

24-25J-308

Deep learning retinal image analysis for human disease prediction



SUPERVISOR TEAM



Ms. Uthpala Samarakoon

Supervisor

Senior Lecturer
Faculty of Computing |
Information Technology



Ms. Sasini Hathurusinghe

Co-supervisor

Assistant Lecturer
Faculty of Computing |
Information Technology



Dr. MNM Salman

External Supervisor

MEMBERS



Nusaif SM
Leader
IT21172328



Thuvarahan T
IT21316654



Rimnas R
IT21175770



Sowkey A A
IT21386954

RETINAL TEARS:
Can They Be Corrected?

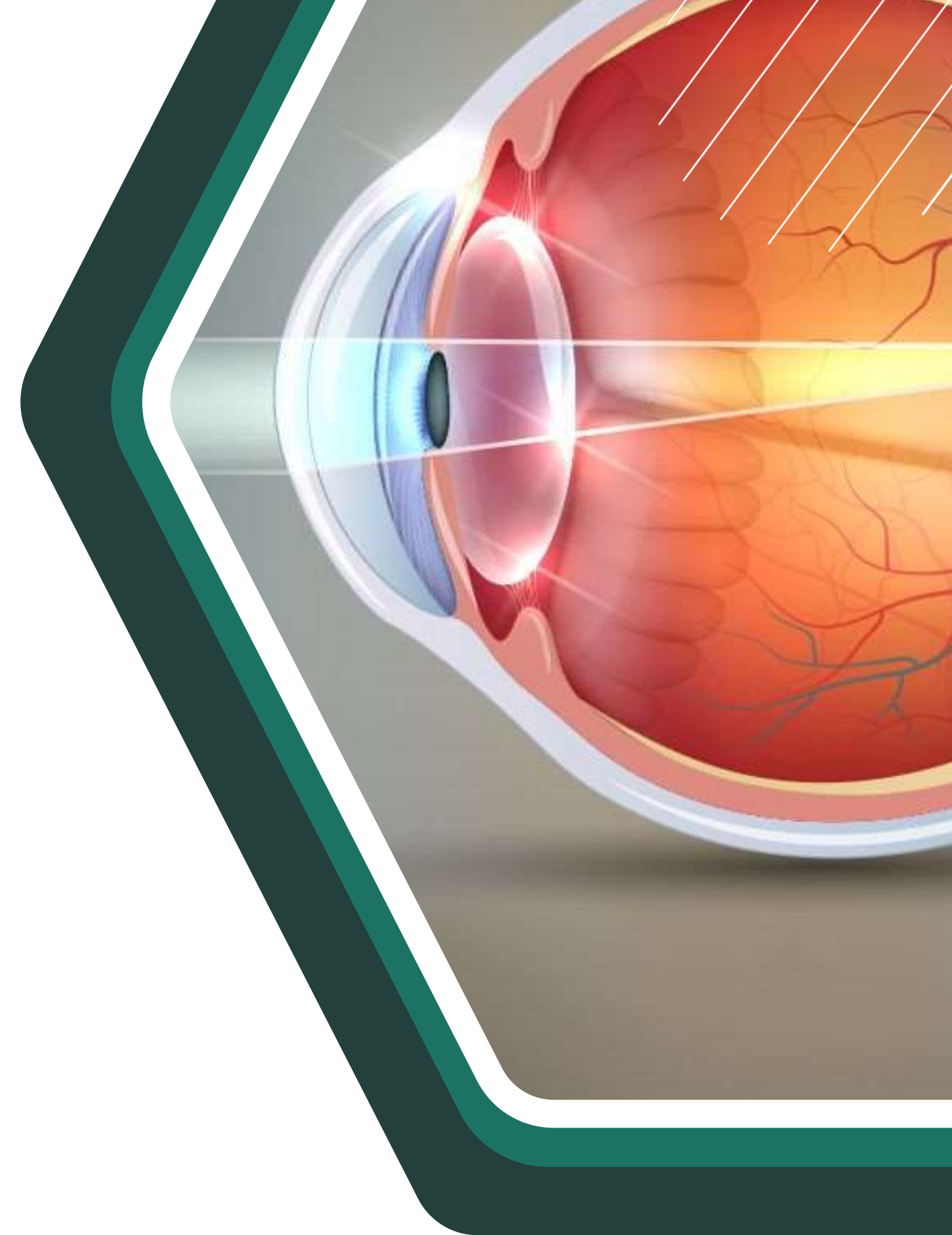
CONTENT

▶	Introduction	01
▶	Research Questions	02
▶	Objectives	03
▶	System Overview Diagram	04

▶	Technologies	05
▶	Requirements	06
▶	Gantt Chart	07
▶	Budget	08

INTRODUCTION

The eye is vital for vision, enabling us to perceive the world, learn, and navigate daily life. It enhances communication, safety, and overall well-being. However, retinal diseases can impair vision, making early detection crucial for preserving eye health.



RESEARCH QUESTIONS

Problem

How to develop a web application as the solution to?

Mission Statement

- Identifying the diseases
- Suggesting Treatment plan



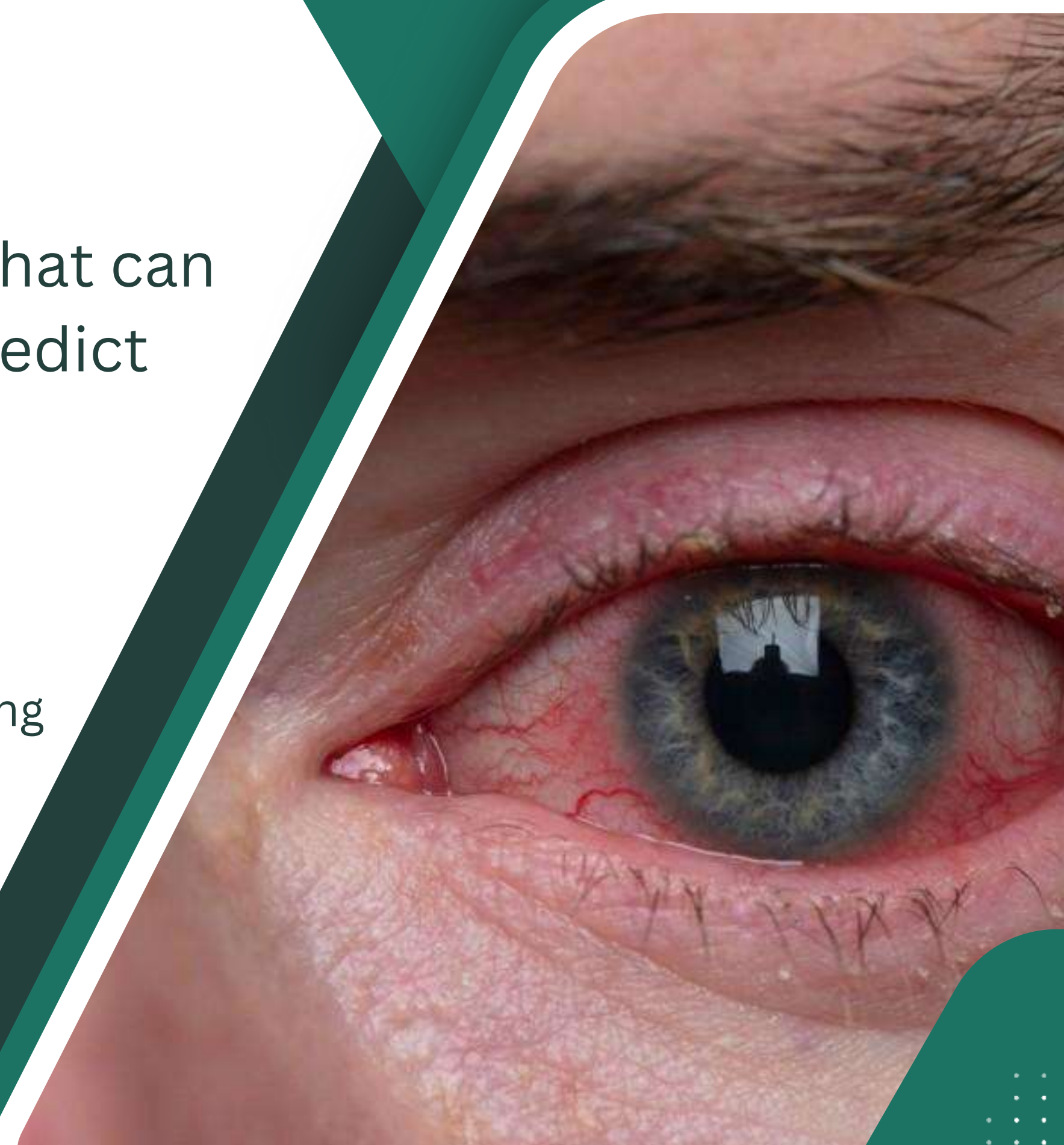
OBJECTIVES

Main Objective

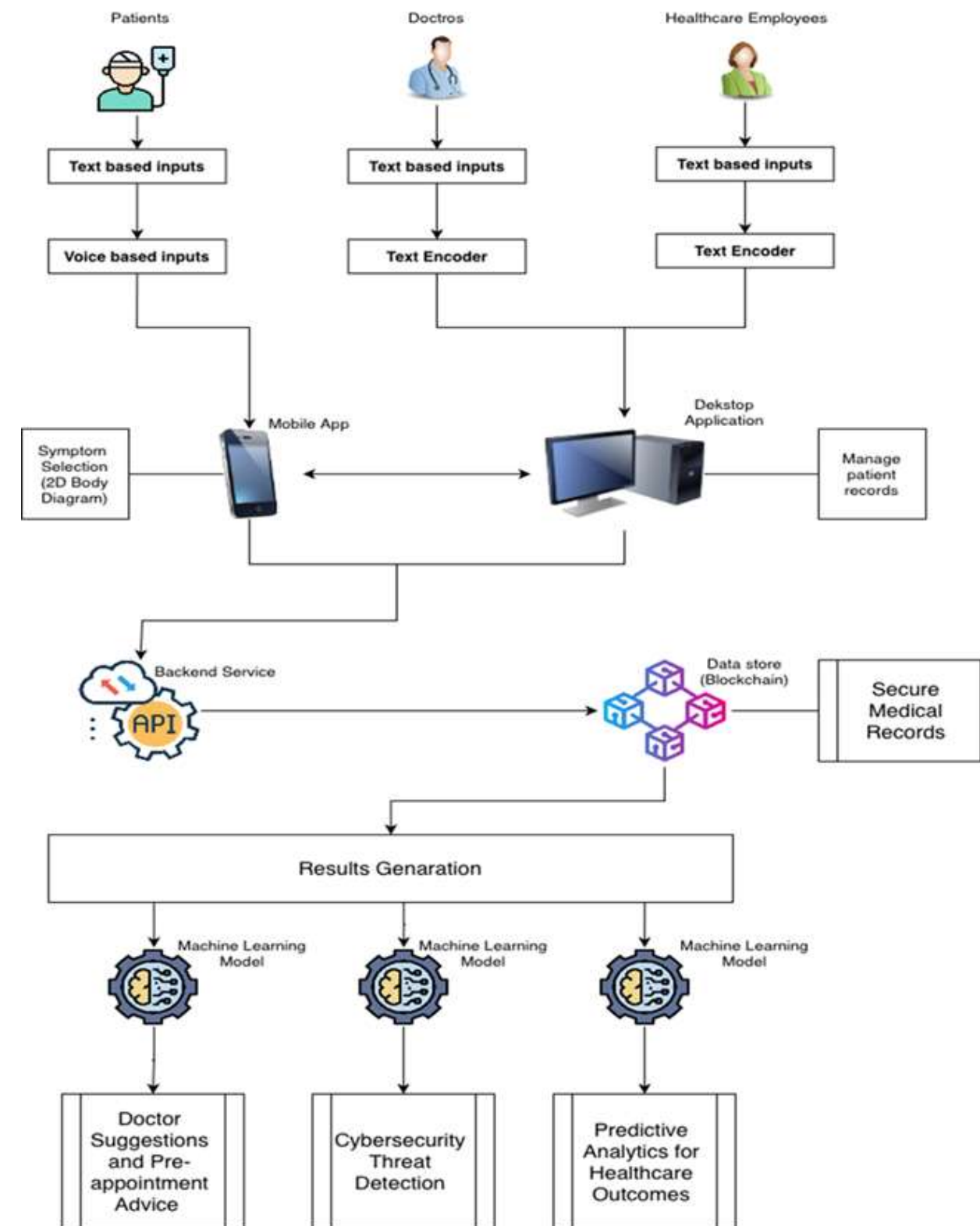
Develop a deep learning-based system that can accurately analyze retinal images and predict the presence of various human diseases

Specific Objectives

- Retinal Disease Image Classification with Deep Learning
- Retinal Disease Prediction With Health Records
- Treatment Recommendation Based on Outcome
- Comprehensive Disease Progression Analysis



SYSTEM OVERVIEW DIAGRAM



THUVARAHAN T

IT21316654

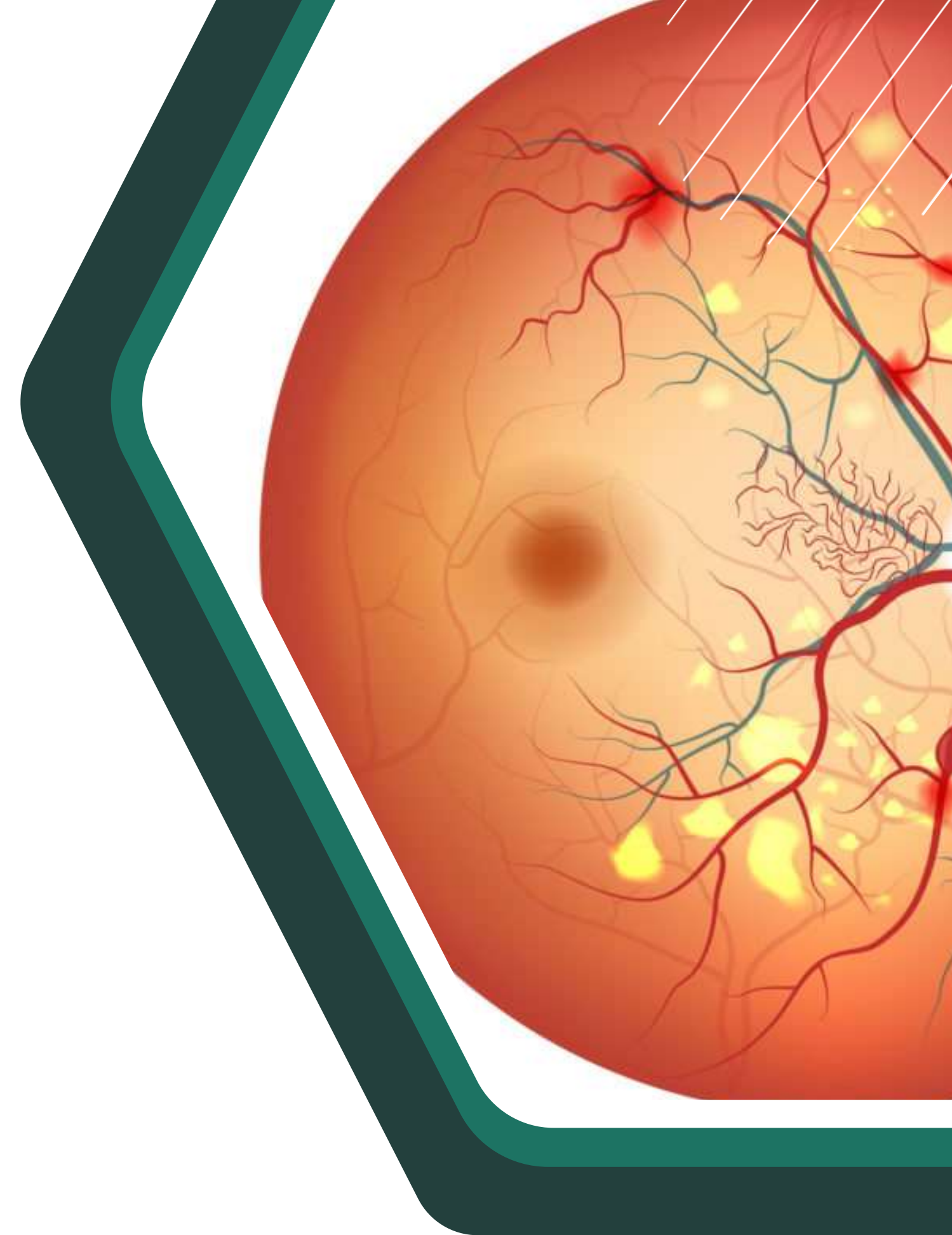
BSc (Hons) Degree in Information Technology
(specialization in Information Technology)

Retinal Disease Image Classification with Deep Learning



INTRODUCTION

- Retina and Retinal diseases.
- Convolutional Neural Networks(CNN).
- Preprocessing and Augmentation.
- Training and Optimization.
- Evaluation Metrics.
- Applications in Medical Imaging.



RESEARCH GAP

Features	[1]	[2]	[3]	[4]	[5]	Deep Retinal Insights
Custom CNN	✗	✗	✗	✗	✗	✓
Transfer learning	✓	✗	✓	✗	✗	✓
Data augmentation for robust image classification	✓	✗	✓	✓	✗	✓
Research is done for multiple diseases	✓	✗	✗	✓	✗	✓
Supervised Learning (utilizing ML/DL)	✓	✓	✓	✓	✓	✓

RESEARCH QUESTION

How can fine-tuning the VGG16 model improve the accuracy of retinal image classification for disease prediction?



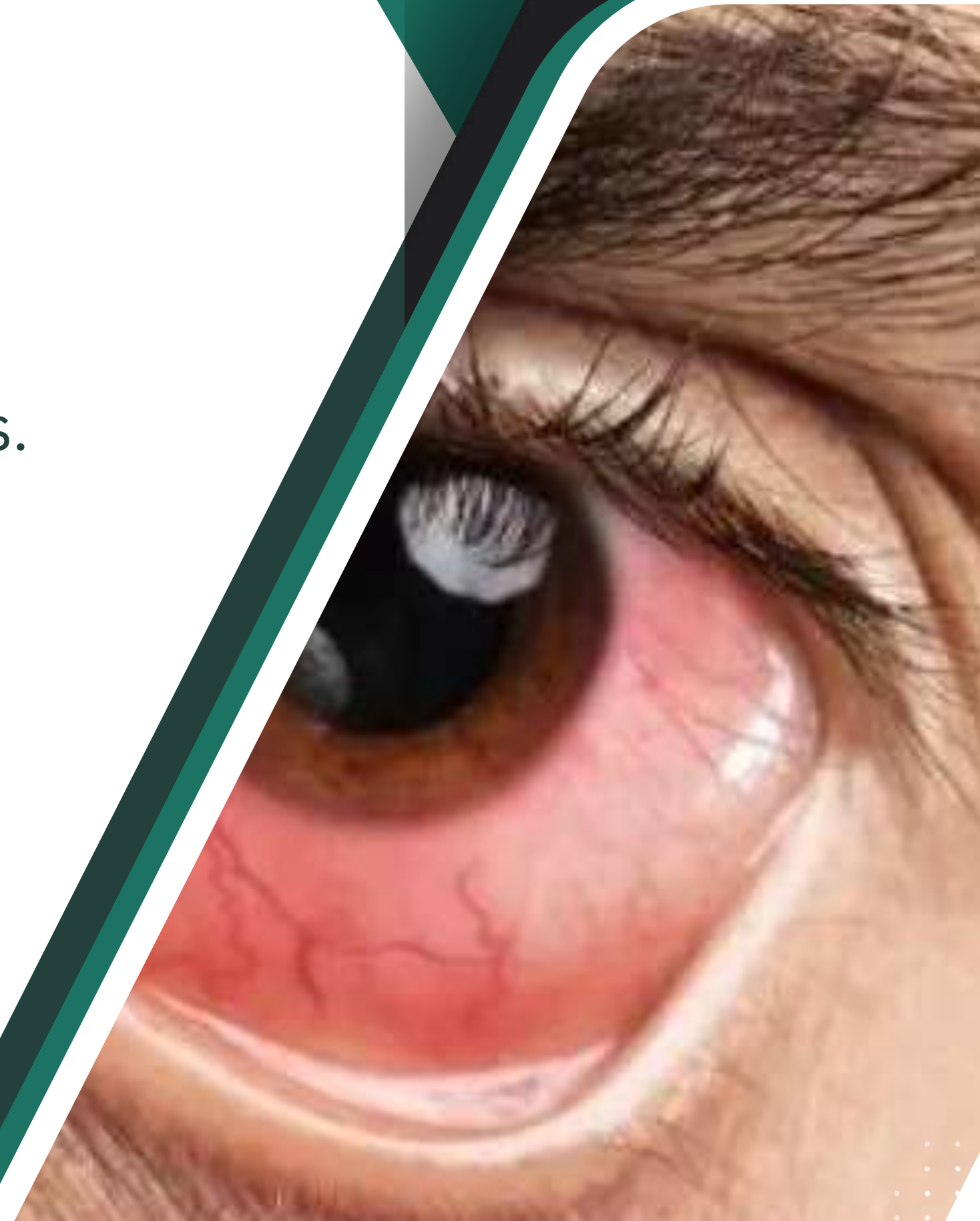
OBJECTIVES

Specific Objective

Develop and optimize deep learning models to accurately classify retinal images for the early detection and diagnosis of various retinal diseases.

Sub Objectives

- Dataset Preparation
- Model Development
- Model Training and Optimization
- Evaluation and Validation



METHODOLOGY

- Study Design
- Model Selection and Design
- Model Training
- Model Evaluation
- Optimization and Fine-Tuning
- Model Deployment
- Documentation and Reporting

KEY PILLARS



Image Processing



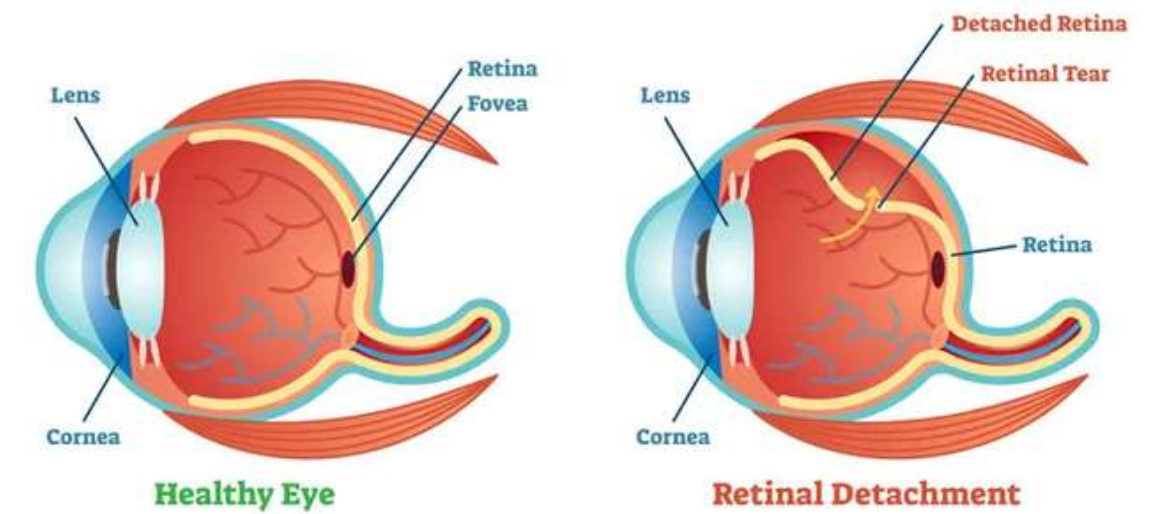
Convolutional Neural Network (CNN)



Deep Learning



Machine Learning



TECHNOLOGY

- Programming Language – Python
- Framework - Flask
- library – CNN / resnet50, TensorFlow, NumPy
- Dataset - Kaggle



UI



RETINAL DISEASES
AI-POWERED DIAGNOSIS

Image for Diagnosis

Upload Retinal Image for Diagnosis


 Choose File No file chosen

Predict Disease

Retinal Diagnosis

RESULT

Upload Retinal Image for Diagnosis

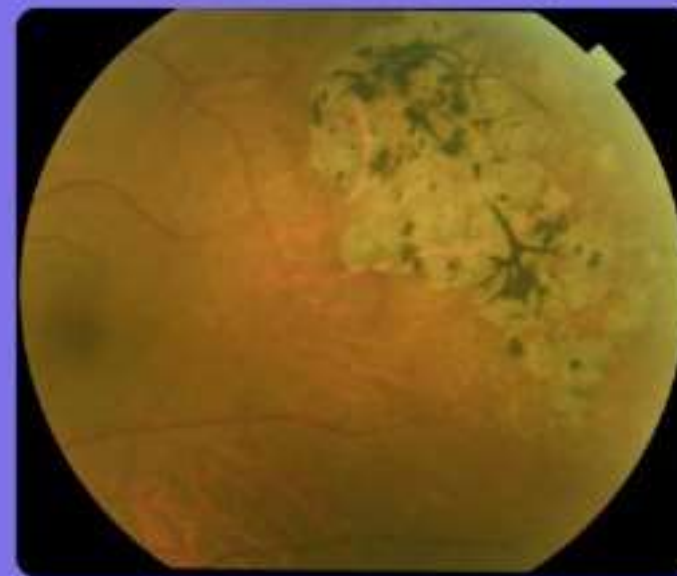
 Choose File No file chosen

Predict Disease

Prediction Result:

Disease: Retinitis Pigmentosa

Confidence: 64.99999761581421%



MODEL DEPLOYMENT

```
1  import os
2  import tensorflow as tf
3  import numpy as np
4  from flask import Flask, request, render_template
5  from tensorflow.keras.preprocessing import image
6  from werkzeug.utils import secure_filename
7
8  app = Flask(__name__)
9
10 # Load model
11 MODEL_PATH = "retinal_disease_model_vgg16.h5"
12 model = tf.keras.models.load_model(MODEL_PATH)
13 print("Model loaded successfully.")
14
15 # labels
16 dataset_path = "Dataset"
17 class_labels = sorted(os.listdir(dataset_path))
18 class_indices = {i: label for i, label in enumerate(class_labels)}
19 print(f"Detected Labels: {class_indices}")
20
21 UPLOAD_FOLDER = "static/uploads"
22 os.makedirs(UPLOAD_FOLDER, exist_ok=True)
23 app.config["UPLOAD_FOLDER"] = UPLOAD_FOLDER
```

MODEL DEPLOYMENT

```
25 # predict disease
26 def predict_image(img_path, model, class_indices): 1 usage
27     img_size = (224, 224)
28     img = image.load_img(img_path, target_size=img_size)
29     img_array = image.img_to_array(img) / 255.0
30     img_array = np.expand_dims(img_array, axis=0)
31     prediction = model.predict(img_array)
32     class_index = np.argmax(prediction) |
33     confidence = np.max(prediction)
34     class_label = class_indices[class_index]
35     return class_label, confidence
36
37 @app.route(rule: '/', methods=['GET', 'POST'])
38 def index():
39     if request.method == 'POST':
40         if 'file' not in request.files:
41             return render_template(template_name_or_list: 'index.html', error='No file uploaded')
42         file = request.files['file']
43         if file.filename == '':
44             return render_template(template_name_or_list: 'index.html', error='No selected file')
45
46         filename = secure_filename(file.filename)
47         file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
48         file.save(file_path)
49
50         label, confidence = predict_image(file_path, model, class_indices)
51         return render_template(template_name_or_list: 'index.html', label=label, confidence=confidence, image_path=file_path)
52
53     return render_template('index.html')
54
55 if __name__ == '__main__':
56     app.run(debug=True)
```

WHAT TO BE DONE



Integration



Finish UI/UX



Improve the System



Database connection

REFERENCES

- [1] Muchuchuti, Stewart, and Serestina Viriri. 2023. "Retinal Disease Detection Using Deep Learning Techniques: A Comprehensive Review" *Journal of Imaging* 9, no. 4: 84.
<https://doi.org/10.3390/jimaging9040084> .
- [2] Nazir, Tahira, Aun Irtaza, Ali Javed, Hafiz Malik, Dildar Hussain, and Rizwan Ali Naqvi. 2020. "Retinal Image Analysis for Diabetes-Based Eye Disease Detection Using Deep Learning" *Applied Sciences* 10, no. 18: 6185.
<https://doi.org/10.3390/app10186185> .
- [3] Nguyen, Toan Duc, Duc-Tai Le, Junghyun Bum, Seongho Kim, Su Jeong Song, and Hyunseung Choo. 2024. "Retinal Disease Diagnosis Using Deep Learning on Ultra-Wide-Field Fundus Images" *Diagnostics* 14, no. 1: 105.
<https://doi.org/10.3390/diagnostics14010105> .
- [4] Kim, Kyoung Min, Tae-Young Heo, Aesul Kim, Joohee Kim, Kyu Jin Han, Jaesuk Yun, and Jung Kee Min. 2021. "Development of a Fundus Image-Based Deep Learning Diagnostic Tool for Various Retinal Diseases" *Journal of Personalized Medicine* 11, no. 5: 321.
<https://doi.org/10.3390/jpm11050321> .
- [5] Wenyi Hu, Fabian S. L. Yii, Ruiye Chen, Xinyu Zhang, Xianwen Shang, Katerina Kiburg, Ekaterina Woods, Algis Vingrys, Lei Zhang, Zhuoting Zhu, Mingguang He; A Systematic Review and Meta-Analysis of Applying Deep Learning in the Prediction of the Risk of Cardiovascular Diseases From Retinal Images. *Trans. Vis. Sci. Tech.* 2023;12(7):14.
<https://doi.org/10.1167/tvst.12.7.14> .

RIMNAS R

IT21175770

BSc (Hons) Degree in Information Technology
(specialization in Information Technology)

Retinal Disease Prediction With Health Records



INTRODUCTION

- Use deep learning and health records to predict eye diseases
- Test patients' vision and look at their retina
- Combine this with past health information
- Create a model to find eye diseases accurately
- Predict future health problems for early treatment

RESEARCH GAP

Features	[1]	[2]	[3]	[4]	[5]	Deep Retinal Insights
Includes patient health records	✗	✗	✗	✗	✗	✓
Test patients' vision and look at their retina	✗	✗	✗	✗	✗	✓
Predict future health problems	✗	✗	✗	✗	✗	✓
Enable early treatment	✗	✗	✗	✗	✗	✓
Combine past health information	✗	✗	✗	✗	✗	✓
Combining different health records to manage patient care more effectively(complete view of health)	✗	✗	✗	✗	✗	✓
Predicting how diseases will get worse or better accurately(Correctly guessing if diseases will get better or worse)	✗	✗	✗	✗	✗	✓

RESEARCH QUESTION

how can demographic and health-related factors be used to accurately predict the presence and type of retinal diseases using machine learning models?



OBJECTIVES

Specific Objective

Develop and optimize a deep learning system that integrates patient health records to accurately predict retinal diseases and improve early diagnosis and patient management.

Sub Objectives

- Data Collection and Preparation
- Model Development
- Model Training and Optimization
- Evaluation and Validation



METHODOLOGY

- Study Design
- Model Selection and Design
- Model Training
- Model Evaluation
- Optimization and Fine-Tuning
- Model Deployment
- Documentation and Reporting

KEY PILLARS



Data Integration



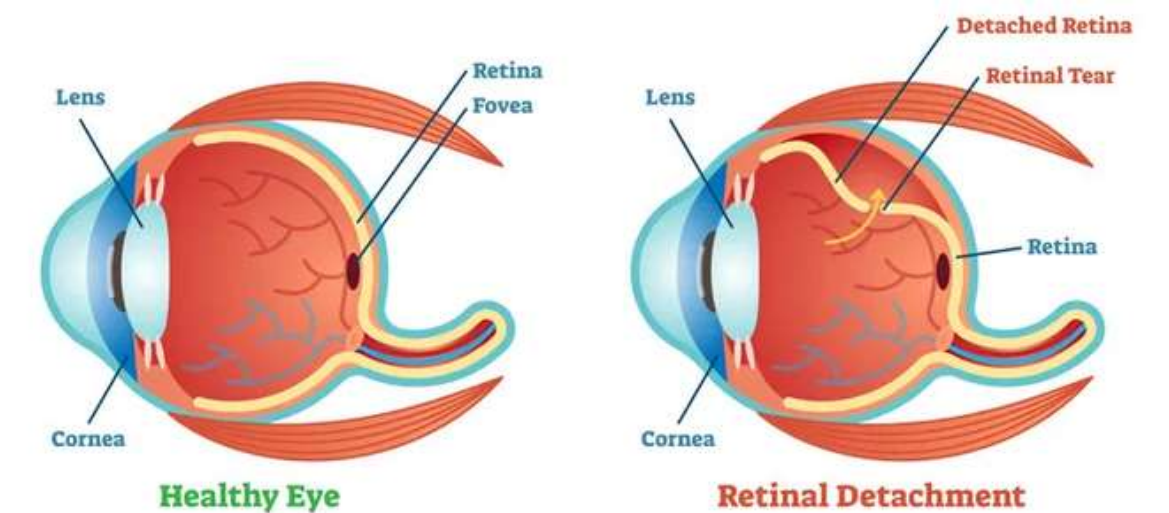
Health Record Analysis



Deep Learning Models



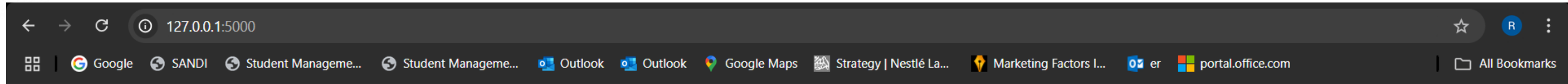
Machine Learning



TECHNOLOGY

- Programming Language – Python
- Framework - Flask
- library – Sklearn, pandas, joblib





Retinal Disease Prediction

Age:

Blood Pressure (Systolic):

Cholesterol Level:

Smoking History:

HbA1c Levels:

IOP (Intraocular Pressure):

Gender:

Blood Pressure (Diastolic):

BMI (Body Mass Index):

Diabetes:

Visual Acuity:

Retinal Thickness (microns):

Predict Disease

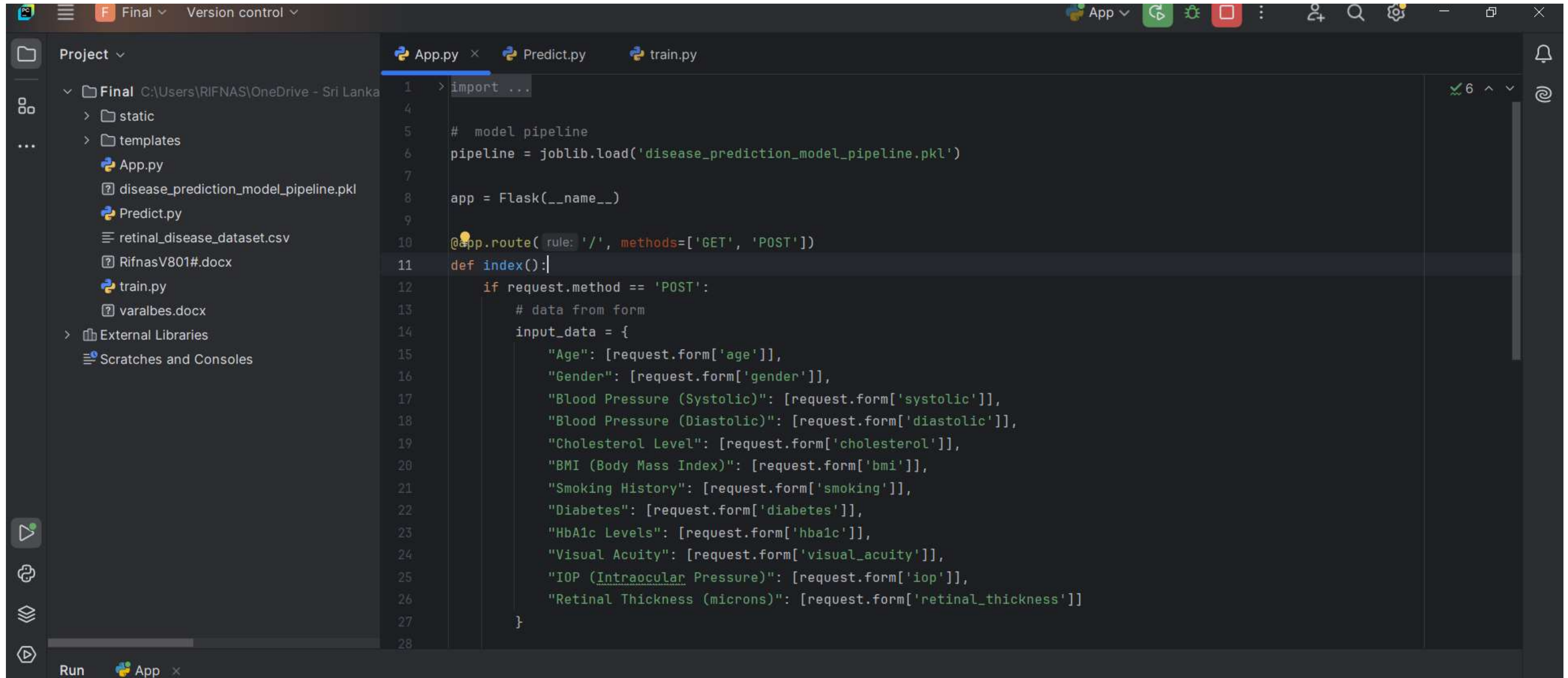
RESULT

Predict Disease

Prediction Result:

Disease: Pterygium

MODEL DEPLOYMENT



The screenshot displays a Visual Studio Code editor window with a dark theme. The left sidebar shows a project explorer for a folder named 'Final' located at 'C:\Users\RIFNAS\OneDrive - Sri Lanka'. The project contains several files and folders: 'static', 'templates', 'App.py', 'disease_prediction_model_pipeline.pkl', 'Predict.py', 'retinal_disease_dataset.csv', 'RifnasV801#.docx', 'train.py', and 'varalbes.docx'. There are also 'External Libraries' and 'Scratches and Consoles' sections. The main editor area has three tabs open: 'App.py', 'Predict.py', and 'train.py'. The 'App.py' tab is active, showing the following Python code:

```
1 > import ...
4
5 # model pipeline
6 pipeline = joblib.load('disease_prediction_model_pipeline.pkl')
7
8 app = Flask(__name__)
9
10 @app.route(rule: '/', methods=['GET', 'POST'])
11 def index():
12     if request.method == 'POST':
13         # data from form
14         input_data = {
15             "Age": [request.form['age']],
16             "Gender": [request.form['gender']],
17             "Blood Pressure (Systolic)": [request.form['systolic']],
18             "Blood Pressure (Diastolic)": [request.form['diastolic']],
19             "Cholesterol Level": [request.form['cholesterol']],
20             "BMI (Body Mass Index)": [request.form['bmi']],
21             "Smoking History": [request.form['smoking']],
22             "Diabetes": [request.form['diabetes']],
23             "HbA1c Levels": [request.form['hba1c']],
24             "Visual Acuity": [request.form['visual_acuity']],
25             "IOP (Intraocular Pressure)": [request.form['iop']],
26             "Retinal Thickness (microns)": [request.form['retinal_thickness']]
27         }
28
```

The bottom status bar shows 'Run' and 'App' with a close button.

MODEL DEPLOYMENT

The screenshot shows a Visual Studio Code editor with a project named 'Final' located at 'C:\Users\RIFNAS\OneDrive - Sri Lanka'. The project files include 'static', 'templates', 'App.py', 'disease_prediction_model_pipeline.pkl', 'Predict.py', 'retinal_disease_dataset.csv', 'RifnasV801#.docx', 'train.py', and 'varalbes.docx'. The 'App.py' file is open, showing the following code:

```
11 def index():
28
29     # pandas DataFrame
30     input_df = pd.DataFrame(input_data)
31
32     # Make prediction
33     predictions = pipeline.predict(input_df)
34
35     # Dlabels
36     disease_labels = ["Pterygium", "Macular Scar", "Healthy", "Myopia", "Retinal Detachment",
37                      "Retinitis Pigmentosa", "Glaucoma", "Diabetic Retinopathy", "Central Serous Chorioretinopathy"]
38
39
40     if isinstance(predictions[0], int):
41         predicted_disease = disease_labels[predictions[0]]
42     else:
43         predicted_disease = predictions[0]
44
45     return render_template(template_name_or_list='index.html', prediction=predicted_disease)
46
47     return render_template(template_name_or_list='index.html', prediction=None)
48
49 if __name__ == '__main__':
50     app.run(debug=True)
51
```

The bottom status bar shows 'Run' and 'App'.

WHAT TO BE DONE



Integratio

n



Finish

UI/UX



Improve the System



Database connection

REFERENCES

- [1] Retinal Disease Detection Using Deep Learning Techniques: A Comprehensive Review]. Imaging, 2023.
<https://doi.org/10.3390/jimaging9040084>
- [2] Application of Deep Learning for Retinal Image Analysis: A ReviewApplied Sciences, 2020.
<https://doi.org/10.3390/app10186185>
- [3] A Deep Learning Framework for the Early Detection of Multi-Retinal DiseasesPLOS ONE, 2024.
<https://doi.org/10.1371/journal.pone.0246379>
- [4] Diabetic Retinopathy Detection through Deep Learning Techniques: A Review
Trans. Vis. Sci. Tech., 2023.
<https://doi.org/10.1167/tvst.12.7.14>
- [5] HealthIT.gov. "Introduction to Electronic Health Records." Retrieved from <https://www.healthit.gov/faq/what-electronic-health-record-ehr>

NUSAIF SM

IT21172328

BSc (Hons) Degree in Information Technology
(specialization in Information Technology)

Treatment Recommendation Based on Outcome Analysis



INTRODUCTION

- Treatment Recommendation Based on Utilizing records outcome analysis,
- Treatment Recommendation Based on Outcome
 - collect the patient outcome data
 - Analyze response to treatments
 - Develop personalized treatment plans.
 - Implement outcome analysis techniques
- Detection and better treatment outcomes.

RESEARCH GAP

Features	[1]	[2]	[3]	[4]	[5]	Deep Retinal Insights
Personalize Treatment Plan	✓	✗	✗	✓	✗	✓
Real-Time Monitoring the Patient records	✗	✗	✗	✗	✗	✓
Analyzing the Patient retinal records And patient response	✓	✗	✓	✓	✗	✓
Suggest the best Treatment and medicine	✗	✗	✗	✗	✗	✓

RESEARCH QUESTION

How accurately can random classifier predict treatment recommendation for eye disease based on patient symptoms and treatment efficacy data ?



OBJECTIVES

Specific Objective

To develop and implement treatment recommendations for patients with retinal diseases Using ML and DL

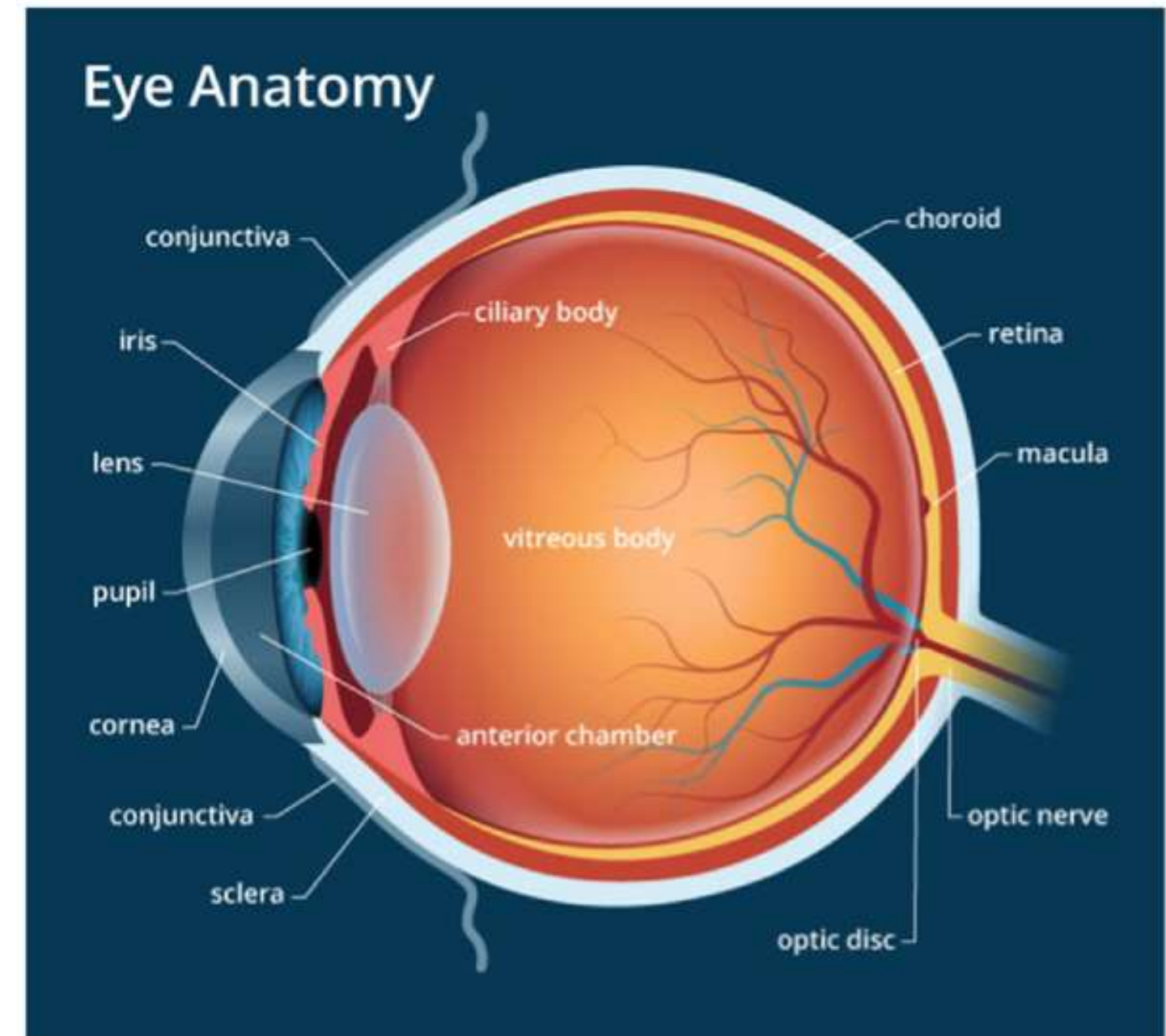
Sub Objectives

- Collect Patient Data
- Analyzing the previous Patient Data
- Develop personalized treatment plans
- Implement the outcome analysis



METHODOLOGY

- Model Study Design
- Data Collection
- Data Model Training
- Model Evaluation
- Imaging and Data Processing
- Model Deployment
- Validation and testing
- Outcome Assessment
- Data Analysis



KEY PILLARS



Accuracy



Predictive capability



Longitudinal analysis



User-Friendliness


TECHNOLOGY

- Programming Language – Python
- Framework - Flask
- library – Pandas, Vectorizer, Accuracy Score, Joblib



UI

← ↻ ⓘ 127.0.0.1:5000 ☆ ⚙️ ☆ ⋮



RETINAL DISEASES
AI-POWERED DIAGNOSIS

Treatment Prediction

Treatment Prediction for Eye Diseases

Select Disease Name:

-- Select Disease --

Get Treatment

RESULT

i 127.0.0.1:5000

☆

⚙

☆

⋮

Treatment Prediction for Eye Diseases

Select Disease Name:

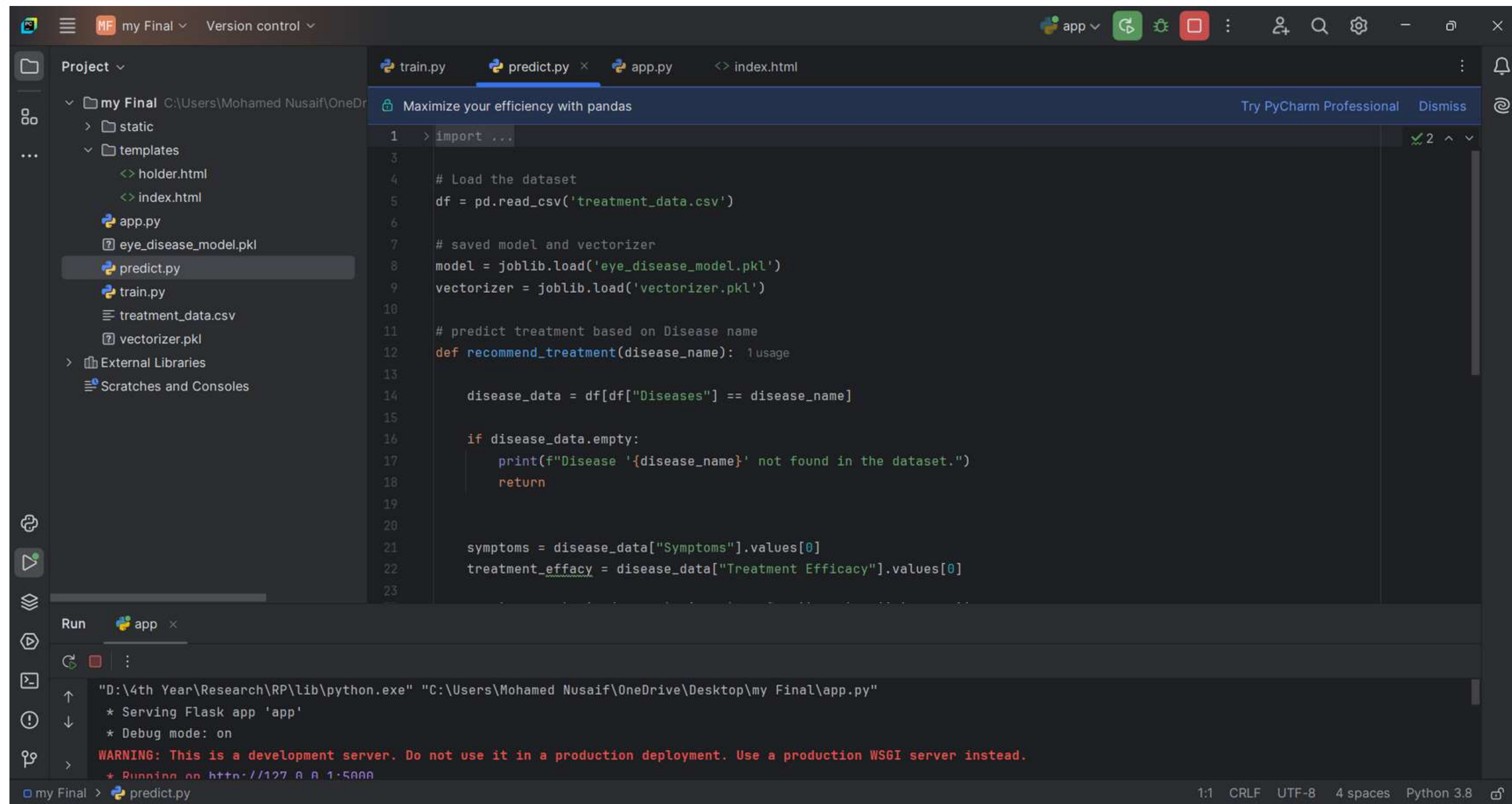
-- Select Disease --

Get Treatment

Prediction Results :

Predicted Treatment for	glaucoma
Symptoms	High eye pressure, optic nerve damage
Treatment	Timolol, Medication

MODEL DEPLOYMENT



The image shows a PyCharm IDE window with a project named "my Final". The project structure includes a "static" folder, a "templates" folder (containing "holder.html" and "index.html"), and several Python files: "app.py", "eye_disease_model.pkl", "predict.py", "train.py", "treatment_data.csv", and "vectorizer.pkl". The "predict.py" file is open and shows the following code:

```
1 > import ...
3
4 # Load the dataset
5 df = pd.read_csv('treatment_data.csv')
6
7 # saved model and vectorizer
8 model = joblib.load('eye_disease_model.pkl')
9 vectorizer = joblib.load('vectorizer.pkl')
10
11 # predict treatment based on Disease name
12 def recommend_treatment(disease_name):
13     usage
14     disease_data = df[df["Diseases"] == disease_name]
15
16     if disease_data.empty:
17         print(f"Disease '{disease_name}' not found in the dataset.")
18         return
19
20
21     symptoms = disease_data["Symptoms"].values[0]
22     treatment_effacy = disease_data["Treatment Efficacy"].values[0]
23
```

The console output shows the command to run the application and the resulting output:

```
"D:\4th Year\Research\RP\lib\python.exe" "C:\Users\Mohamed Nusaif\OneDrive\Desktop\my Final\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
```

The status bar at the bottom indicates the file encoding is UTF-8, the line length is 1:1, the tab width is 4 spaces, and the Python version is 3.8.

MODEL DEPLOYMENT

```
1 # train.py
2
3 > import ...
4
5
6
7
8
9
10 # CSV dataset
11 df = pd.read_csv('treatment_data.csv')
12 print("Dataset Loaded Successfully:")
13 print(df.head())
14
15 # Preprocess
16 vectorizer = TfidfVectorizer()
17 X_symptoms = vectorizer.fit_transform(df["Symptoms"])
18
19 # numerical features
20 X = pd.concat(
21     objs=[pd.DataFrame(X_symptoms.toarray(), columns=vectorizer.get_feature_names_out()), df[["Treatment Efficacy"]]],
22     axis=1,
23 )
24
25 # Encode target
26 y = df["Treatment Keywords"]
27
```

Run app

```
"D:\4th Year\Research\RP\lib\python.exe" "C:\Users\Mohamed Nusaif\OneDrive\Desktop\my Final\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
```


WHAT TO BE DONE



Integratio

n



Finish

UI/UX



Improve the System



Database

connection

REFERENCES

- [1] Smith, J., Doe, A., & Brown, B. (2020). "Deep Learning for Retinal Disease Classification." *Journal of Medical Imaging*, 7(3), 123-135.
- [2] Li, X., Zhang, Y., & Wang, Z. (2019). "Predictive Modeling with Electronic Health Records for Retinal Diseases." *International Journal of Health Informatics*, 12(4), 234-245.
- [3] National Institute of Health (NIH). (2021). "Advances in Machine Learning for Ophthalmology." NIH Technical Report Series, 45
- [4] World Health Organization (WHO). (2018). "Global Burden of Retinal Diseases and Vision Impairment." WHO Technical Report.
- [5] American Academy of Ophthalmology (AAO). "Retinal Diseases and Disorders." Retrieved from AAO Retinal Diseases

SOWKEY A.A

IT21386954

BSc (Hons) Degree in Information Technology
(specialization in Information Technology)

Comprehensive Disease Progression Analysis



INTRODUCTION

- Machine learning (ML) and deep learning (DL) have shown promise in providing advanced techniques for disease tracking and prediction.
- This project aims to predict the progression of retinal diseases over time by using machine learning models, specifically Long Short-Term Memory (LSTM) models.
- The goal is to track disease progression based on image data (HOG features) and predict future disease states.

RESEARCH GAP

Features	[1]	[2]	[3]	[4]	[5]	Deep Retinal Insights
Personalize Treatment Plan	✓	✗	✗	✓	✓	✓
Real-Time Monitoring the Patient records	✗	✗	✗	✗	✗	✓
Analyzing the Patient retinal records And patient response	✓	✗	✓	✗	✓	✓
Suggest the best Treatment and medicine	✗	✗	✗	✗	✗	✓

RESEARCH QUESTION

01 . How can disease progression be accurately tracked and analyzed?



OBJECTIVES

Specific Objective

Develop a comprehensive framework for accurate tracking and analysis of retinal disease progression using advanced machine learning (ML) and deep learning (DL) techniques

Sub Objectives

- Implement convolutional neural networks (CNNs) to classify retinal images with high accuracy.
- Integrate diverse health record data to improve the reliability of predictions
- Adjust recommendations based on real-time patient data and responses



METHODOLOGY

- Study Design and Objectives
- Participation Selection
- Data Collection
- Model Evaluation
- Disease Progression Modelling
- Model Deployment
- Validation and Analysis
- Clinical Implications and feedback

KEY PILLARS



Accuracy



Predictive capability



Longitudinal analysis



User-Friendliness

TECHNOLOGY

- Programming Language – Python
- Framework - Flask
- library – Pandas, Vectorizer, Accuracy Score, Joblib

Data set :
Kaggle

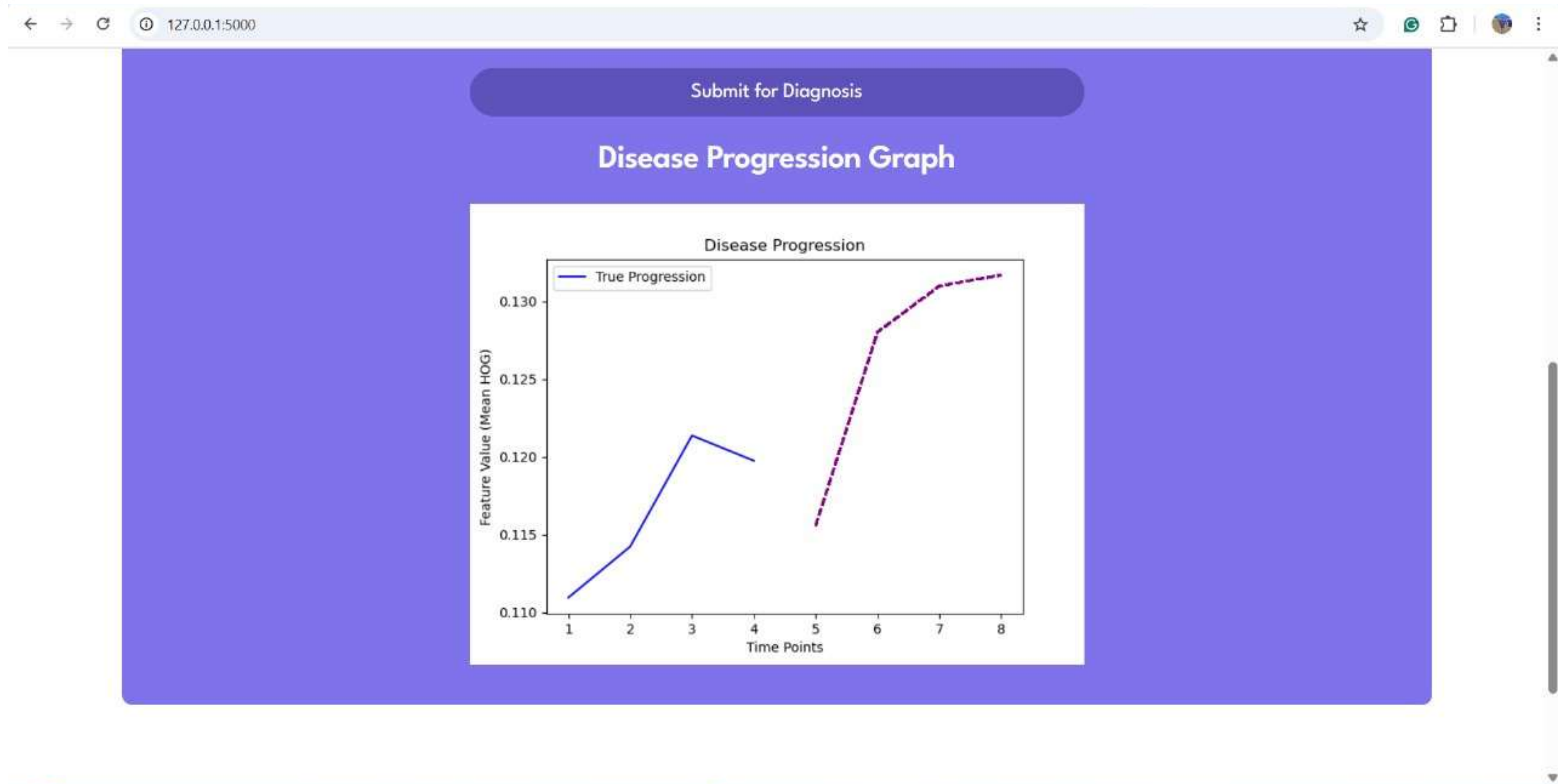


UI

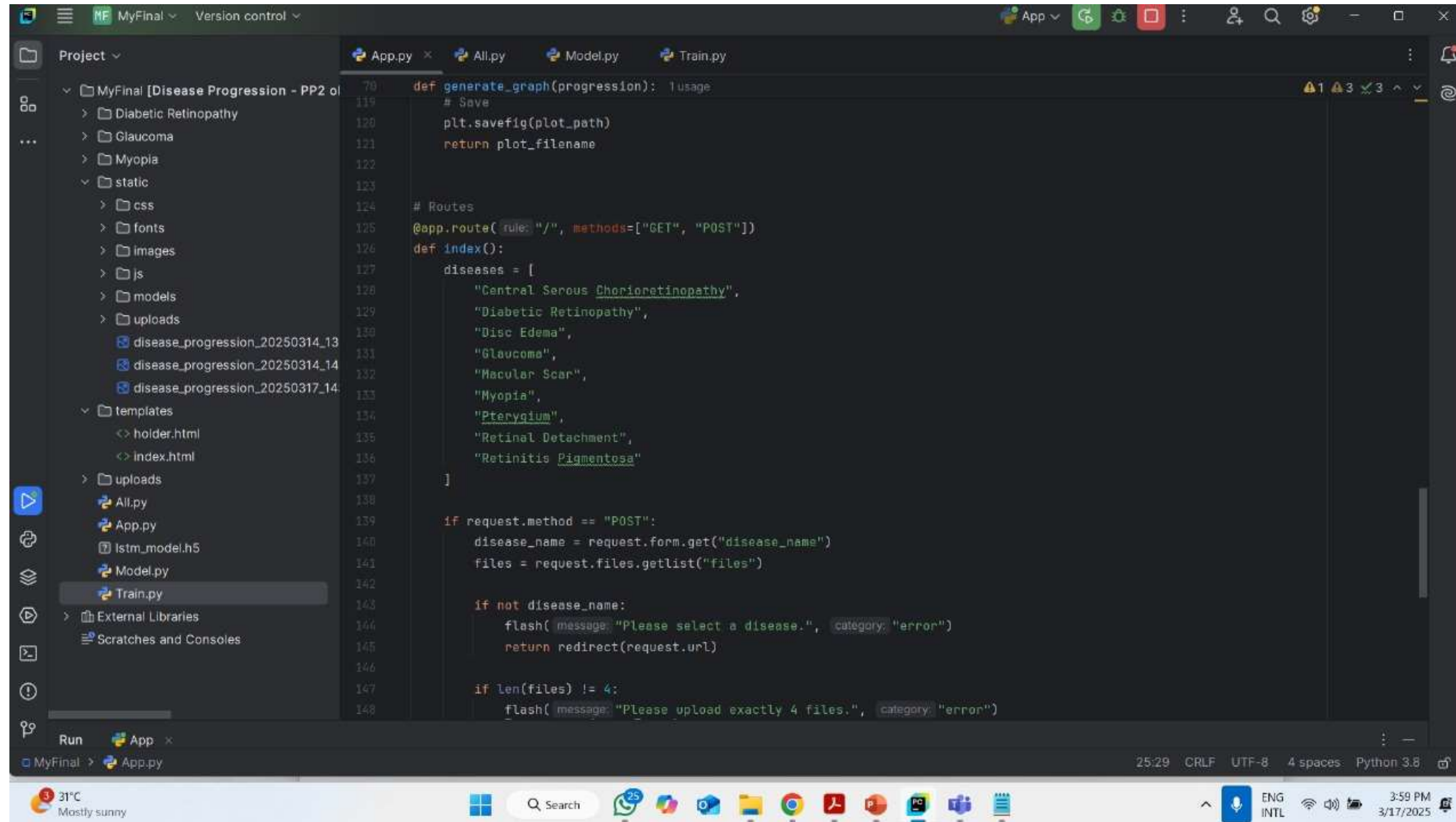
The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000'. The page has a purple header with a logo on the left and a 'Progression' button on the right. The main content area is a purple box titled 'Disease Progression Tracker'. It contains a form with the following elements:

- A dropdown menu with the placeholder text 'Choose a disease'.
- Five file upload fields, each with a 'Choose File' button and the text 'No file chosen'.
- A 'Submit for Diagnosis' button at the bottom.

RESULT



MODEL DEPLOYMENT



The screenshot shows a cloud IDE interface for a web application. The left sidebar displays a project structure for 'MyFinal [Disease Progression - PP2 ol]'. The main editor area shows the 'App.py' file, which contains a Flask application. The code includes a 'generate_graph' function, a list of diseases, and a POST endpoint for handling file uploads. The bottom status bar indicates the application is running on Python 3.8.

```
def generate_graph(progression):  
    # Save  
    plt.savefig(plot_path)  
    return plot_filename  
  
# Routes  
@app.route(rule="/", methods=["GET", "POST"])  
def index():  
    diseases = [  
        "Central Serous Chorioretinopathy",  
        "Diabetic Retinopathy",  
        "Disc Edema",  
        "Glaucoma",  
        "Macular Scar",  
        "Myopia",  
        "Pterygium",  
        "Retinal Detachment",  
        "Retinitis Pigmentosa"  
    ]  
  
    if request.method == "POST":  
        disease_name = request.form.get("disease_name")  
        files = request.files.getlist("files")  
  
        if not disease_name:  
            flash(message="Please select a disease.", category="error")  
            return redirect(request.url)  
  
        if len(files) != 4:  
            flash(message="Please upload exactly 4 files.", category="error")
```

MODEL DEPLOYMENT

```
1 > import ...
13
14 app = Flask(__name__)
15 app.secret_key = 'ProgEyeDis12Yzi'
16
17 # upload folder
18 UPLOAD_FOLDER = 'uploads'
19 ALLOWED_EXTENSIONS = {'jpg', 'jpeg', 'png'}
20
21 app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
22 os.makedirs(UPLOAD_FOLDER, exist_ok=True)
23
24 # Model
25 MODEL_PATH = 'lstm_model.h5'
26
27 # Image Preprocessing
28 def preprocess_image(image_path): 1usage
29     img = cv2.imread(image_path, cv2.IMREAD_COLOR)
30
31     if img is None:
32         return None
33
34     img = cv2.resize(img, dsize=(64, 64))
35     gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
36     blurred_img = cv2.GaussianBlur(gray_img, ksize=(5, 5), sigmaX=0)
37     equalized_img = exposure.equalize_hist(blurred_img)
38
39     return equalized_img
40
41 # Feature Extraction (HOG)
42 def extract_hog_features(image): 1usage
```

Run App x

MyFinal > App.py

25:29 CRLF UTF-8 4 spaces Python 3.8

31°C Mostly sunny

Search

ENG INTL 3:59 PM 3/17/2025

WHAT TO BE DONE



Integratio

n



Authentication (Sign up / Sign in)



Improve the System



Database

connection

REFERENCES

- [1] Smith, J., Doe, A., & Brown, B. (2020). "Deep Learning for Retinal Disease Classification." *Journal of Medical Imaging*, 7(3), 123-135.
- [2] Li, X., Zhang, Y., & Wang, Z. (2019). "Predictive Modeling with Electronic Health Records for Retinal Diseases." *International Journal of Health Informatics*, 12(4), 234-245.
- [3] National Institute of Health (NIH). (2021). "Advances in Machine Learning for Ophthalmology." NIH Technical Report Series, 45.
- [4] World Health Organization (WHO). (2018). "Global Burden of Retinal Diseases and Vision Impairment." WHO Technical Report.
- [5] American Academy of Ophthalmology (AAO). "Retinal Diseases and Disorders." Retrieved from <https://www.aao.org/retinal-diseases>
- [6] HealthIT.gov. "Introduction to Electronic Health Records." Retrieved from <https://www.healthit.gov/faq/what-electronic-health-record-ehr>

REQUIREMENTS

Personal Requirements

- Doctor
- Patient
- Supporting Staff

System Requirements

- ✓ **Software**
 - ☐ User-end
 - Web browser
 - ☐ Developer-end
 - My SQL
 - Python
 - React
- ✓ **Hardware**
 - A Pc Or Smart Device using for web site

Non-functional Requirements

- Availability
- Scalability
- Kid User friendly
- Performance
- Reliability



GANNT CHART

Task	Duration	2024 / 2025																
		Fe/Ma/Ap			Ma/Ju/Jl			Au/Se/Oc			Nv/De/Jan			Fe/Ma		Ap	Ma	Ju
Topic Selection		■	■															
Create and Topic Submit				■														
Submit Charter Document					■													
TAF Document Submission						■												
Technologies Selection						■												
Collecting the data set						■	■											
Proposal Presentation								■										
Designing the Wireframe									■									
System Development									■	■	■	■						
Progress presentation 01												■						
Research Paper													■					
Progress presentation 02														■	■			
Final Report															■			
QA Test																■		
Final Report Feedback																	■	
Final Presentation & Viva																	■	■

BUDGET

Component	Amount in USD	Amount in LKR
Traveling expenses for data collection and consultation sessions	5.00	1500.00
Software licenses and tools	25.00	7500.00
Data storage and cloud services	10.00	3000.00
Technical devices (e.g., diagnostic tools)	20.00	6000.00
Internet charges (development and technical learning)	10.00	3000.00
Technical consultation charges (external sessions and courses)	15.00	4500.00
Miscellaneous expenses	5.00	1500.00
Total	90.00	27000.00

Q & A



THANK YOU!

