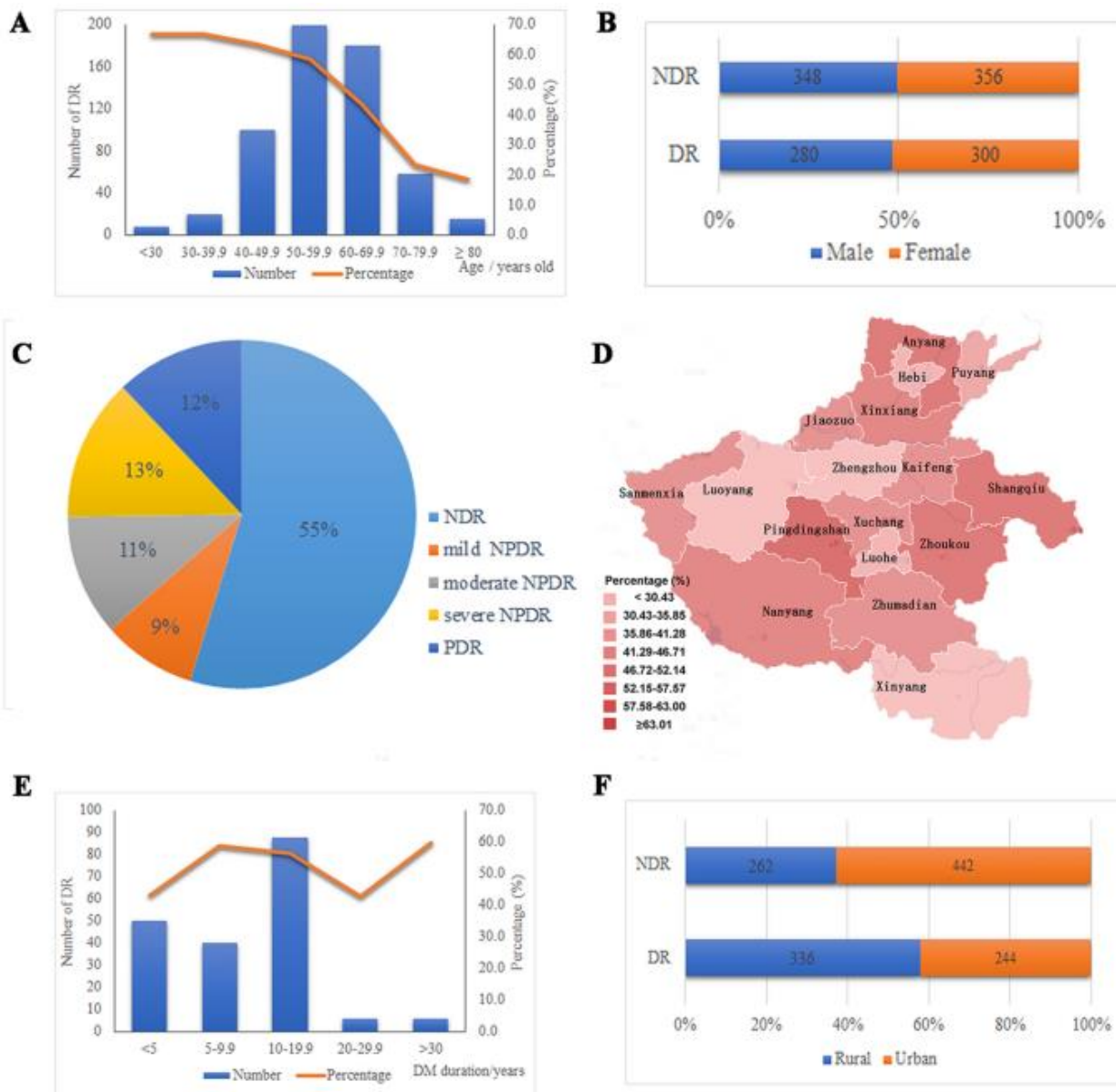
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Screening method	USN-1	As a user,I can find the method more efficient and accurate.	7	High
Sprint-2		USN-2	As a user,I can use it with minimal physical interaction with the device.	6	Medium
Sprint-4	Physical features	USN-3	As a user, I can find it portable and light weight.	10	Low
Sprint-3	Safety	USN-4	As a user, I can be safe as the detection methodis free from radiations.	8	High
Sprint-1	Testing	USN-5	As a user,I can undergo testing withoutany fear of pain as this method is pain-free .	7	Low
Sprint-3		USN-6	As a user ,I will be comfortable as it requiresminimum / no human involvement.	3	Medium
Sprint-1	Results	USN-7	As a user, I can rely on the resultswithout any suspicion.	6	High
Sprint-3		USN-8	As a user,Ican benefit from the resultsas it will help me know whethertreatment is necessaryor not.	8	Medium
Sprint-2		USN-9	As a user,Ican get the results on the spot	7	Low

			immediately after the screening process.		
--	--	--	--	--	--

Sprint Delivery Schedule:-

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on PlannedEnd Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	01 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Reports from JIRA:-



JIRA FILES

The screenshot shows the Jira Software interface for a project named "ibm-diabetic-retinopat...". The left sidebar contains navigation options: PLANNING (Roadmap, Backlog, Board), DEVELOPMENT (Code), Project pages, Add shortcut, and Project settings. The main area displays "Sprint-1" with a search bar and a filter icon. The sprint board is divided into three columns: TO DO, IN PROGRESS, and DONE. The DONE column contains three issues: "As a user, I can register for the application by entering my email, password, and confirming my password." (IDR-11), "As a user, I will receive confirmation email once I have registered for the application." (IDR-12), and "As a user, I will receive confirmation message." (IDR-13). The top navigation bar includes "Your work", "Projects", "Filters", "Dashboards", "People", "Apps", and a "Create" button. A search bar is located in the top right corner.

The screenshot shows the Jira Software interface for the same project, now displaying "Sprint-2". The left sidebar is identical to the previous screenshot. The main area shows "Sprint-2" with a search bar and a filter icon. The sprint board is divided into three columns: TO DO, IN PROGRESS, and DONE. The DONE column contains three issues: "Validates user login id" (IDR-15), "As a user, I can register for the application through mail" (IDR-14), and "Prediction part" (IDR-16). The top navigation bar includes "Your work", "Projects", "Filters", "Dashboards", "People", "Apps", and a "Create" button. A search bar is located in the top right corner. The file size "64.5 KB" is displayed in the top left corner of the interface.

Jira Software

Your work

Projects

Filters

Dashboards

People

Apps

Create

Q Search

ibm-diabetic-retinopat...
Software project

PLANNING

Roadmap

Backlog

Board

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

Projects / ibm-diabetic-retinopathy

Sprint-3

3 days remaining

Q Search

SK

+

Epic

Sprint

Clear filters

GROUP BY

None

TO DO

IN PROGRESS 1 OF 1 ISSUE

DONE 1 OF 7 ISSUE ✓

+

Doctor uses this report for seeing patient's activity

IDR-18

Connecting to database

IDR-17

Jira Software

Your work

Projects

Filters

Dashboards

People

Apps

Create

Q Search

ibm-diabetic-retinopat...
Software project

PLANNING

Roadmap

Backlog

Board

DEVELOPMENT

Code

Project pages

Add shortcut

Project settings

Projects / ibm-diabetic-retinopathy

Sprint-4

2 days remaining

Q Search

SK

+

Epic

Sprint

Clear filters

GROUP BY

None

TO DO

IN PROGRESS

DONE 2 OF 9 ISSUES ✓

+

User can see their dashboard

IDR-19

Logout from the current user

IDR-20

CODING & SOLUTIONING:-

Feature 1:-

Admin Page Code:-

```
from django.contrib import admin
# Register your models here.
```

Explanation:-

Admin is the role with the highest level of access to the website. Admins can add content on all pages and access all items in the Admin Toolbar. This means that Admins can control site-wide settings like the design of the website and the homepage layout. They can add and delete other admin users, and can approve or deny edits made by authors.

Home Page Code:-

```
<!DOCTYPE html>
{% load static %}
<meta name="viewport" content="width=device-width, initial-scale=1">
<html>
<head>
<style>
html {
    background: "{% static 'assets/images/.jpg' %}";
    background-size: cover;
    background-attachment: fixed;
    background-position: center bottom;
}
h2 {
    color: "red";
}
body {
    font-family: Arial;
    background-repeat: no-repeat;
    background-position: center bottom;
    background-size: 80% 100%;
    color: rgb(255, 255, 255);
    background-color: #002447 ;
}
button{
    height: 75px;
```

```

        width: 270px;
        background-color: transparent;
        font-size: large;
        color: rgb(255, 255, 255);
        border: 2px solid rgb(255, 255, 255);
    }
</script>
function openCity(evt, cityName) {
    var i, tabcontent, tablinks;
    tabcontent = document.getElementsByClassName("tabcontent");
    for (i = 0; i < tabcontent.length; i++) {
        tabcontent[i].style.display = "none";
    }
    tablinks = document.getElementsByClassName("tablinks");
    for (i = 0; i < tablinks.length; i++) {
</script>
</body>
</html>

```

Explanation:-

A home page is the default or front page of a site. It is the first page that visitors see when they load a URL. Web managers can control the home page as a way of directing the user experience. Home pages are located in the root directory of the website. The home page often serves to orient visitors by providing titles, headlines and images and visuals that show what the website is about, and in some cases, who owns it and maintains it.

Feature 2:-

```

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

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

    <!-- CSS only -->

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"
rel="stylesheet"

        integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

    <title>Document</title>

    <body>

        <section class="vh-100">

```

```

        <div class="container-fluid h-custom">

        <div class="row d-flex justify-content-center align-items-center h-100">

        <div class="col-md-9 col-lg-6 col-xl-5">

    </div>

        <div class="col-md-8 col-lg-6 col-xl-4 offset-xl-1">

        <form>

    </div>

    </div>

    </div>

    </div>

    </section>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"
    crossorigin="anonymous"></script>

</body>

</html>

```

Explanation:-

The login page allows a user to gain access to an application by entering their username and password .A user navigates to an application and is presented with a login page as a way to gain access to the application. There are two possible results:

Authentication is successful and the user is directed to the application landing page.

Authentication fails and the user remains on the login page. If authentication fails, the screen should show an informational or error message about the failure.

Feature 2:-

Upload page:-

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Diabetic Retinopathy</title>
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.8.2/css/all.css">

```

```

<style>
    body {
        background:
            url("http://html.xpeedstudio.com/agmycoo/assets/images/welcome/welcome-icon-bg-3.png");
    }
</head>
<body>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.4/umd/popper.min.js"></script>
    <script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/mdbootstrap/4.19.0/js/mdb.min.js"></script>
<script>
    $(document).ready(() => {
        $("input[id='image']").on('change', function (event) {
            let input = this;
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#banner').css('width', '350px')
                $('#banner').addClass('img-thumbnail')
                $('#banner').attr('src', e.target.result);
            }
            reader.readAsDataURL(input.files[0]);
        })
        $.ajax({
            url: "http://127.0.0.1:8000/api/",
            type: "POST",
            dataType: 'json',
            success: function (xhr) {
                alert("Error while processing")
            },
            error: function (xhr) {
                $('#title').html("Result")
                let result = (xhr.responseText).split("-");
                let disease = result[0];
                let accuracy = result[1];
                $('.loader').hide()
                $('#disease').html("Result: " + disease)
                $('#accuracy').html("Accuracy: " + parseInt(accuracy).toFixed(2) + "%")
                $('#graph').attr('src', '{% static "graph.png" %}')
                $('#result').show()
            }
        })
    })
</script>
</body>
</html>

```


Database Schema:-

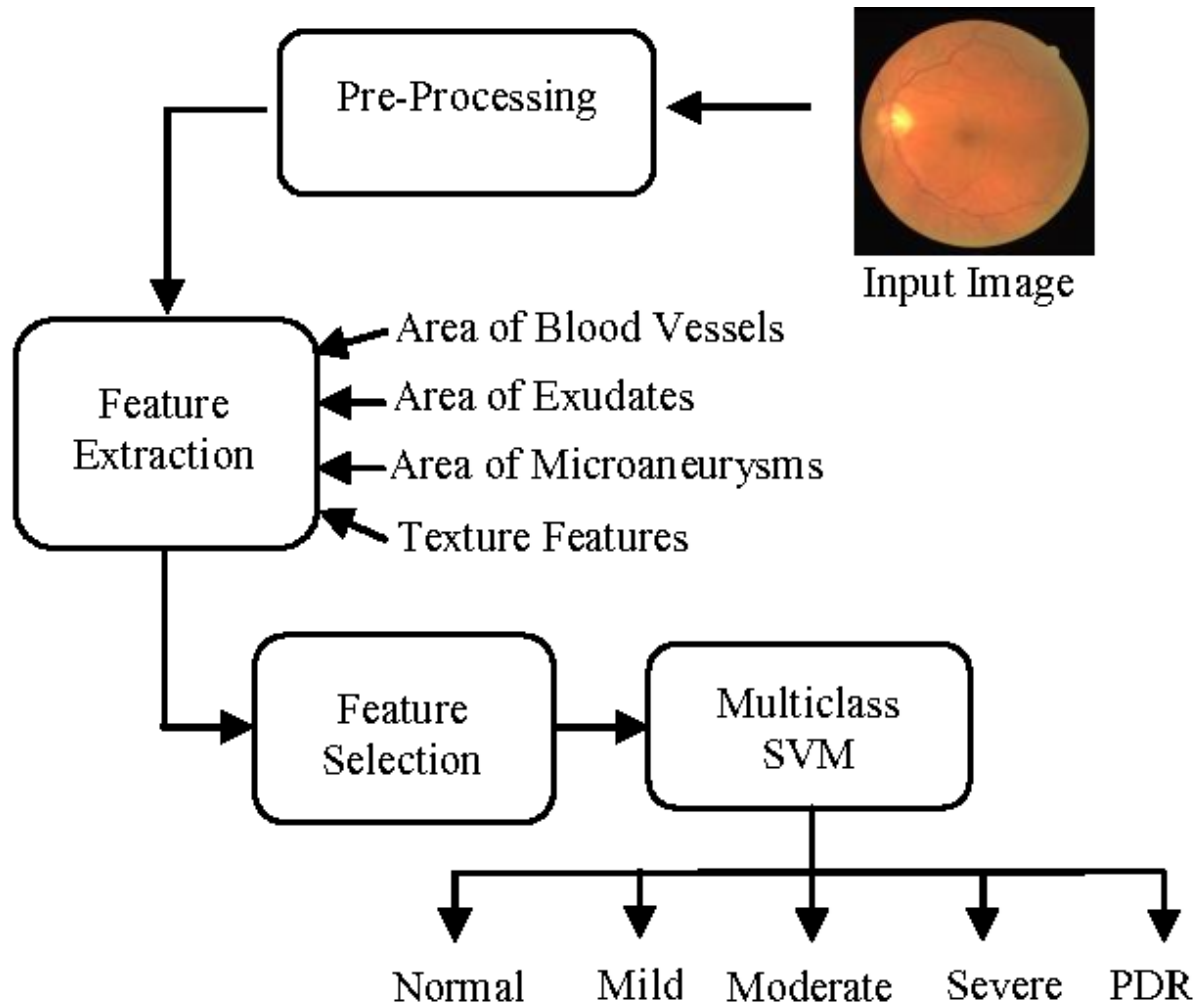


Fig. 2 Block diagram of the automated diabetic retinopathy classification of PDR

TESTING:-

Test Cases:-

```
from django.test import TestCase
# Create your tests here.
```

User Acceptance Testing:-

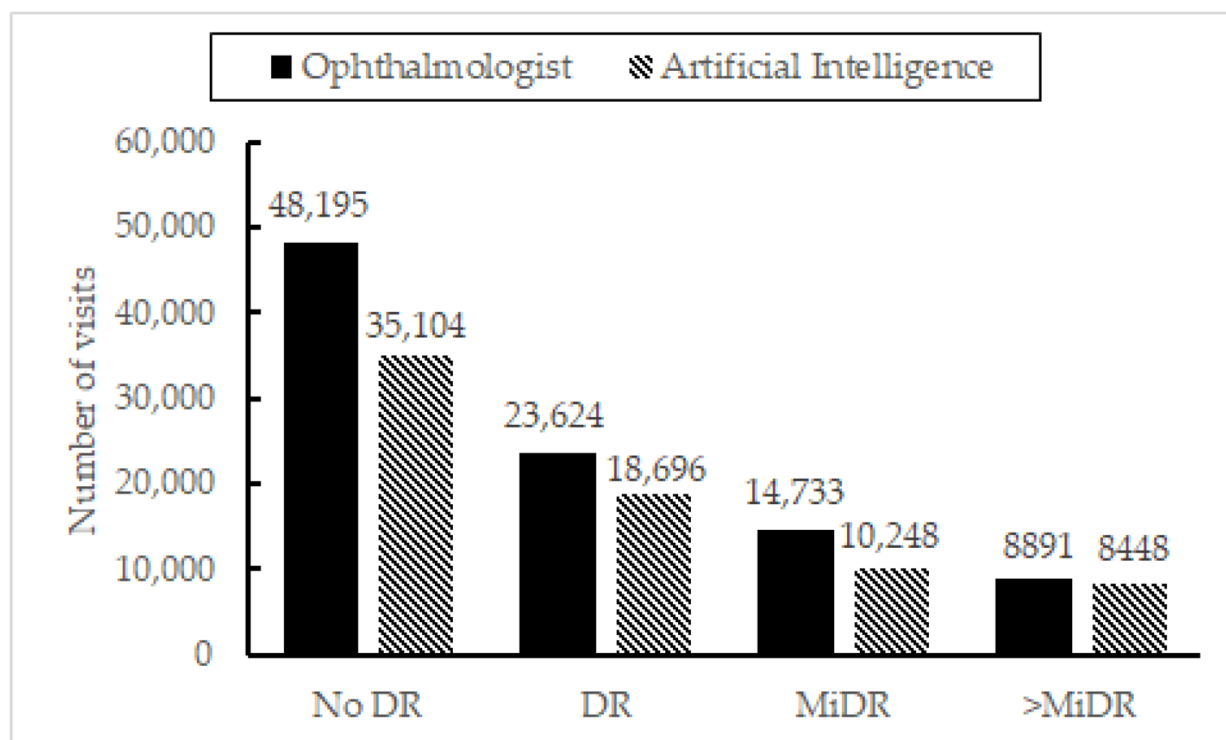
```
import tensorflow.keras
from PIL import Image, ImageOps
import numpy as np
import os

def process_img(img):
    np.set_printoptions(suppress=True)
    model = tensorflow.keras.models.load_model(os.path.dirname(__file__) +
'/keras_model.h5')

    data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
    image = Image.open(os.path.dirname(__file__) + '/test/' + img)
    image_array = np.asarray(image)
    data[0] = normalized_image_array
    prediction = model.predict(data)
    pred_new = prediction[0]
    pred = max(pred_new)
    print(pred_new)
    index = pred_new.tolist().index(pred)
    plt.xlabel('x - axis')
    plt.ylabel('y - axis')
    plt.savefig(os.path.dirname(__file__) + '/output/graph.png')
    plt.show()
    result = []
    if index == 0:
        result.append("No DR")
    elif index == 1:
        result.append("Mild")
    elif index == 2:
        result.append("Moderate")
    elif index == 3:
        result.append("Severe")
    elif index == 4:
        result.append("Proliferative DR")
    accuracy = round(pred, 2)
    result.append("-")
    result.append(accuracy * 100)
    return result
```

RESULTS:-

Performance Metrics:-



Acceptance Testing UAT Execution & Report Submission

1. Purpose of Document:-

This document serves as a quick reference for the Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy project's test coverage and open issues as of the project's release for user acceptance testing.

2. Defect Analysis:-

This shows how many bugs were fixed or closed at each severity level and how they were fixed.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	0	0	0	1
Duplicate	4	1	3	0	8
External	1	3	0	0	4
Fixed	2	4	4	2	12
Not Reproduced	0	0	0	1	1
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	8	8	4	2	22

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	5	0	0	5
Client Application	10	0	0	10
Security	2	0	0	2
Out-source Shipping	0	0	0	0
Exception Reporting	2	0	0	2
Final Report Output	4	0	0	4
Version Control	2	0	0	2

ADVANTAGES:-

- Relatively inexpensive.
- Does not require any special facilities to use.
- Can be linked to computers and images can be stored for the long term.

DISADVANTAGES:-

- Requires pupil dilation.
- In people with opacities in the lens, such as cataracts, it might not be possible to take an image; this is the main source of failure in diabetic retinopathy screening, and people will need to be rescreened using other methods.

CONCLUSION:-

Diabetic retinopathy is a serious complication of diabetes mellitus, leading to progressive damage and even blindness of the retina. Its early detection and treatment is important in order to prevent retina's damage. The interest in applying deep learning in detecting diabetic retinopathy has increased during the past years and as several DL systems evolve and become integrated into the clinical practice, they will enable the clinicians to treat the patients in need more effectively and efficiently. This report presents the application of deep learning in diagnosing diabetic retinopathy. Although deep learning has paved the way for more accurate diagnosis and treatment, further improvements are still necessary regarding performance, interpretability and trustworthiness from ophthalmologists.

FUTURE SCOPE:-

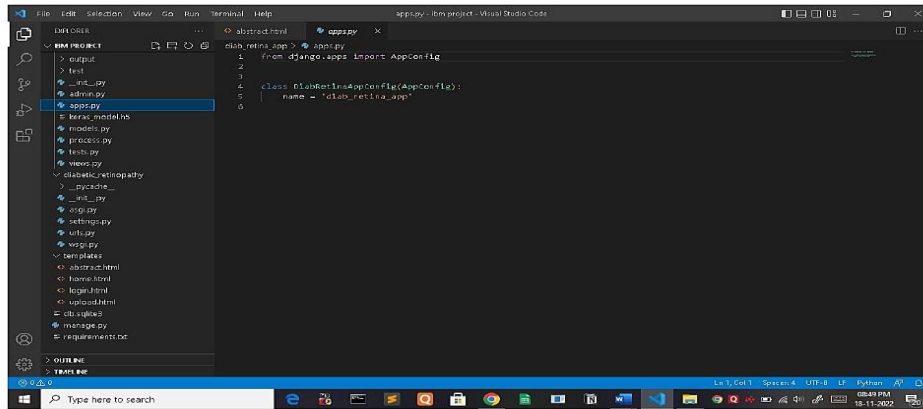
As ophthalmic imaging technologies continue to expand their capabilities, eye care practitioners have an increasing abundance of data points necessary for managing diabetic eye disease over time. To improve prediction the future management and treatment of DR will have to rely increasingly on longitudinal clinical data via

various imaging technologies and also artificial intelligence (AI) platforms that allow integrating this increasing amount of data. The implementation of AI and machine learning in the eye care world will provide auxiliary data points that factor into the classification/risk assessment of a patient's current level of DR.

APPENDIX:-

Source Code:-

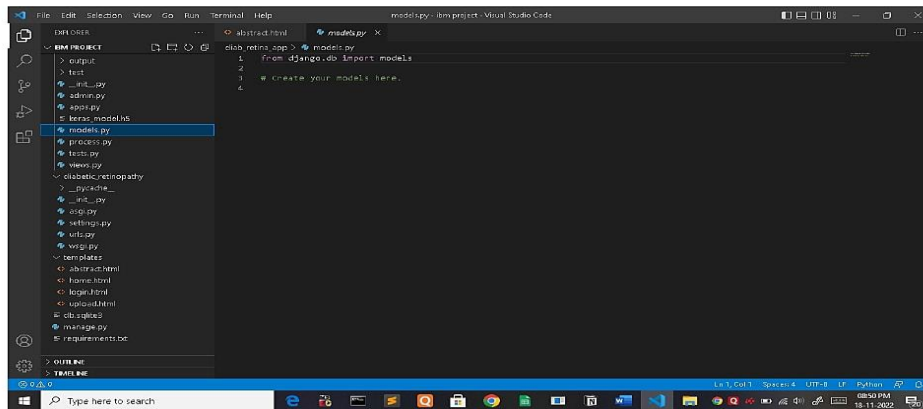
APPLICATION (PYTHON CODE)



The screenshot shows the Visual Studio Code interface with a Django project named 'diabetic-retina-app'. The file explorer on the left shows the project structure, including 'apps.py' under the 'diabetic_retinopathy' app. The main editor displays the content of 'apps.py'.

```
1 from django.apps import AppConfig
2
3
4 class DiabeticRetinaAppConfig(AppConfig):
5     name = 'diabetic_retina_app'
6
```

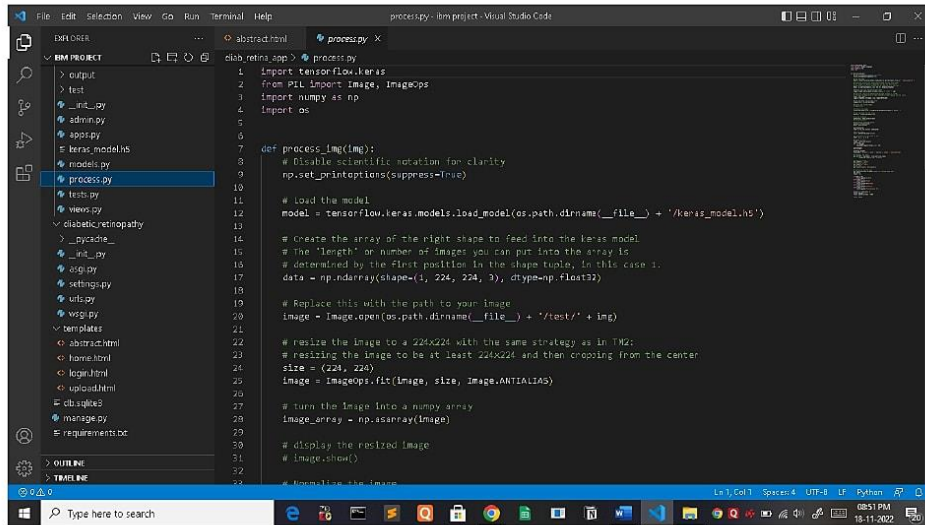
MODEL(PYTHON CODE)



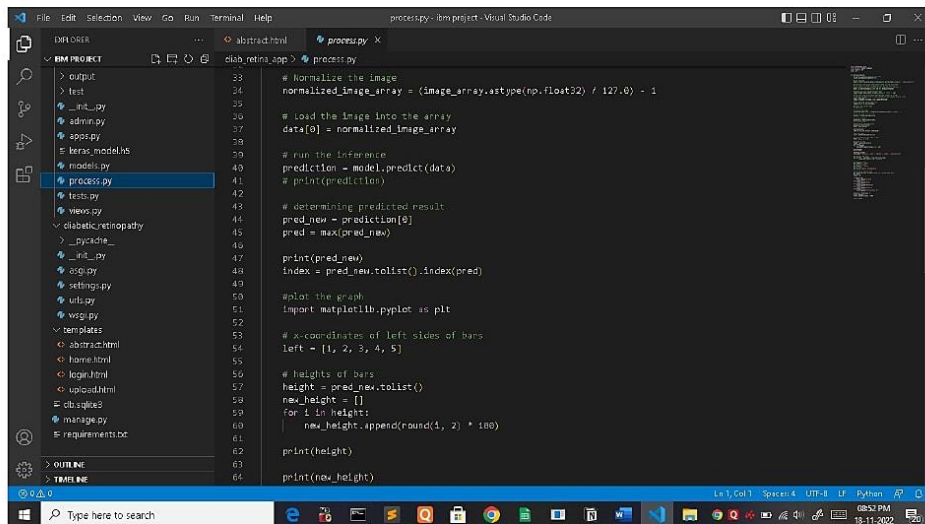
The screenshot shows the Visual Studio Code interface with the same Django project. The file explorer on the left shows the project structure, including 'models.py' under the 'diabetic_retinopathy' app. The main editor displays the content of 'models.py'.

```
1 from django.db import models
2
3 # Create your models here.
4
```


PROCESS(PYTHON CODE)



```
1 import tensorflow.keras
2 from PIL import Image, ImageOps
3 import numpy as np
4 import os
5
6
7 def process_img(img):
8     # Disable scientific notation for clarity
9     np.set_printoptions(suppress=True)
10
11     # Load the model
12     model = tensorflow.keras.models.load_model(os.path.dirname(__file__) + '/keras_model.h5')
13
14     # Create the array of the right shape to feed into the keras model
15     # The 'length' or number of images you can put into the array is
16     # determined by the first position in the shape tuple, in this case 1:
17     data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
18
19     # Replace this with the path to your image
20     image = Image.open(os.path.dirname(__file__) + '/test/' + img)
21
22     # resize the image to a 224x224 with the same strategy as in TM2:
23     # resizing the image to be at least 224x224 and then cropping from the center
24     size = (224, 224)
25     image = imageops.fit(image, size, image.ANTIALIAS)
26
27     # turn the image into a numpy array
28     image_array = np.asarray(image)
29
30     # display the resized image
31     # image.show()
32
33     # Normalize the image
```



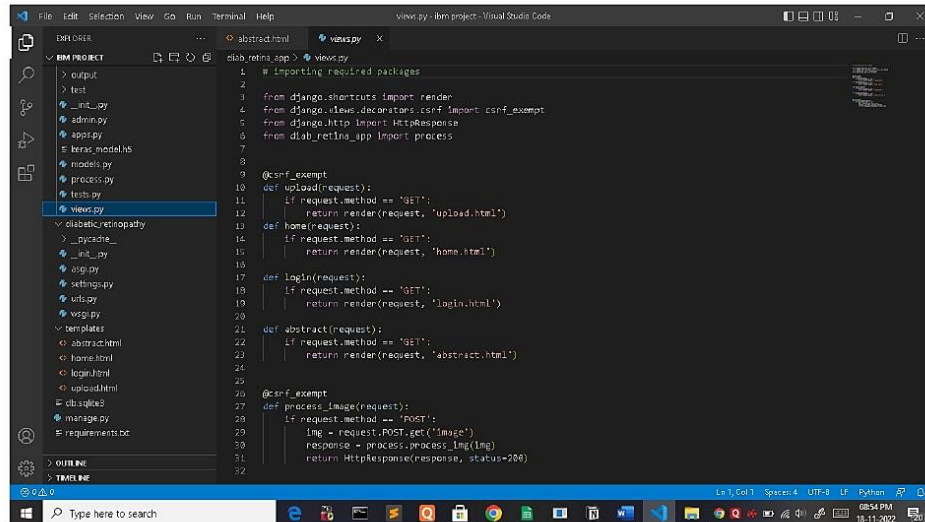
```
33     # Normalize the image
34     normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1
35
36     # Load the image into the array
37     data[0] = normalized_image_array
38
39     # Run the inference
40     prediction = model.predict(data)
41     # print(prediction)
42
43     # determining predicted result
44     pred_new = prediction[0]
45     pred = max(pred_new)
46
47     print(pred_new)
48     index = pred_new.tolist().index(pred)
49
50     #plot the graph
51     import matplotlib.pyplot as plt
52
53     # x-coordinates of left sides of bars
54     left = [1, 2, 3, 4, 5]
55
56     # heights of bars
57     height = pred_new.tolist()
58     new_height = []
59     for i in height:
60         new_height.append(round(i, 2) * 100)
61
62     print(height)
63
64     print(new_height)
```

```
64 print(new_height)
65 tick_label = ['no_dtr', 'mild', 'moderate', 'severe', 'proliferative']
66
67 # plotting a bar chart
68 plt.bar(left, new_height, tick_label=tick_label,
69         width=0.5, color=['red', 'green'])
70
71 # naming the x-axis
72 plt.xlabel('x - axis')
73 # naming the y-axis
74 plt.ylabel('y - axis')
75 # plot title
76 plt.title("Diabetic Retinopathy")
77
78 # function to show the plot
79 plt.savefig(os.path.dirname(__file__) + '/output/graph.png')
80 plt.show()
81 result = []
82
83 if index == 0:
84     result.append("No DR")
85 elif index == 1:
86     result.append("Mild")
87 elif index == 2:
88     result.append("Moderate")
89 elif index == 3:
90     result.append("Severe")
91 elif index == 4:
92     result.append("Proliferative DR")
93
94 accuracy = round(pred, 2)
95 result.append("%")
```

TEST(PYTHON CODE)

```
1 from django.test import TestCase
2
3 # Create your tests here.
4
```

VIEWS(PYTHON CODE)



```
1 # Importing required packages
2
3 from django.shortcuts import render
4 from django.views.decorators.csrf import csrf_exempt
5 from django.http import HttpResponseRedirect
6 from diab_retina_app import process
7
8
9 @csrf_exempt
10 def upload(request):
11     if request.method == 'GET':
12         return render(request, 'upload.html')
13
14 def home(request):
15     if request.method == 'GET':
16         return render(request, 'home.html')
17
18 def login(request):
19     if request.method == 'GET':
20         return render(request, 'login.html')
21
22 def abstract(request):
23     if request.method == 'GET':
24         return render(request, 'abstract.html')
25
26
27 @csrf_exempt
28 def process_image(request):
29     if request.method == 'POST':
30         img = request.POST.get('image')
31         response = process.process_img(img)
32         return HttpResponseRedirect(response, status=200)
```

ABSTRACT PAGE(HTML CODE)

The screenshot shows a Windows 10 desktop with a VS Code editor open. The editor is displaying a file named 'abstract.html' in the 'templates' folder. The file contains the following HTML code:

```

1 <!DOCTYPE html>
2 <html>
3 <body style="background-image: url('https://cd.wallpaperflare.com/wallpaper/370/366/3/abstract-eyes-dors-dig');>
4 <div>
5 <h1 style="color: #ff0000; text-align: center;">ABSTRACT</h1>
6 <h3 style="color: #ff0000; font-family: Arial;">
7 Diabetic Retinopathy (DR) is a complication caused by diabetes that affects the human eye. It is caused by t
8 (DR)
9 (DR)
10
11 </div>
12
13
14 </body>
15 </html>
16
17

```

The VS Code interface includes a sidebar on the left with a file explorer showing a project structure. The 'templates' folder is expanded, showing files like 'abstract.html', 'home.html', 'login.html', 'upload.html', 'db.sql', 'manage.py', and 'requirements.txt'. The main editor area shows the HTML code for 'abstract.html'. The status bar at the bottom indicates the file is at line 1, column 1, in UTF-8 encoding, with a cursor at the end of the line.

HOME PAGE(HTML CODE)

The screenshot shows a Windows 10 desktop with a VS Code editor window open. The editor is displaying a file named `home.html` within a project named `ibm-project - Visual Studio Code`. The file contains HTML code for a web page with a red background and a logo. The code is as follows:

```

1 <!DOCTYPE html>
2 (<% load static %>)
3 <meta name="viewport" content="width=device-width, initial-scale=1">
4 <html>
5
6 <head>
7
8 <style>
9
10     background: "({ static 'assets/images/.jpg' %})";
11     background-size: cover;
12     background-attachment: fixed;
13     background-position: center bottom;
14 }
15
16 h2 {
17     color: "red";
18 }
19
20 body {
21     font-family: Arial;
22     background-repeat: no-repeat;
23     background-attachment: fixed;
24     background-position: center bottom;
25     background-size: 80% 180%;
26     color: #000255;
27     background-color: #000244;
28     background-image: url('https://preview123r.com/images/andreu/andreu1400/andreu140005276731976451');
29     /* background-repeat: no-repeat; */
30     /* background-size: 180% 820px; */
31 }
32

```

The VS Code interface shows the Explorer sidebar on the left with a file tree containing `__init__.py`, `admin.py`, `apps.py`, `keras_models.py`, `models.py`, `process.py`, `tests.py`, `views.py`, `diabetic_retinopathy.py`, `pycadie.py`, `__init__.py`, `api.py`, `settings.py`, `urls.py`, `wsgi.py`, `templates`, `home.html` (selected), `login.html`, `upload.html`, `diagnostic.py`, `manage.py`, and `requirements.txt`. The Run and Debug sidebar is also visible on the left. The bottom status bar shows the file is at line 1, column 1, with UTF-8 encoding and a 100% zoom level. The Windows taskbar at the bottom shows the Start button, a search bar, and several pinned applications including Edge, File Explorer, and VS Code. The system clock in the bottom right corner shows the date as 8/11/2022 and the time as 10:10 AM.

LOGIN PAGE(HTML CODE)

The screenshot shows the Visual Studio Code interface with a web application project. The Explorer sidebar on the left displays a file tree with folders like 'output', 'test', and 'src'. The 'home.html' file is selected. The main editor area displays the content of 'home.html', which includes a header with a title 'Home', a main content area with a list of links, and a footer. The code is written in HTML and JavaScript. The status bar at the bottom shows the current file is 'home.html' and the cursor is at line 1, column 1.

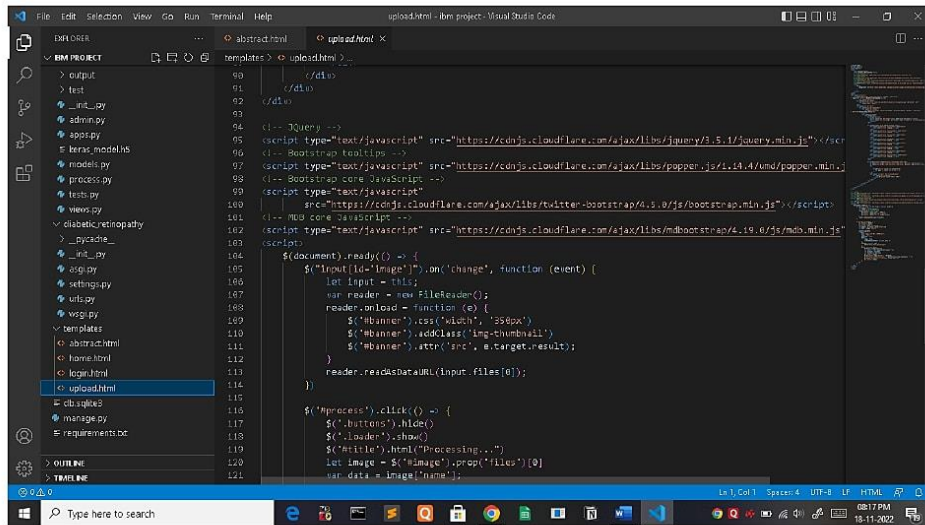
The screenshot shows a Windows 10 desktop with a VS Code editor open. The editor is displaying a Bootstrap login form template. The sidebar on the left shows a file explorer with a project structure. The main editor area shows the HTML code for 'login.html'. The code includes a container with a flex justify-content center, a form with email and password inputs, and a login button. The code is commented with Bootstrap classes and IDs. The status bar at the bottom shows 'Ln1, Col1', 'Spaces: 4', 'UTF-8', 'CSS', 'HTML', and a clock showing 18:11:02.

```

File Edit Selection View Go Run Terminal Help
login.html - Ibm project - Visual Studio Code

explorer
...
abstract.html X
login.html X

> output
33
34 (body)
35 <section class="vh-100">
36   <div class="container-fluid h-custom">
37     <div class="row d-flex justify-content-center align-items-center h-100">
38       <div class="col-md-9 col-lg-6 col-xl-5">
39         
40       </div>
41       <div class="col-md-6 col-lg-6 col-xl-4 offset-xl-1">
42         <form>
43           <!-- Email input -->
44           <div class="form-outline mb-4">
45             <input type="text" id="formExample3" class="form-control form-control-lg"
46               placeholder="Enter username" required/
47             <label class="form-label" for="formExample3">Email Username/Label</label>
48           </div>
49           <!-- Password input -->
50           <div class="form-outline mb-3">
51             <input type="password" id="formExample4" class="form-control form-control-lg"
52               placeholder="Enter password" required/
53             <label class="form-label" for="formExample4">Password/Label</label>
54           </div>
55           <div class="text-center text-lg-start mt-4 pt-2">
56             <a href="#">X</a> <a href="#">Upload 2</a>
57             <button type="button" class="btn btn-primary btn-lg"
58               style="padding-left: 2.5em; padding-right: 2.5em;">Login/Butten</button>
59           </div>
60         </form>
61       </div>
62     </div>
63   </div>
64 </div>
65
  
```

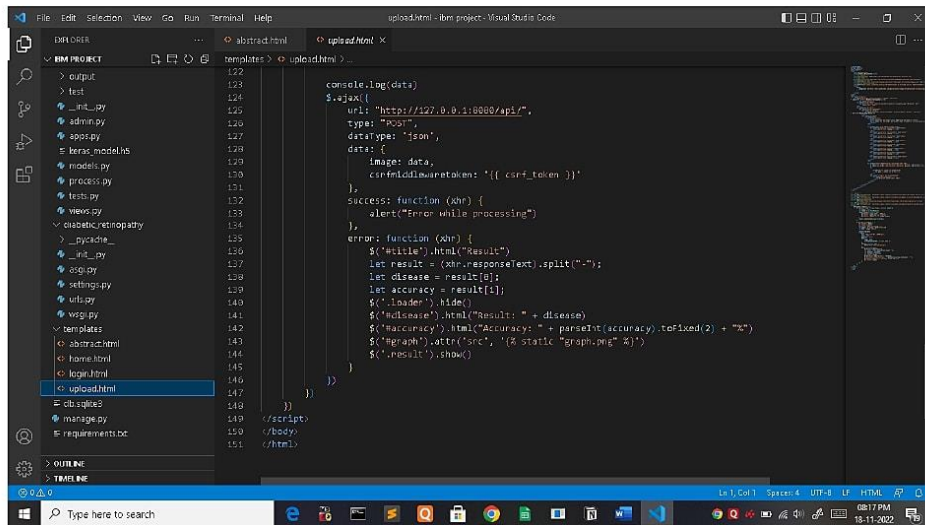
```
File Edit Selection View Go Run Terminal Help
upload.html - IBM project: Visual Studio Code

EXPLORER
  IBM PROJECT
    > output
    > test
    > _init_.py
    > admin.py
    > app.py
    > keras_model.py
    > models.py
    > process.py
    > test.py
    > view.py
    > diabetic_retinopathy
    > _pycache
    > _init_.py
    > asgi.py
    > settings.py
    > urls.py
    > wsgi.py
    > templates
      > abstract.html
      > home.html
      > login.html
      < upload.html
    > db.sqlite3
    > manage.py
    > requirements.txt

  > OUTLINE
  > TIMELINE

Type here to search
```

```
90 //div
91 //div
92 //div
93
94 <!-- jQuery -->
95 <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
96 <!-- Bootstrap tooltips -->
97 <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.4/umd/popper.min.js"></script>
98 <!-- Bootstrap core JavaScript -->
99 <script type="text/javascript"
100       src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.1.0/js/bootstrap.min.js"></script>
101 <!-- MDBootstrap core JavaScript -->
102 <script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/mdbootstrap/4.19.0/js/mdbootstrap.min.js"></script>
103
104 $(document).ready(() => {
105     $('#input[id="image"]').on('change', function (event) {
106         let input = this;
107         let reader = new FileReader();
108         reader.onload = function (e) {
109             $('#banner').css('width', '50px');
110             $('#banner').addClass('img-thumbnail');
111             $('#banner').attr('src', e.target.result);
112         }
113         reader.readAsDataURL(input.files[0]);
114     })
115
116     $('#process').click(() => {
117         $('#button').hide();
118         $('#loader').show();
119         $('#result').html('Processing...');
120         let image = $('#image').prop('files')[0];
121         let data = {
122             'image': image,
123             'csrf_token': '{{ csrf_token }}'
124         };
125         $.ajax({
126             url: 'http://127.0.0.1:8080/api/',
127             type: 'POST',
128             dataType: 'json',
129             data: data,
130             success: function (xhr) {
131                 alert("Error while processing")
132             },
133             error: function (xhr) {
134                 $('#result').html("Result")
135                 let result = (xhr.responseText).split("-");
136                 let disease = result[0];
137                 let accuracy = result[1];
138                 $('#loader').hide();
139                 $('#result').html("Result: " + disease);
140                 $('#accuracy').html("Accuracy: " + parseFloat(accuracy).toFixed(2) + "%");
141                 $('#graph').attr('src', '{% static "graph.png" %}');
142                 $('#result').show();
143             }
144         })
145     })
146 })
147
148 </script>
149 </body>
150 </html>
```



```
File Edit Selection View Go Run Terminal Help
upload.html - IBM project: Visual Studio Code

EXPLORER
  IBM PROJECT
    > output
    > test
    > _init_.py
    > admin.py
    > app.py
    > keras_model.py
    > models.py
    > process.py
    > test.py
    > view.py
    > diabetic_retinopathy
    > _pycache
    > _init_.py
    > asgi.py
    > settings.py
    > urls.py
    > wsgi.py
    > templates
      > abstract.html
      > home.html
      > login.html
      < upload.html
    > db.sqlite3
    > manage.py
    > requirements.txt

  > OUTLINE
  > TIMELINE

Type here to search
```

```
122 console.log(data)
123 $.ajax({
124     url: 'http://127.0.0.1:8080/api/',
125     type: 'POST',
126     dataType: 'json',
127     data: {
128         'image': data,
129         'csrf_token': '{{ csrf_token }}'
130     },
131     success: function (xhr) {
132         alert("Error while processing")
133     },
134     error: function (xhr) {
135         $('#result').html("Result")
136         let result = (xhr.responseText).split("-");
137         let disease = result[0];
138         let accuracy = result[1];
139         $('#loader').hide();
140         $('#result').html("Result: " + disease);
141         $('#accuracy').html("Accuracy: " + parseFloat(accuracy).toFixed(2) + "%");
142         $('#graph').attr('src', '{% static "graph.png" %}');
143         $('#result').show();
144     }
145 })
146
147 </script>
148 </body>
149 </html>
```

ADMIN PAGE

