Diabeto-Vision

By:

Mudabbir Ahmed
35830
Saad Munaf
38748
Saad Bin Rizwan
37968

Supervised by:
Tajamul Shahzad

Faculty of Computing
Riphah International University, Islamabad

Fall 2025

A Dissertation Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfillment of the Requirement for the Award

of the Degree of

Bachelor of Science in Computer Science

Faculty of Computing
Riphah International University, Islamabad

Date: [16-05-2025]

Final Approval

This is to certify that we have read the report submitted by ***Mudabbir Ahmed (35830), Saad Munaf (3878) and Saad Bin Rizwan (37968)*** for the partial fulfillment of the requirements for the degree of the Bachelor of Science in Computer Science (BSCS). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelor of Science in Computer Science (BSCS).

Committee:

1 ______________________________

[Tajamul Shahzad]
(Supervisor)

2 ______________________________

[Dr.Musharraf Ahmed]
(Head of Department/chairman)

Declaration

We hereby declare that this document “**Diabeto-Vision**” neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely based on our personal efforts, under the proficient guidance of our teachers, especially our supervisor **Tajamul Shahzad**. If any part of the system is proved to be copied out from any source or found to be a reproduction of any project from anywhere else, we shall stand by the consequences.

[Mudabbir Ahmed]

[35830]

[Saad Munaf]

[38748]

[Saad Bin Rizwan]

[37968]

Dedication

Our project is dedicated to our work to our parents, seniors, friends and our supervisor "Tajamul Shahzad" who has been our continual source of inspiration and whose support has helped this project succeed. This project would not have been possible without their trust and support.

Acknowledgement

First, we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We owe a heartfelt thank you to our project supervisor “Tajamul Shahzad”. His guidance has been a beacon of light throughout our project journey. His patience and knowledge were key in overcoming the challenges we faced. We are truly thankful for his dedication and the time he invested in us. We also extend our deepest gratitude to our parents and family. Their unwavering belief in us and the values of hard work and integrity they have nurtured within us have been our guiding stars. It is with their blessings and constant encouragement that we have been able to achieve this milestone.

[Mudabbir Ahmed]

[35830]

[Saad Munaf]

[38748]

[Saad Bin Rizwan]

[37968]

Table of Contents

List of Tables

List of Figures

Abstract

Diabetic Retinopathy is a serious complication of diabetes that affects the eyes, potentially leading to vision loss if left untreated. Early detection and accurate grading are crucial for timely treatment and better outcomes.

Diabeto Vision is a web application developed to help in scanning and determining the degree of severity of diabetic retinopathy using the latest machine learning algorithms. The system retrieves the fundus images of the eye and analyses them to see if the patient suffers from the problem and if yes, the level of severity is established. This easy-to-use platform seeks to give both patients and health care providers a fast and accurate method of detecting and coordinating the treatment of Diabetic Retinopathy.

Chapter 1:

Introduction

Chapter 1: Introduction

According to the WHO, the number of visually impaired people worldwide is estimated to be 2.2 billion, of whom at least 1 billion have vision impairment could have been prevented or are yet to be addressed. The world faces considerable challenges in terms of eye care, including inequalities in the coverage and quality of prevention, treatment, and forestall of rehabilitation services. Early detection and diagnosis of ocular pathologies would enable forestall visual impairment. The traditional diagnosis systems are slow, time-consuming, and expensive require a certain level of expertise to use, whereas the proposed system will provide an easy-to-use, reliable, fast, and cheap alternative for the users. It will be a web-based project which will integrate image-processing techniques. Medical professionals can also benefit from the system, as it will enable them to verify the results from conventional systems. The users are required to input fundus and retinal photographs of their eyes, and the system will preprocess them, extract features, and make a diagnosis based on the available datasets.

1.1 Goals and Objectives

1.1.1 Goals

- Design a user friendly React based Frontend.
- Implement secure logging and registration system.
- Allow users to upload fundus image for analysis.
- Integrate trained ML Model for detection and severity, grading.
- Allow Doctor to chat with the patient

1.1.2 Objectives

- Develop a web-based platform for detecting and grading Diabetic Retinopathy
- Utilize Machine Learning model to analyze fundus image for accurate diagnosis

1.2 Scope of the Project

Healthcare Support: Helps doctors and patients detect diabetic retinopathy early and understand its severity

Web-Based Access: Users can access the platform from anywhere to upload fundus images and get results.

Machine Learning Powered: Uses trained ML models to provide accurate and reliable diagnoses.

User-Friendly Design: Simple and intuitive interface, making it easy for anyone to use.

Data Security: Ensures that all user data and images are kept private and secure.

Scalable and Future-Ready: Designed to handle more users and datasets as it grows.

Educational Purpose: Useful for medical research, training, and learning about diabetic retinopathy.

Chapter 2:
Literature Review

Chapter 2: Literature Review

2.1 Introduction

Diabetic Retinopathy (DR) is one of the leading causes of vision impairment globally, with an increasing prevalence due to the rise in diabetes cases. The detection of DR in its early stages is crucial to prevent vision loss. This report explores various CNN architectures and methodologies to automate DR detection using retinal images, enhancing accuracy, efficiency, and scalability.

2.2 Background and Problem Elaboration

Traditional DR diagnosis involves manual examination of retinal fundus images by ophthalmologists. However, this process is time-consuming, prone to human error, and dependent on specialized skills. The advent of Convolutional Neural Networks (CNNs) has revolutionized image recognition tasks, offering a promising solution for automating DR detection. CNNs can identify complex patterns in retinal images, enabling early detection and classification of DR stages with higher precision and speed.

2.3 Detailed Literature Review

Contributions of various studies aimed at improving DR detection through advanced deep learning methods. It focuses on different CNN architectures, preprocessing techniques, datasets, and their performance metrics

2.3.1 Related Research Work 1:

2.3.1.1 Literature Review of Paper 1:

(Singh et al., 2024) [1] paper proposes a deep learning model named DRCNNRB for the automated detection and classification of diabetic retinopathy (DR) into five classes: No DR, Mild, Moderate, Severe, and Proliferative DR. The model leverages a Convolutional Neural Network (CNN) architecture enhanced with Residual Blocks (Res-Blocks), which include both convolutional and identity blocks connected by skip connections to address challenges such as vanishing gradients and performance degradation in deeper networks. The final classification is achieved through a SoftMax activation function in the dense layer after applying average pooling and flattening layers. To address class imbalance and improve image quality, data augmentation techniques such as zooming, sharing, rotation, flipping, and rescaling were applied along with preprocessing steps. The model was trained and evaluated on the Diabetic Retinopathy 2015 Colored Resized dataset and achieved an accuracy of 96.71%, with high performance across other metrics like precision, recall, and F1-score. The results demonstrate that DRCNNRB outperforms many existing state-of-the-art methods and holds strong potential for real-time clinical diagnosis of DR, reducing the workload on ophthalmologists and improving screening accuracy.

2.3.1.2 Literature Review of Paper 2:

(Perumal et al., 2024) [2] proposed "Enhancing Multi-class Diabetic Retinopathy Detection Using Tuned Hyper-parameters and Modified Deep Transfer Learning" by Yeganeh Modaresnia et al. presents a highly accurate method for classifying diabetic retinopathy (DR) into five severity levels: No DR, Mild, Moderate, Severe, and Proliferative DR. The proposed approach integrates modified AlexNet with genetic algorithms and Bayesian hyperparameter optimization. The model benefits from powerful preprocessing techniques such as CLAHE, contrast enhancement, and color constancy, which significantly improve feature extraction and classification. To improve model generalization and reduce overfitting, L2 regularization and data augmentation strategies like up sampling were employed. The model's decision-making was validated using Grad-CAM, enhancing its clinical trustworthiness. Evaluations across several datasets (Kaggle, Messidor-1, Messidor-2) confirmed its robustness, making the method highly effective for real-time, multi-class DR detection in medical settings.

2.3.1.3 Literature Review of Paper 3:

(Chaudhary et al., 2024) [3] proposes a deep learning model for detecting diabetic retinopathy (DR) in retinal fundus images. The model utilizes Efficient Net B0 as a base architecture for transfer learning, which is fine-tuned to improve classification accuracy. To combat overfitting, the model incorporates Dropout with a 25% deactivation rate and applies L2 regularization with a rate of 0.01 in Dense layers. The final classification is produced by a SoftMax activation in the last Dense layer. Data augmentation techniques are employed to address class imbalance and enhance the model's performance, with SVM classifiers outperforming Decision Trees (DT) for microaneurysm detection. The authors discuss the challenges of detecting microaneurysms and report that the use of CNN models and data preprocessing improved accuracy. The model is evaluated with a batch size of 32, and the results show its capability in accurately classifying DR severity levels.

2.3.1.4 Literature Review of Paper 4:

(Panthi et al., 2025) [4] proposes an advanced model for classifying diabetic retinopathy (DR) severity using ensemble transfer learning and data fusion techniques. The model utilizes three pre-trained convolutional neural networks (CNNs): EfficientNetB2, DenseNet121, and ResNet50, all fine-tuned for the specific task of DR classification. Modifications were made to the first and last layers of these models, with a Global Average Pooling (GAP) layer added before the final SoftMax activation function to handle multi-class classification. The model's performance is enhanced through data augmentation and preprocessing techniques. These improvements contributed to higher accuracy when compared to previous models in multi-class DR classification. The study also highlights the importance of early detection of DR by ophthalmologists and the potential for lifestyle changes, medication, or therapy to prevent vision loss. The authors demonstrate that their proposed model, by integrating data fusion and ensemble methods, achieves superior classification results compared to other DR detection models.

2.3.1.5 Literature Review of Paper 5:

(Shoaib et al., 2025) [5] presents a novel approach for diabetic retinopathy (DR) detection using Generative Adversarial Networks (GANs) combined with transfer learning. The model incorporates a generator and discriminator: the generator creates synthetic images using leaky ReLU and tanh activation functions, while the discriminator distinguishes between real and synthetic images using leaky ReLU and sigmoid. Regularization and data augmentation are employed to mitigate overfitting and enhance model generalization. The study uses three models: InceptionResNetv2, Inceptionv3, and the newly developed DiaGAN-CNN, which integrates a residual-based CNN architecture with skip connections, fine-tuned for DR detection. GANs are leveraged for data augmentation, generating realistic DR images to enrich training datasets. The InceptionResNetv2 model achieved high performance, and the Adam optimizer with binary cross-entropy loss was used in training. The approach demonstrates significant improvements in DR detection, with GAN-generated images enhancing model robustness and accuracy.

4o mini

2.3.1.6 Research work of base paper:

(Ashraf et al. 2024) [6] The paper "HFF-Net: A Hybrid Convolutional Neural Network for Diabetic Retinopathy Screening and Grading" introduces an efficient CNN-based model that addresses key challenges in diabetic retinopathy (DR) detection, such as vanishing gradients, class imbalance, and overfitting. The model uses the Swish activation function for improved gradient flow and learning stability, and residual blocks to allow deeper network training. Preprocessing steps include CLAHE enhancement and region of interest extraction, with data augmentation (rotation, flipping, shearing, etc.) used to balance the dataset. The proposed HFF-Net architecture extracts and fuses multiscale features at various depths, and achieves 98.70% accuracy for DR screening and 73.77% for grading, outperforming comparable CNNs with a lightweight design of only 1.18 million parameters. The model was evaluated on the APTOS 2019 and Messidor-200 datasets using techniques such as SCL loss function, transfer learning with Xception, and visualization tools for interpretability.

2.4 Literature Review Summary table

Table 1 Summary table

Paper No.	Title	Dataset Used	Algorithm/Model	Best Accuracy	5-Class Accuracy
1	Dual-Stream CNN (EfficientNetB4 + MobileNet)	APTOS, EyePACS	Ensemble CNN	96.25%	96.25% (APTOS)
2	MobileNetV2 + Ensemble CNN	APTOS, Messidor, EyePACS	Lightweight Ensemble CNN	94.37%	94.37% (APTOS)
3	EfficientNetB4 + Hybrid Classifier	APTOS 2019	Hybrid EfficientNetB4	95.58%	95.58% (APTOS)
4	DenseNet121 + ECA Module	Kaggle, Messidor	DenseNet + Attention	97.2%	97.2% (APTOS)
5	ResNet18 with Swish Activation	APTOS, SN Fundus (real-time)	ResNet18 + Swish	93.51%	93.51% (APTOS), 75% (SN)

2.5 Research Gap

Despite the advancements, several gaps persist in DR detection research:

- Difficulty in handling highly imbalanced datasets.
- Limited real-time applications due to high computational costs.
- Challenges in generalizing models across diverse datasets. 98.50%
- Lack of robust systems for early detection in resource-constrained settings.

2.6 Problem Statement

Manual diagnosis of diabetic retinopathy is inefficient, error-prone, and lacks scalability. Automated CNN-based systems address these challenges but face issues such as imbalanced datasets, computational limitations, and generalizability. This project aims to develop an efficient, accurate, and scalable CNN-based solution for DR detection, bridging these gaps and improving accessibility in diverse healthcare environments.

Chapter 3:

Requirements and Design

Chapter 3: Requirements and Design

3.1 Requirements

3.1.1 Functional Requirements

3.1.1.1 FR-01: Admin

Table 2 Admin FR

ID	Requirement
FR-1.1	Admin can login into account.
FR-1.2	Admin can be able to add patients…
FR-1.3	Admin can be able to add doctors.
FR-1.4	Admin can view doctors.
FR-1.5	Admin can delete doctors
FR-1.6	Admin can be able to not approve of doctor.
FR-1.7	Admin can be able to logout

3.1.1.2 FR-02: System

Table 3 System FR

ID	Requirement
FR-2.1	System will analyze the image.
FR-2.2	System can be able to provide the results of the disease
FR-2.3	The system can be able to show the accuracy of the disease.

3.1.1.3 FR-03: Patient

Table 4 Patient FR

ID	Requirement
FR-3.1	Patients be able to sign up for the account.
FR-3.2	Patients can be able to log into account
FR-3.3	Patients be able to search for an available doctor.
FR-3.4	Patients can detect Diabetic Retinopathy by uploading his/her retina picture.
FR-3.5	Patients can be able to view result.
FR-3.6	Patients can chat with doctor.
FR-3.7	Patient can download his report.

3.1.1.4 FR-04: Doctor

Table 5 Doctor FR

ID	Requirement
FR – 1.1	Doctor can sign up for his/her account.
FR – 1.2	Doctors can be able to login his/her account.
FR – 1.3	The doctor can see the report of the patient.
FR – 1.4	Doctors can chat with patient.

3.1.2 Non-Functional Requirements

- The deep learning model shall utilize datasets like Aptos and IDRiD to achieve high accuracy in DR detection.
- The system shall maintain a response time under 5 seconds for image analysis.
- MongoDB shall be used to ensure efficient and secure storage of patient and image data.
- The web interface shall be intuitive and user-friendly, enabling easy upload and result retrieval.

3.2 Hardware and Software Requirements

Training our model required high-end systems for smooth processing.

3.2.1 Hardware requirements:

- GPU: Our group uses the Colab Pro (GPU) that is given by the colab for fast training of the model.
- We can take pictures from the fundoscopy camera and then upload this picture to our system and get results

3.2.2 Software Requirements:

- The complete system will be built by using the following tech stack:
- Programming Language: Python (for model training and backend APIs).
- Frameworks: TensorFlow/Keras for deep learning model development.
- Libraries: OpenCV for image processing; Scikit-learn for evaluation metrics.
- Web Technologies: React, CSS for frontend development, Flask/Python for backend integration.
- Database: MongoDB for storing patient data and results.

- Development Environment: Google Colab for model training and experimentation.

3.3 Proposed Methodology:

The Diabetic Retinopathy Detection System is a machine learning-based solution that uses a trained deep learning model to identify the presence and stage of diabetic retinopathy from retinal images. The system is accessible via a web application, making it user-friendly and scalable.

3.4 System Architecture:

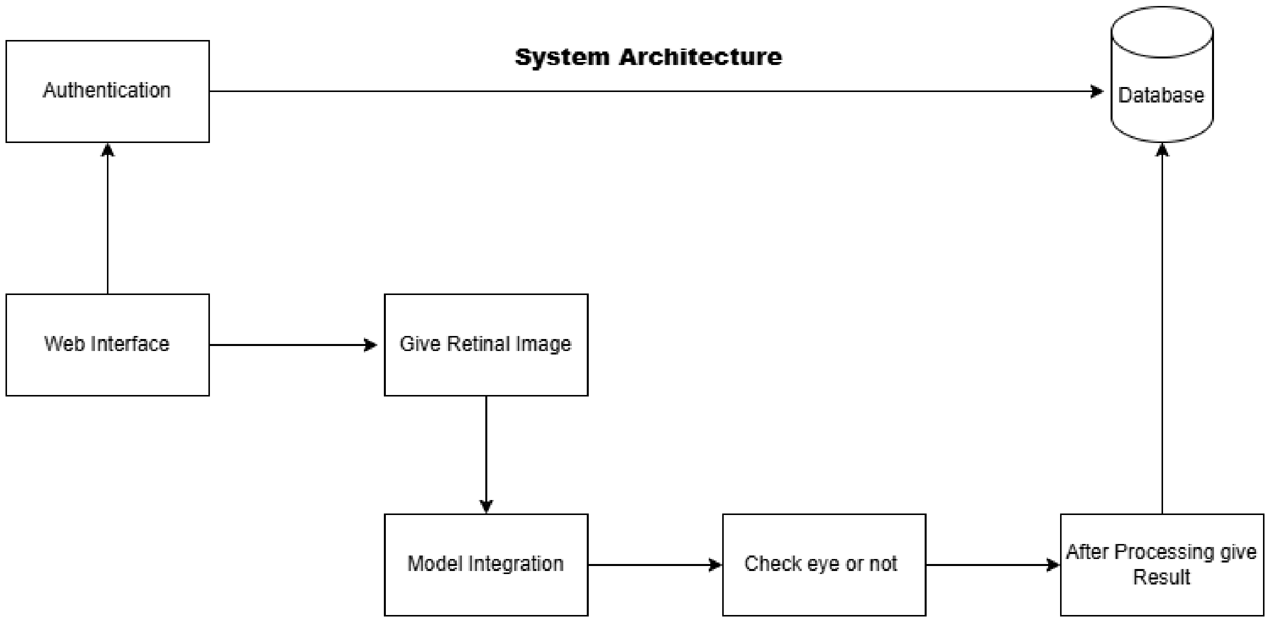


Figure 1 System Architecture Diagram

3.5 Use Cases

3.5.1 Admin Use case:

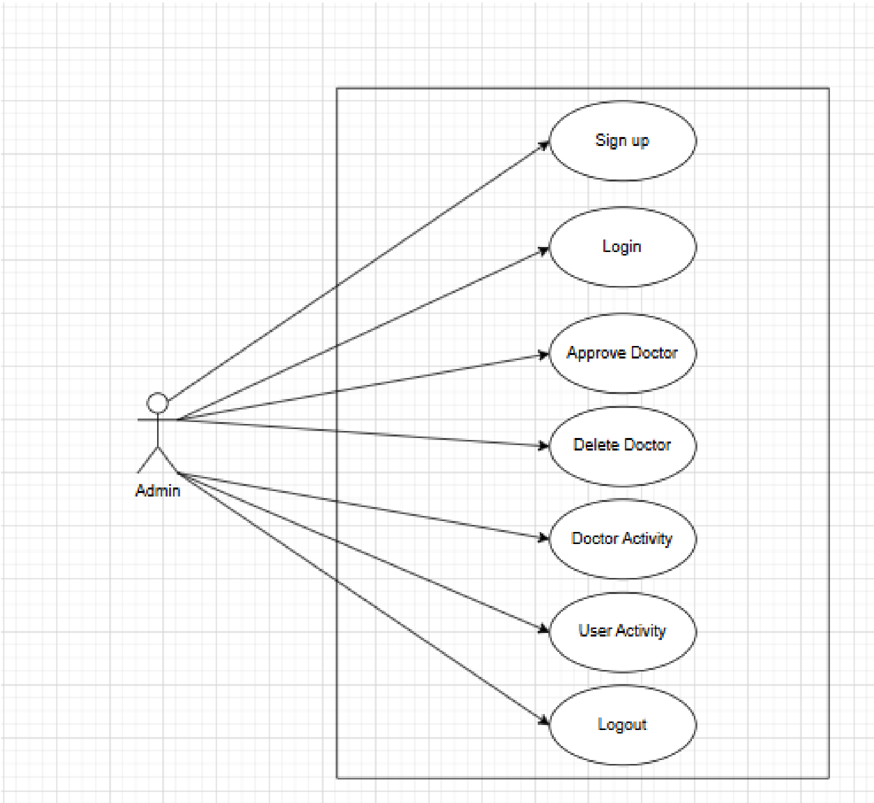


Figure 2 Admin Use case

3.5.2 Patient Use Case:

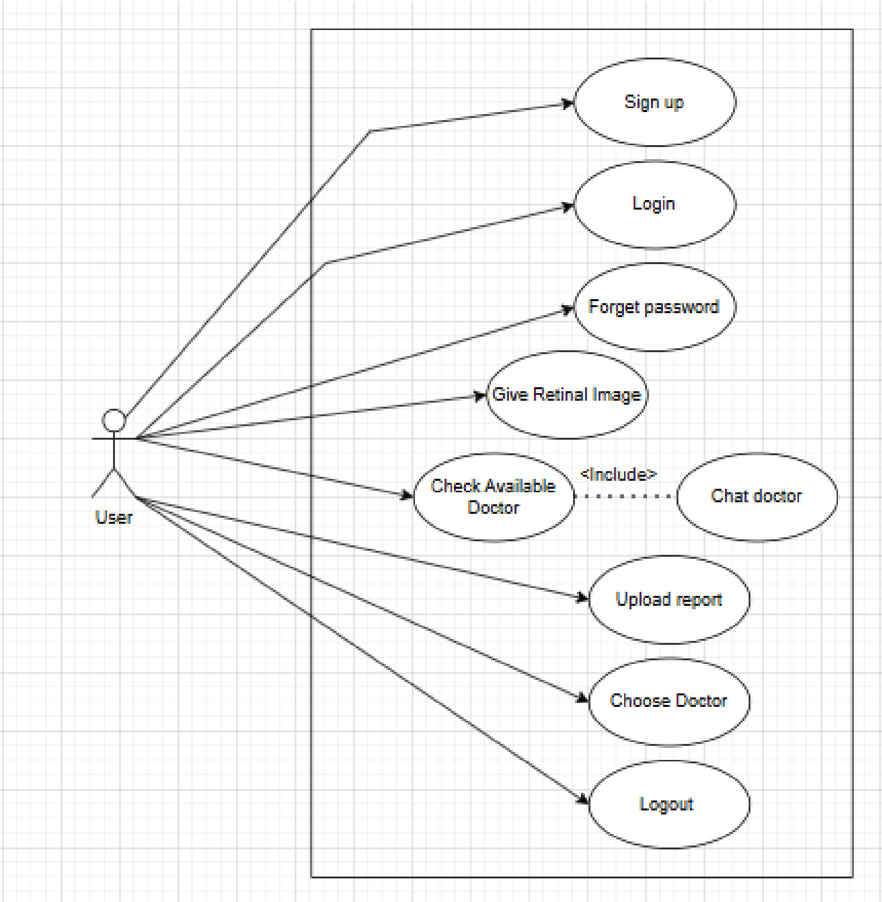


Figure 3 Patient Use Case

3.5.3 Doctor Use Case:

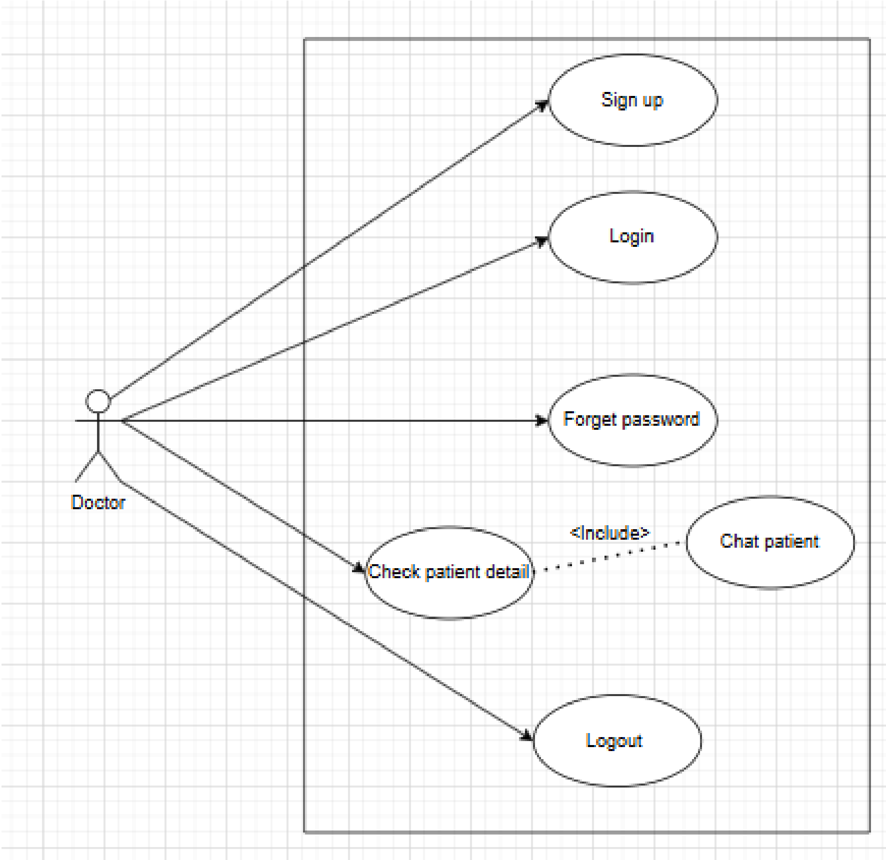


Figure 4 Doctor Use Case

3.6 Full Dressed Use cases:

3.6.1 Admin:

3.6.1.1 Admin login:

Table 6 Admin Login Use case

<table>
<tr><td colspan="2">Name</td><td colspan="2">Admin Login</td></tr>
<tr><td colspan="2">Actors</td><td colspan="2">Admin</td></tr>
<tr><td colspan="2">Summary</td><td colspan="2">Shows the step and interaction involved when an administrator logs into the system.</td></tr>
<tr><td colspan="2">Pre-Conditions</td><td colspan="2">The system is running.</td></tr>
<tr><td colspan="2">Post-Conditions</td><td colspan="2">Admin gets access to the system</td></tr>
<tr><td colspan="2">Special Requirements</td><td colspan="2">None</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>The admin clicks on the "Login" button.</td><td>2</td><td>The admin goes to the admin dashboard.</td></tr>
<tr><td>3</td><td>The admin enters a valid registered email.</td><td>4</td><td>The admin gets access to the system.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>3</td><td>The admin enters an unregistered email.</td><td>4-A</td><td>The system displays: "Email not found."</td></tr>
</table>

3.6.1.2 Approve Doctor:

Table 7 Approve Doctor Use case

<table>
<tr><td>Name</td><td colspan="3">Approve Doctor</td></tr>
<tr><td>Actors</td><td colspan="3">Admin</td></tr>
<tr><td>Summary</td><td colspan="3">Admin approves newly registered doctors before they can access the system.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">Doctors must have signed up and are pending approval.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">The selected doctor’s status is updated to “Approved.”.</td></tr>
<tr><td>Special Requirements</td><td colspan="3">None</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>The admin navigates to the list of pending doctors.</td><td>2</td><td>The system displays doctors awaiting approval.</td></tr>
<tr><td>3</td><td>The admin clicks “Approve” on a doctor.</td><td>4</td><td>The system updates the doctor’s status to approve.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>3-A</td><td>The admin attempts to approve of a doctor who doesn’t exist.</td><td>4-A</td><td>The system displays: "Doctor not found or already approved."</td></tr>
</table>

3.6.1.3 Delete Doctor:

Table 8 Delete Doctor Use case

<table>
<tr><td>Name</td><td colspan="3">Delete Doctor</td></tr>
<tr><td>Actors</td><td colspan="3">Admin</td></tr>
<tr><td>Summary</td><td colspan="3">Admin deletes a doctor’s account from the system.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">The doctor must exist in the system.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">The selected doctor’s data is permanently removed or deactivated.</td></tr>
<tr><td>Special Requirements</td><td colspan="3">None</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>The admin navigates to the doctor management section.</td><td>2</td><td>The system displays a list of registered doctors.</td></tr>
<tr><td>3</td><td>The admin selects a doctor and clicks “Delete.”</td><td>4</td><td>The system removes the doctor’s data.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>3-A</td><td>The admin selects a non-existent doctor.</td><td>4-A</td><td>The system displays: "Doctor record not found."</td></tr>
</table>

3.6.1.4 Doctor Activity:

Table 9 Doctor activity Use case

<table>
<tr><td>Name</td><td colspan="3">Doctoral Activity</td></tr>
<tr><td>Actors</td><td colspan="3">Admin</td></tr>
<tr><td>Summary</td><td colspan="3">Admin views a log of doctor-related activities like appointments, edits, or logins.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">Doctors must have interacted with the system</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">Activity log is shown.</td></tr>
<tr><td>Special Requirements</td><td colspan="3">None</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>The admin navigates to the “Doctor Activity” page.</td><td>2</td><td>The system fetches and displays activity logs.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>3-A</td><td>No activity is recorded.</td><td>4-A</td><td>The system shows: "No activity available."</td></tr>
</table>

3.6.1.5 Logout:

Table 10 Admin Logout Use case

Name	Logout		
Actors	Admin		
Summary	Admin ends their current session.		
Pre-Conditions	Admin must be logged in.		
Post-Conditions	Session is terminated and the user is redirected to the login page.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The admin clicks the “Logout” button.	2	The system ends the session and redirects to the login screen.
Alternative Flow			
3-A	None	4-A	None

3.6.2 Doctor:

3.6.2.1 Doctor sign up

Table 11 Doctor Signup Use case

<table>
<tr><td>Name</td><td colspan="3">Sign up</td></tr>
<tr><td>Actors</td><td colspan="3">Doctor</td></tr>
<tr><td>Summary</td><td colspan="3">A doctor creates an account by providing registration details.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">The doctor must not already be registered.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">A registration request is submitted and awaits admin approval.</td></tr>
<tr><td>Special Requirements</td><td colspan="3">None</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>The doctor opens the sign-up page.</td><td>2</td><td>The system displays a form for registration.</td></tr>
<tr><td>3</td><td>The doctor fills in all required information.</td><td>4</td><td>The system stores the data and notifies admin for approval.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>3-A</td><td>Doctor submits incomplete or invalid data.</td><td>4-A</td><td>System shows: "Please fill all required fields correctly."</td></tr>
</table>

3.6.2.2 Login

Table 12 Doctor Login Use case

Name	Login		
Actors	Doctor		
Summary	A doctor logs into the system using valid credentials.		
Pre-Conditions	Doctor must be approved by the admin and not already logged in.		
Post-Conditions	A session is created, and the doctor is redirected to the dashboard.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	Doctor opens the Login page	2	System displays Login form
3	Doctor fills valid credentials.	4	The system authenticates and logs in to the doctor.
Alternative Flow			
3-A	Doctor enters invalid credentials.	4-A	System displays: **"Incorrect email or password."**

3.6.2.3 Forget Password:

Table 13 Forget Password Use case

Name	Forget Password		
Actors	Doctor		
Summary	Allow the doctor to reset their password if forgotten.		
Pre-Conditions	Doctor must be registered in the system.		
Post-Conditions	A password reset email/link is sent.		
Special Requirements	Email must match existing records.		
Basic Flow			
Actor Action		**System Response**	
1	Doctor clicks "Forget Password"	2	System asks for registered email.
3	Doctor provides email.	4	System sends reset links to the email.
Alternative Flow			
3-A	The email is not found.	4-A	System displays: **"Email not registered."**

3.6.2.4 Check Patient details:

Table 14 Patient Detail Use case

Name	Check Patient Detail.		
Actors	Doctor		
Summary	Doctor views patient's profile and medical history.		
Pre-Conditions	Doctors must be logged in and have access to the patient list.		
Post-Conditions	A password reset email/link is sent.		
Special Requirements	Data must be securely handled		
Basic Flow			
Actor Action		**System Response**	
1	Doctor selects a patient from the list.	2	The system retrieves and displays patient's details.
Alternative Flow			
3-A	Patient records are missing or inaccessible.	4-A	System shows: **"Patient details not available."**

3.6.2.5 Chat Patient (Include in Check Patient Details

Table 15 Chat Patient Use case

Name	Chat Patient		
Actors	Doctor		
Summary	Enables real time chat with Doctor and Patient.		
Pre-Conditions	The doctor must be viewing the patient’s details.		
Post-Conditions	A secure chat session has been established.		
Special Requirements	Live chat module or integration required.		
Basic Flow			
Actor Action		**System Response**	
1	Doctor clicks “Chat” inside patient detail.	2	The system opens a secure chat window with the patient.
Alternative Flow			
3-A	Patients are offline.	4-A	System shows: **"Patient not available for chat."**

3.6.2.6 Logout:

Table 16 Doctor logout Use case

Name	Logout		
Actors	Doctor		
Summary	Ends the doctor's session in the system.		
Pre-Conditions	Doctor must be logged in.		
Post-Conditions	Session ends and doctor is redirected to the login page.		
Special Requirements	None.		
Basic Flow			
Actor Action		**System Response**	
1	Doctor clicks on "Logout."	2	The system logs out the doctor and redirects to login.
Alternative Flow			
3-A	None	4-A	None

3.6.3 User/Patient:

3.6.3.1 User/Patient sign up:

Table 17 User Signup

Actors	Users / Patient		
Summary	A user creates an account by providing registration details.		
Pre-Conditions	Users must not be registered.		
Post-Conditions	Accounts are created and saved in the system.		
Special Requirements	Valid information must be entered.		
Basic Flow			
Actor Action		**System Response**	
1	User opens the sign-up page.	2	The system displays the sign-up form.
Alternative Flow			
3-A	User submits incomplete or invalid info.	4-A	System shows: **"Please provide valid information."**

3.6.3.2 Login

Table 18 User Login

Name	Login		
Actors	Users		
Summary	A user logs into the system using valid credentials.		
Pre-Conditions	User must be registered		
Post-Conditions	User session is established and redirected to dashboard.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	The user opens the login page.	2	The system displays login form.
3	User enters credentials.	4	The system authenticates and logs in to the user.
Alternative Flow			
3-A	Credentials are invalid.	4-A	System displays: **"Incorrect email or password."**

3.6.3.3 Forget Password:

Table 19 Forget password user Use case

Name	Forget Password		
Actors	Users		
Summary	Allow users to reset passwords if forgotten.		
Pre-Conditions	Email must be registered.		
Post-Conditions	A reset link is sent to the user's email.		
Special Requirements	Email Verification.		
Basic Flow			
Actor Action		**System Response**	
1	User clicks "Forget Password".	2	System asks for registered email.
3	User submits email.	4	System sends a password reset link.
Alternative Flow			
3-A	Email doesn't exist	4-A	System shows: **"Email not registered."**

3.6.3.4 Give Retinal Image:

Table 20 Give eye image

Name	Give Retinal Image		
Actors	User		
Summary	Users upload a retinal image to be analyzed.		
Pre-Conditions	User must be logged in.		
Post-Conditions	Retinal image is uploaded for analysis		
Special Requirements	Image must be in supported format.		
Basic Flow			
Actor Action		**System Response**	
1	User clicks "Give Retinal Image".	2	System prompts to upload image.
3	User uploads retinal image.	4	System stores the image and begins analysis.
Alternative Flow			
3-A	Image format is unsupported	4-A	System shows: **"Invalid image format."**

3.6.3.5 Check Available Doctor:

Table 21 Available Doctor Use case

<table>
<tr><td>Name</td><td colspan="3">Check Available Doctor</td></tr>
<tr><td>Actors</td><td colspan="3">User</td></tr>
<tr><td>Summary</td><td colspan="3">Shows list of doctors available for consultation.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">User must be logged in.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">The system displays a list of available doctors.</td></tr>
<tr><td>Special Requirements</td><td colspan="3">None</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>User selects “Check Available Doctor”.</td><td>2</td><td>The system displays real-time lists of available doctors.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>3-A</td><td>No Doctors available.</td><td>4-A</td><td>System shows: "No Doctors currently available."</td></tr>
</table>

3.6.3.6 Chat Doctor:

Table 22 Chat Doctor Use Case

Name	Chat Doctor		
Actors	User		
Summary	User chats with a doctor in real-time		
Pre-Conditions	Doctors must be available and selected.		
Post-Conditions	Secure chat is established between user and doctor.		
Special Requirements	None		
Basic Flow			
Actor Action		**System Response**	
1	User selects an available doctor and clicks "Chat".	2	System opens chat window with the doctor
Alternative Flow			
3-A	No Doctors available.	4-A	System shows: **"No Doctors currently available."**

3.6.3.7 Upload report:

Table 23 Upload Report Use case

<table>
<tr><td>Name</td><td colspan="4">Upload Report</td></tr>
<tr><td>Actors</td><td colspan="4">User</td></tr>
<tr><td>Summary</td><td colspan="4">Users upload medical reports for doctor review.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="4">User must be logged in.</td></tr>
<tr><td>Post-Conditions</td><td colspan="4">Reports are uploaded and stored securely.</td></tr>
<tr><td>Special Requirements</td><td colspan="4">Allowed file types (PDF, images).</td></tr>
<tr><td colspan="5">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="3">System Response</td></tr>
<tr><td>1</td><td>User selects “Upload Report”.</td><td>2</td><td colspan="2">System prompts to choose a file.</td></tr>
<tr><td>3</td><td>User uploads report.</td><td>4</td><td colspan="2">The system confirms upload success.</td></tr>
<tr><td colspan="5">Alternative Flow</td></tr>
<tr><td>3-A</td><td>File is not accepted format.</td><td>4-A</td><td colspan="2">System shows: "Invalid report file."</td></tr>
</table>

3.6.3.8 Choose doctor:

Table 24 Choose Doctor Use case

<table>
<tr><td>Name</td><td colspan="3">Choose</td></tr>
<tr><td>Actors</td><td colspan="3">User</td></tr>
<tr><td>Summary</td><td colspan="3">Users select a doctor for treatment or follow-up.</td></tr>
<tr><td>Pre-Conditions</td><td colspan="3">A list of doctors must be available.</td></tr>
<tr><td>Post-Conditions</td><td colspan="3">Selected doctor is saved in user profile.</td></tr>
<tr><td>Special Requirements</td><td colspan="3">Doctor ID and Rating.</td></tr>
<tr><td colspan="4">Basic Flow</td></tr>
<tr><td colspan="2">Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>1</td><td>User views list of doctors.</td><td>2</td><td>The system displays all available options.</td></tr>
<tr><td>3</td><td>User selects preferred doctor.</td><td>4</td><td>The system links users to the selected doctor.</td></tr>
<tr><td colspan="4">Alternative Flow</td></tr>
<tr><td>3-A</td><td>The doctor is no longer available.</td><td>4-A</td><td>System shows: "Doctor unavailable, please choose another."</td></tr>
</table>

3.6.3.9 Logout:

Table 25 User Logout Use case

Name	Logout		
Actors	User		
Summary	Ends the user's session in the system.		
Pre-Conditions	User must be logged in.		
Post-Conditions	The user is logged out and returned to login screen.		
Special Requirements	None.		
Basic Flow			
Actor Action		**System Response**	
1	User selects "Logout".	2	System prompts to choose a file.
Alternative Flow			
3-A	None	4-A	None

3.7 Sequence diagram:

3.7.1 Login

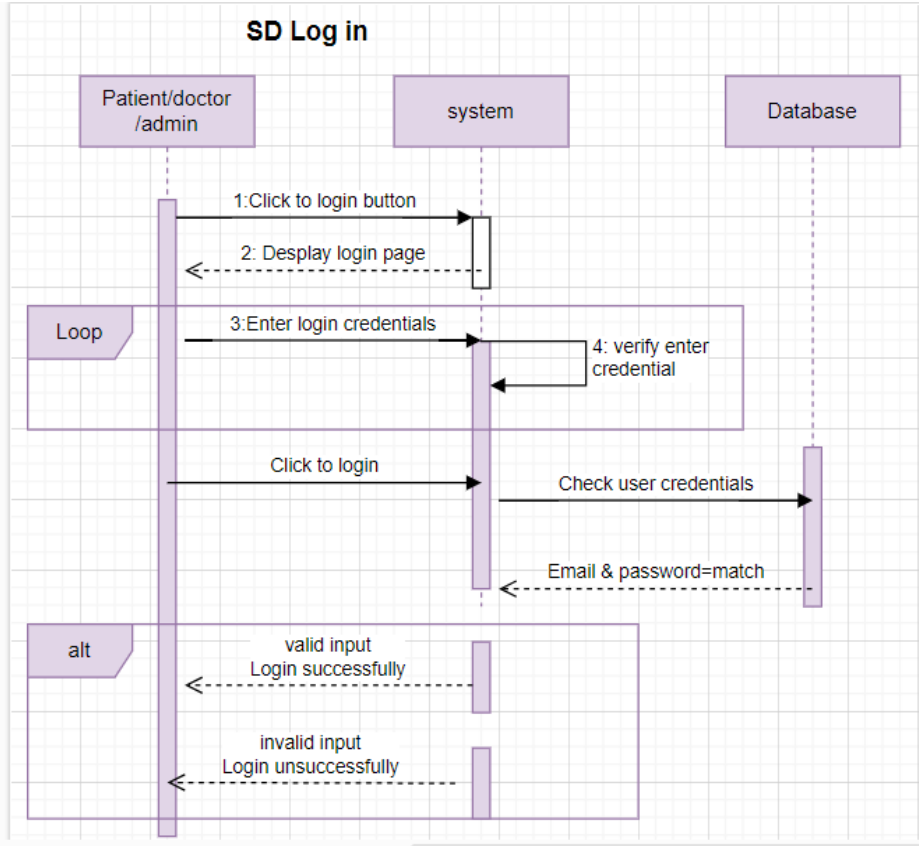


Figure 5 Login SD

3.7.2 Sign up:

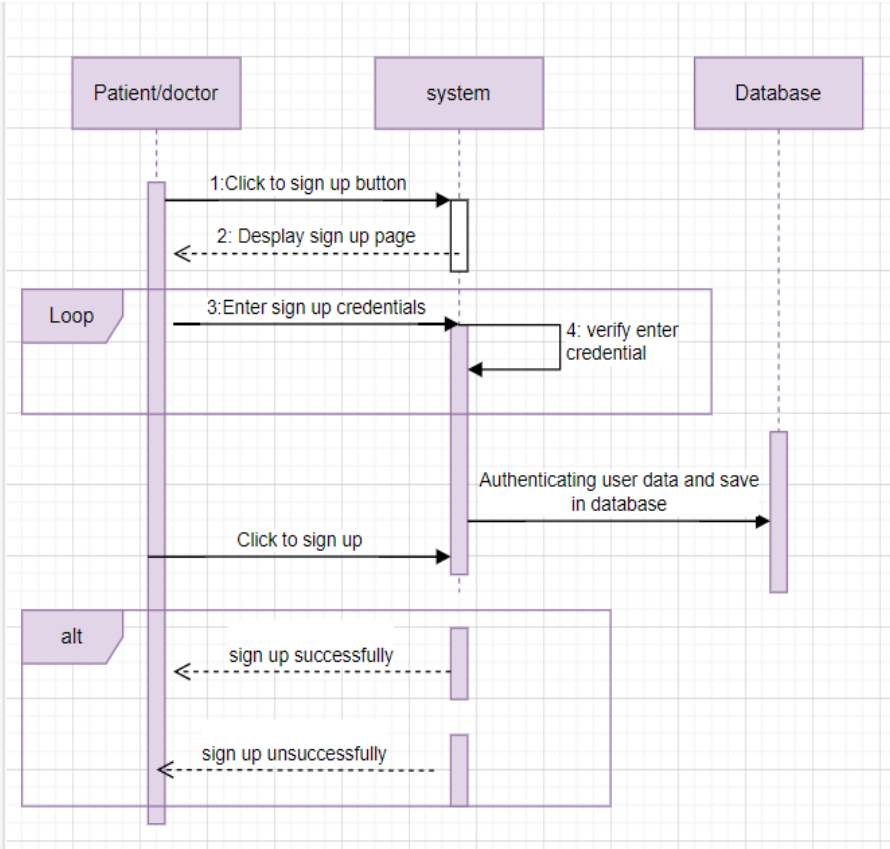


Figure 6 Sign up SD

3.7.3 Doctor Detail:

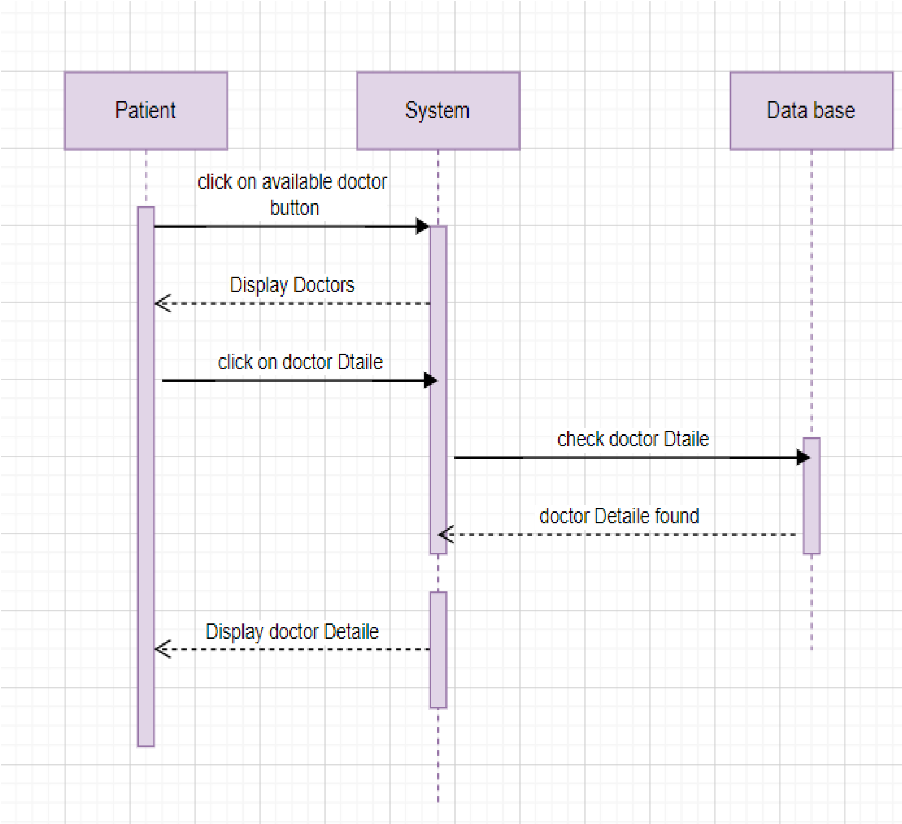


Figure 7 Doctor detail SD

3.7.4 Forget Password:

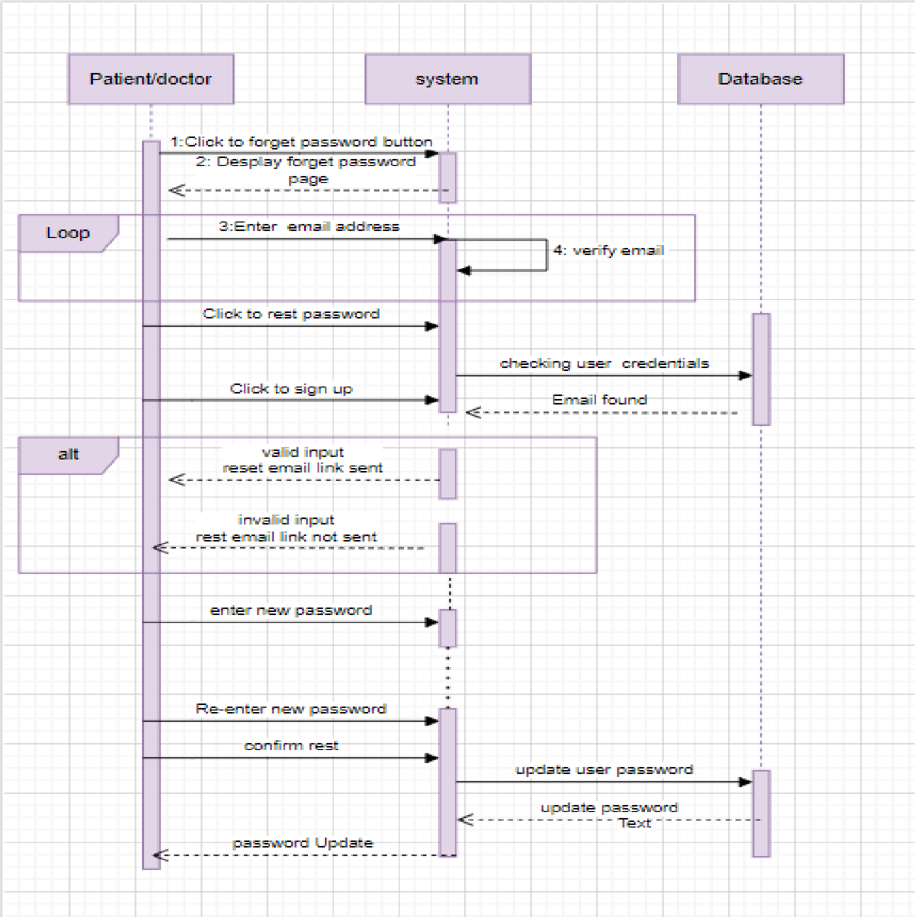


Figure 8 Forget Password SD

3.7.5 Logout:

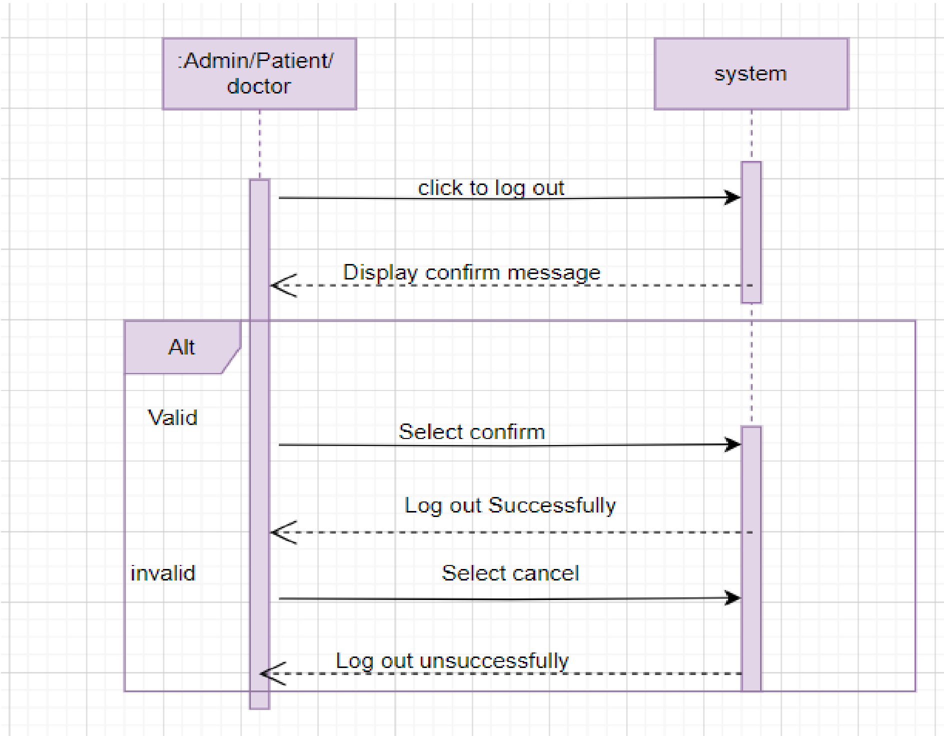


Figure 9 Log out SD

3.8 Flow control:

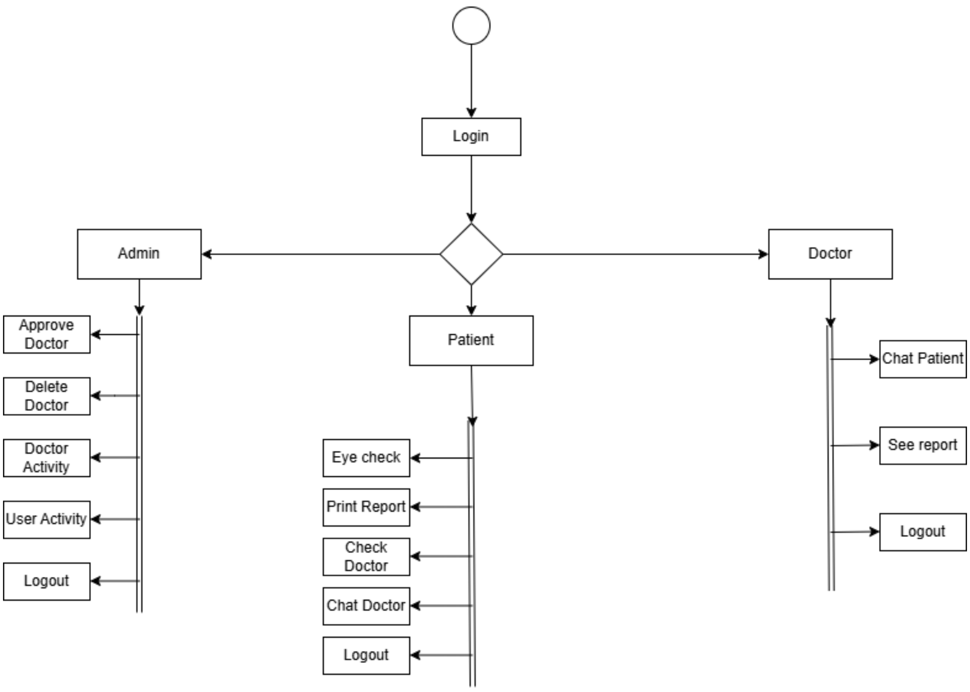
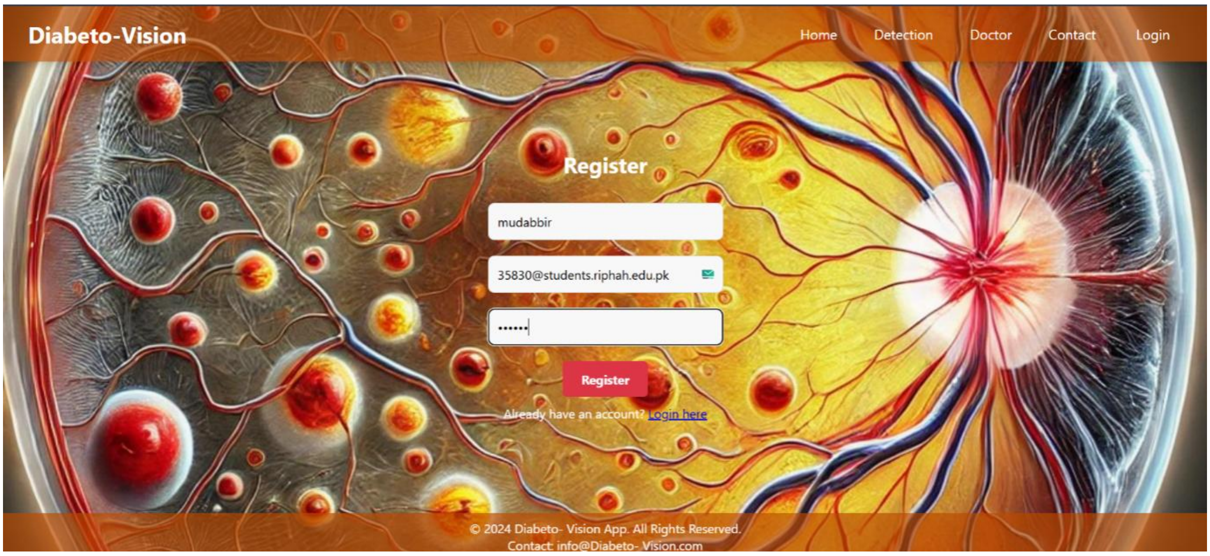


Figure 10 Flow Control

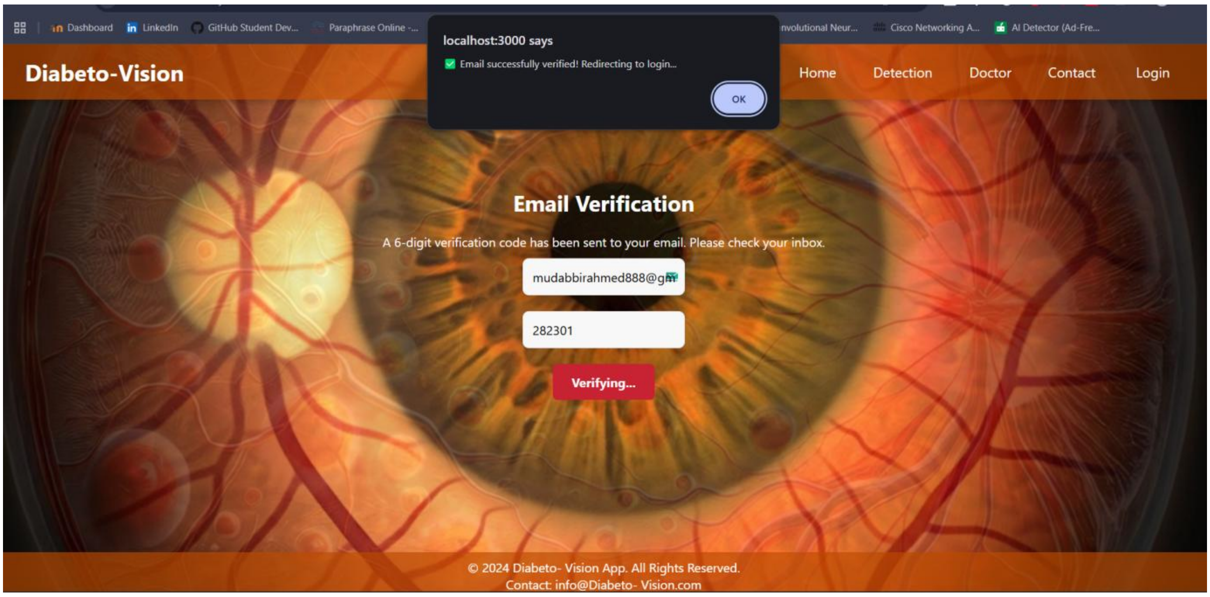
3.8 GUI Graphical User Interfaces

We are adding some screenshots of our web interface (GUI)

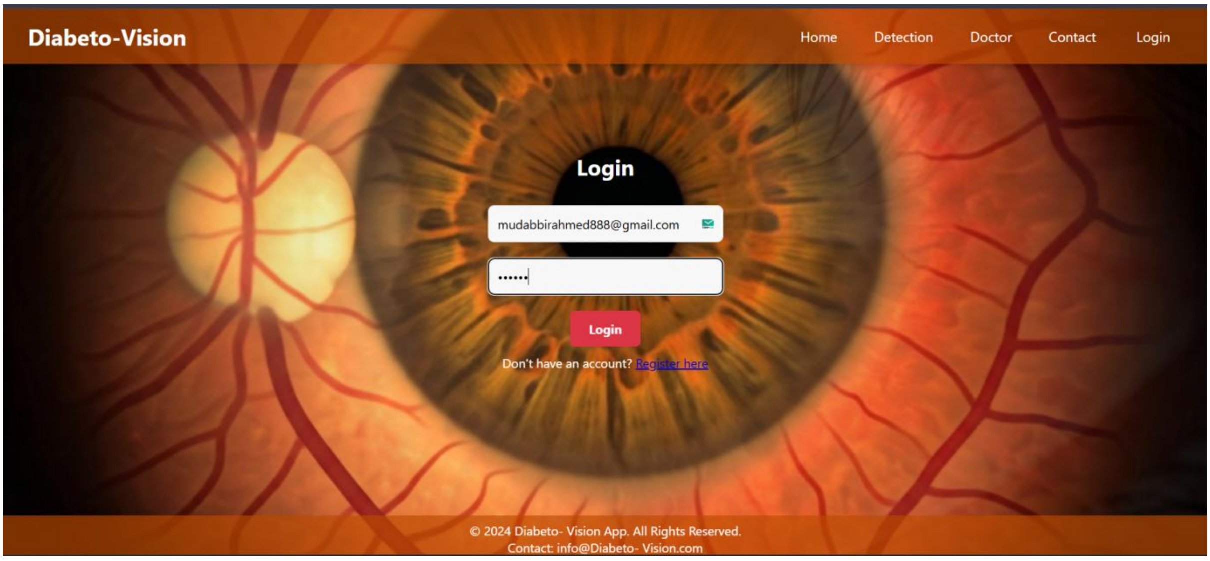
3.8.1 Patient Login



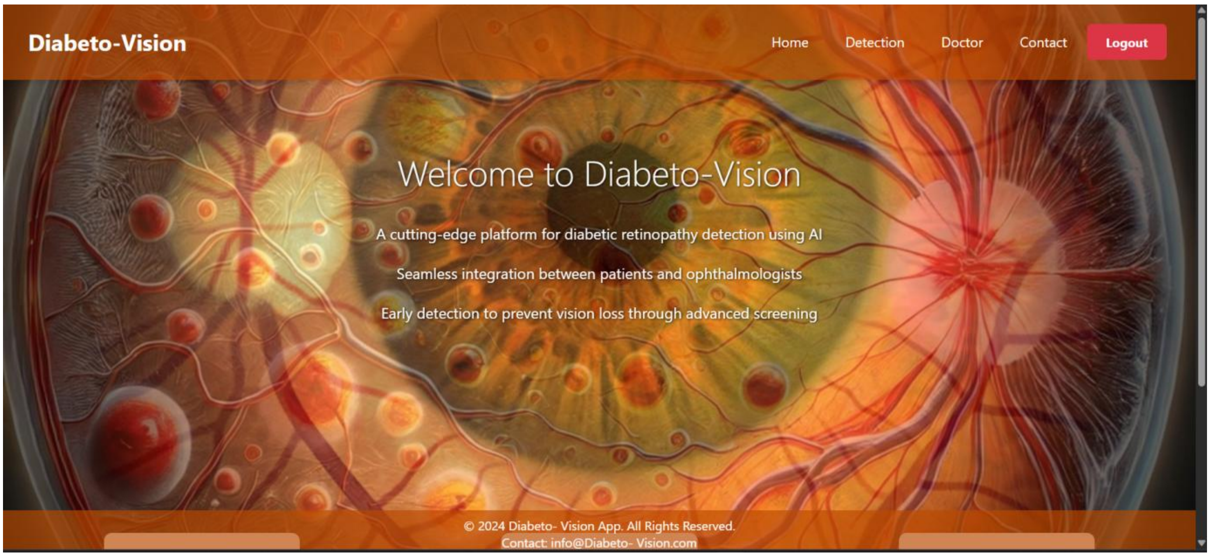
3.8.2 Email Verification



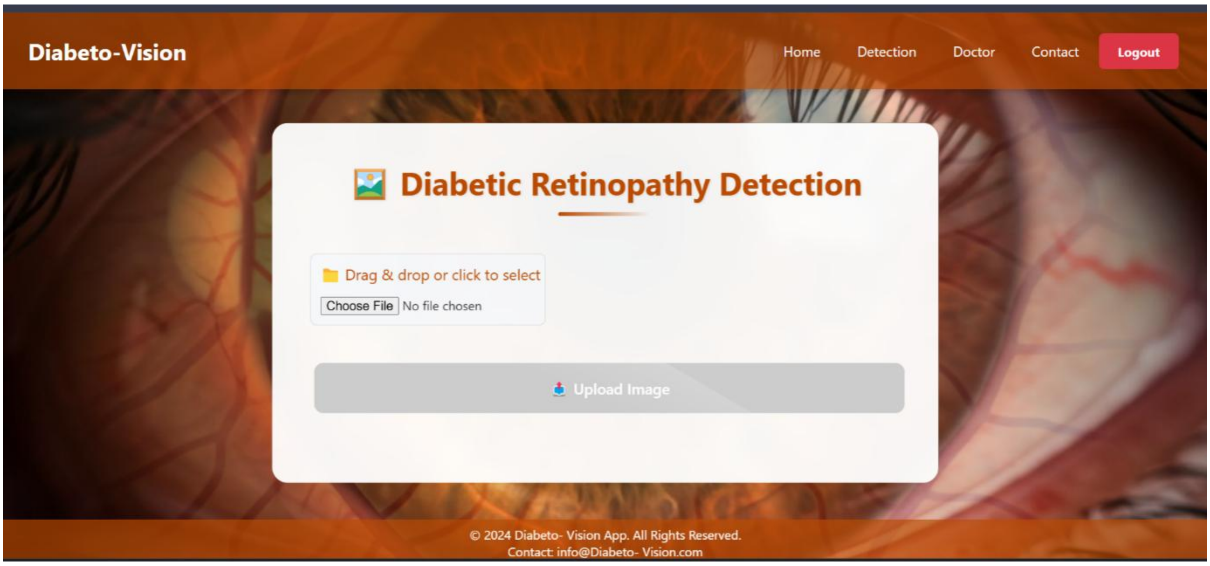
3.8.3 Doctor Login



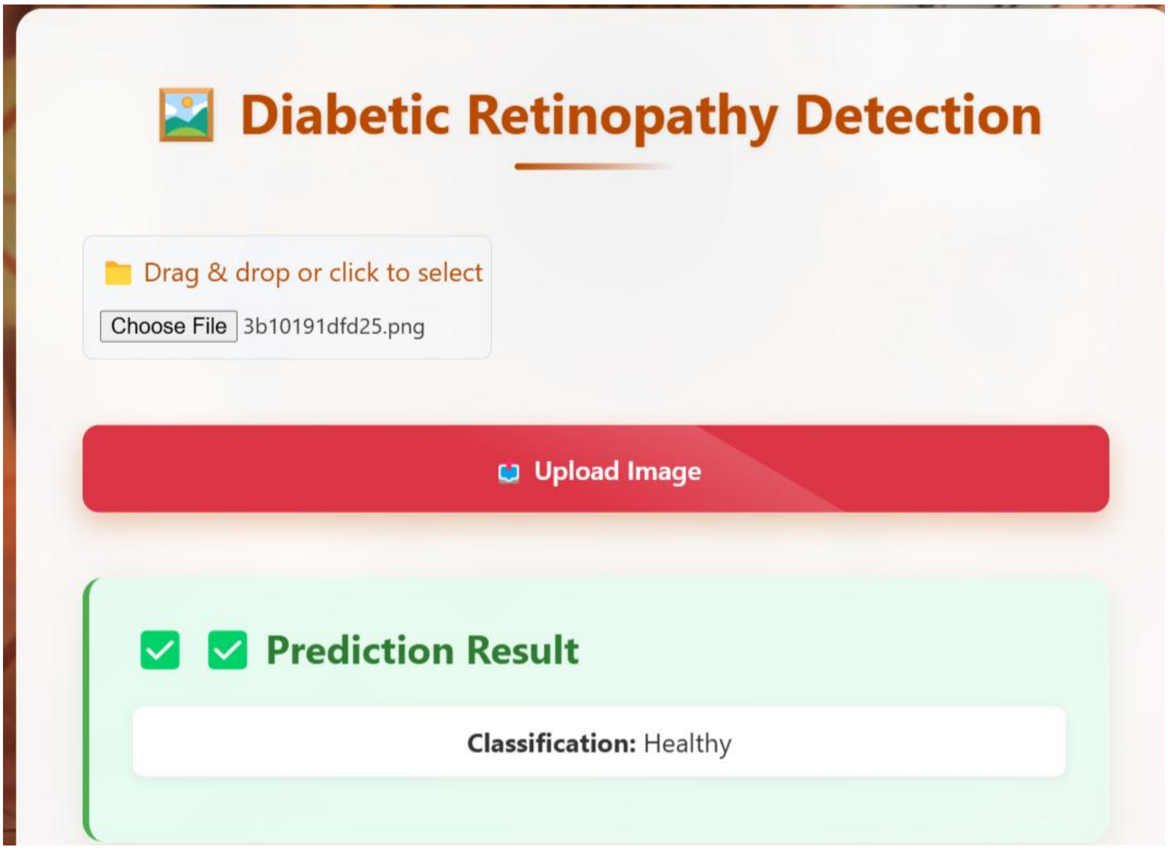
3.8.4 Dashboard



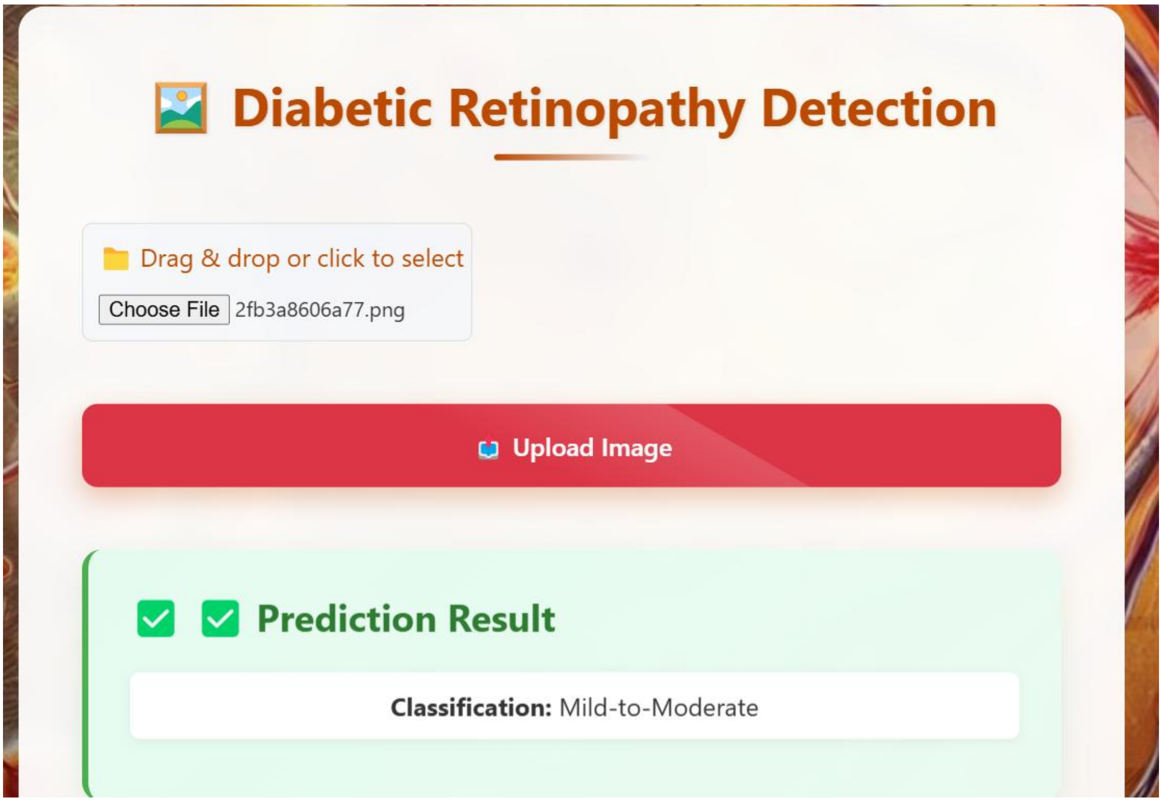
3.8.5 Choose Image from the device



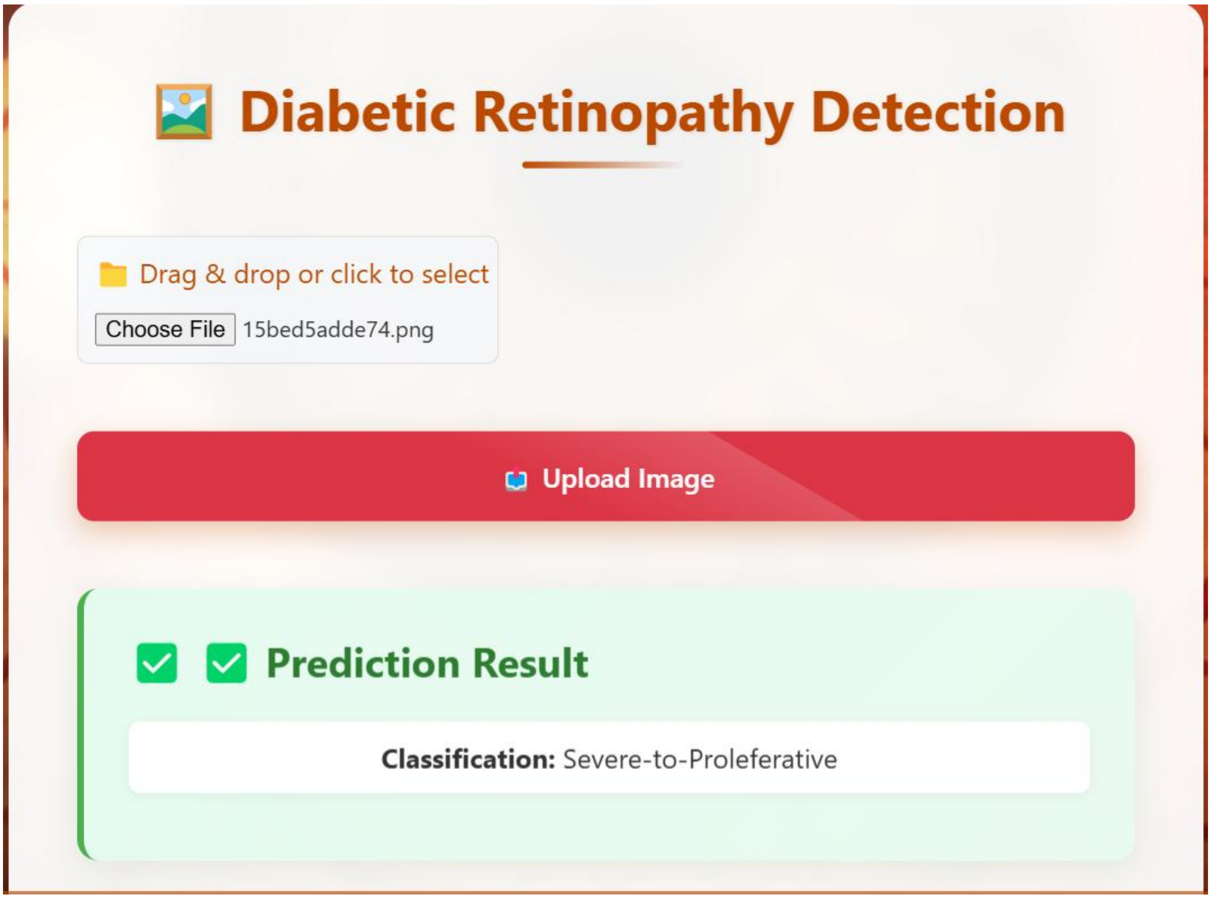
3.8.6 Healthy Result



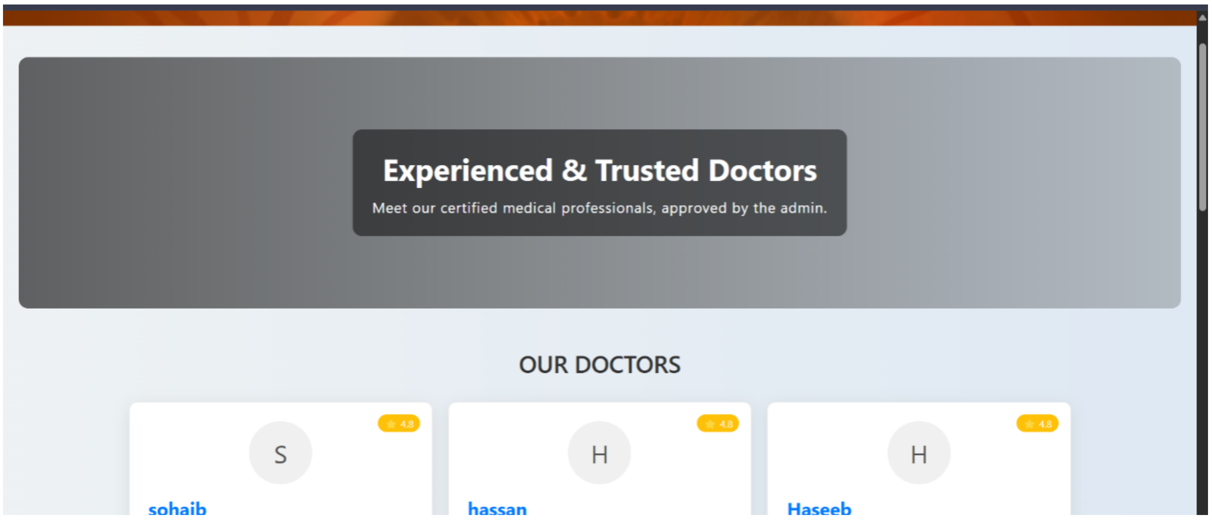
3.8.7 Mild to Moderate Result:



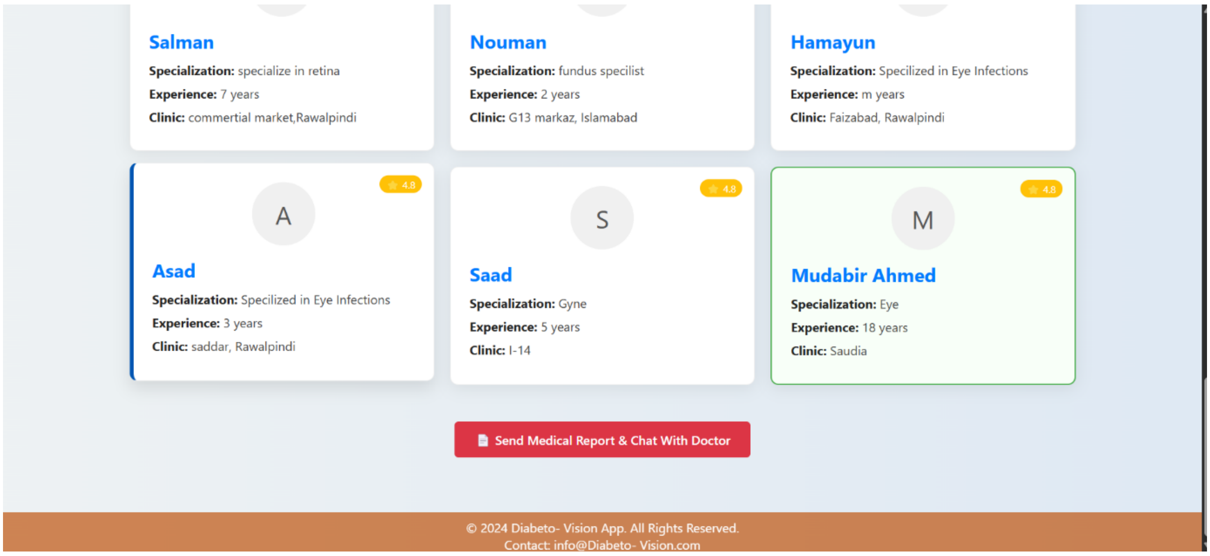
3.8.9 Severe to Proliferative Result



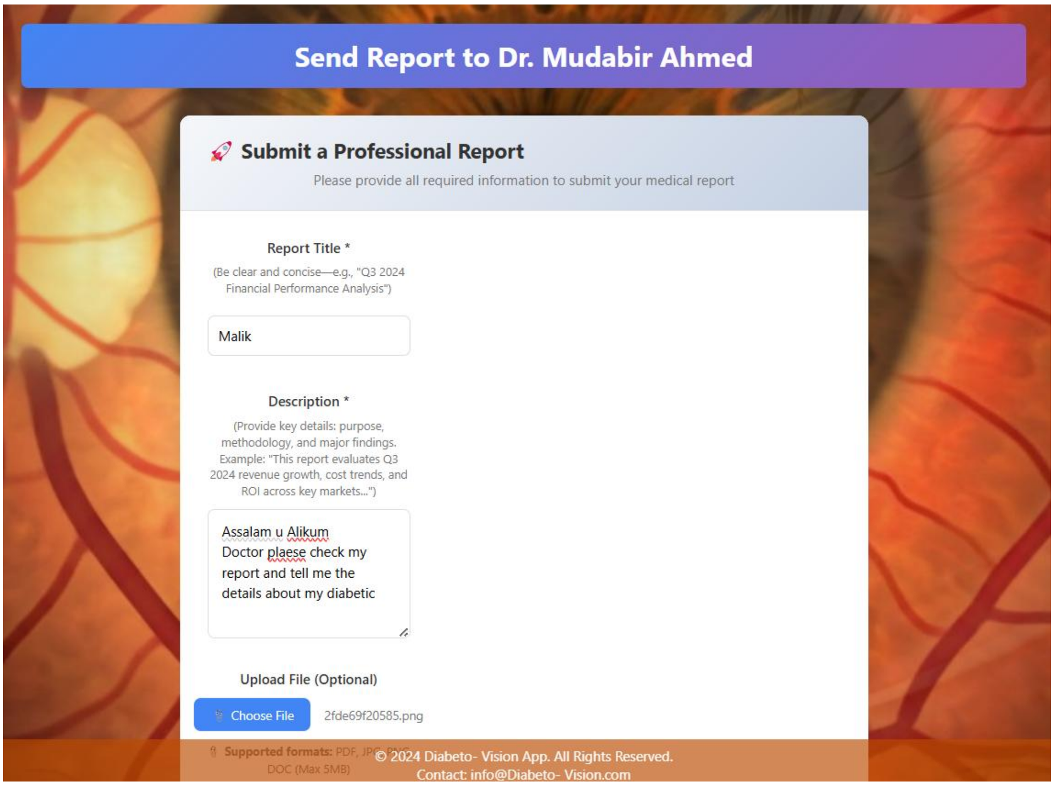
3.8.10 Doctor Profile



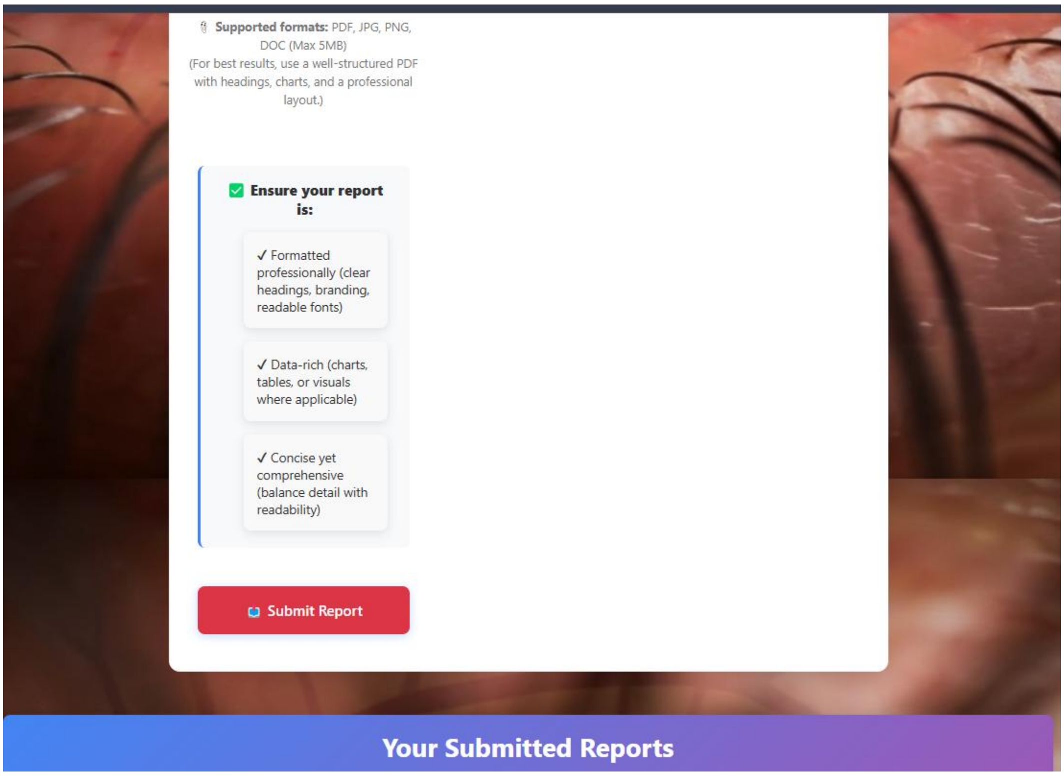
3.8.11 Communicate with doctor:



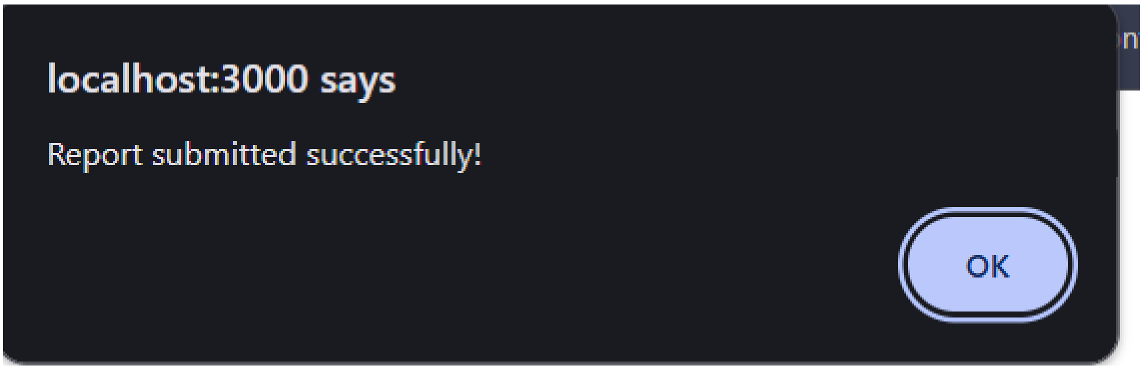
3.8.12 Chat with Doctor:



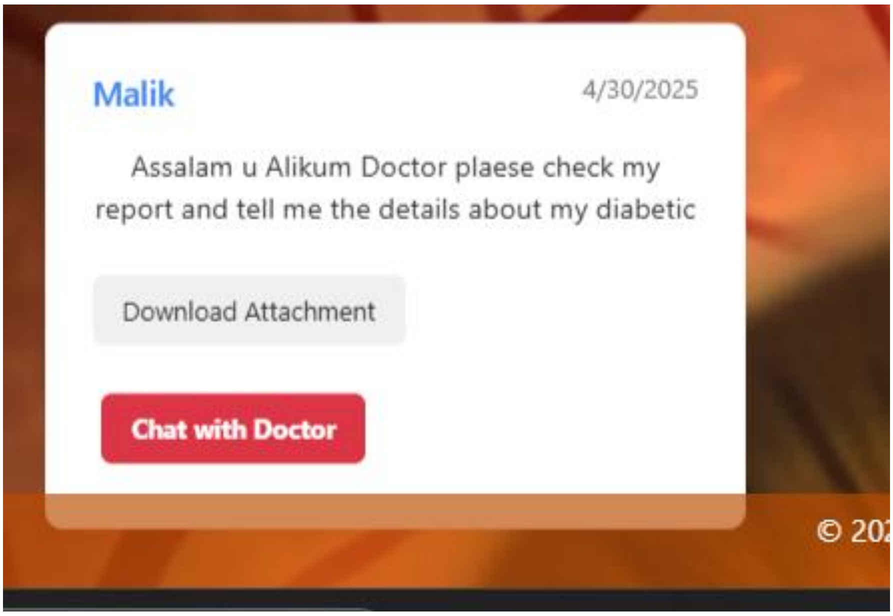
3.8.13 Submission report



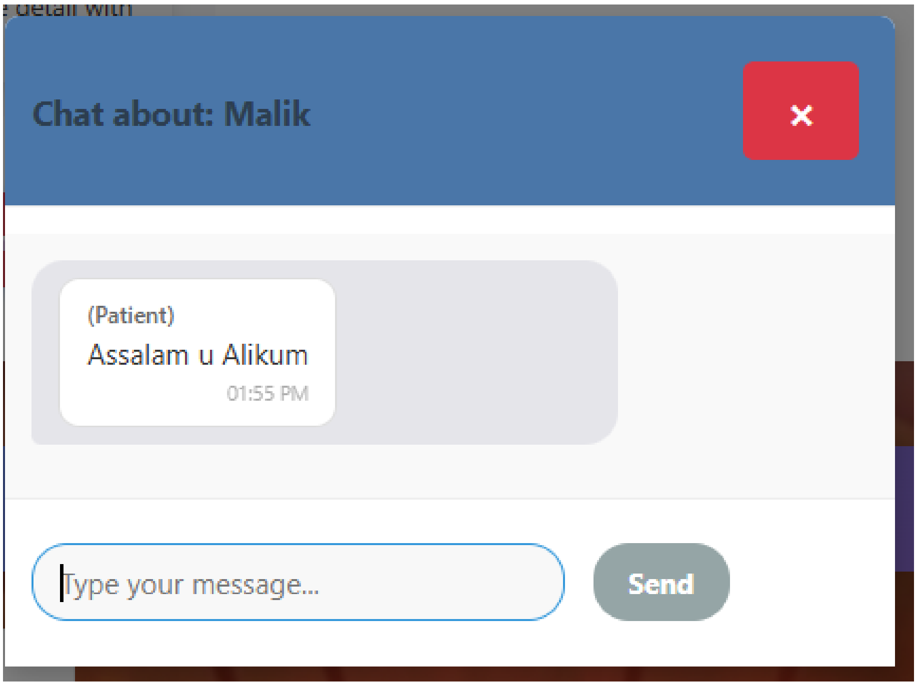
3.8.14 Submitted Successfully:



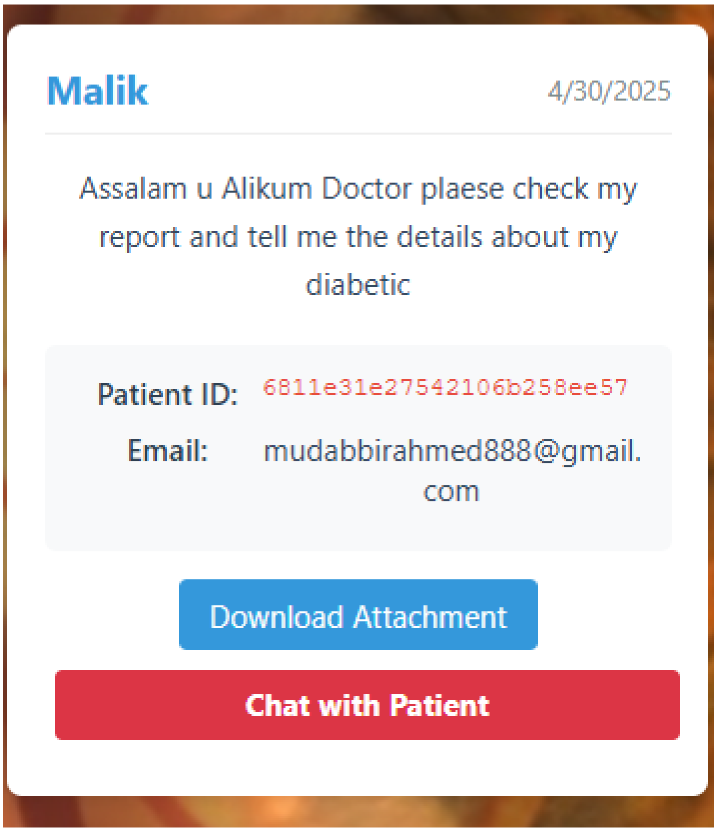
3.8.15 Chat with Patient:



3.8.16 Chat Screenshot



3.8.17 Showing Patient ID:



Chapter 4:
Implementation And Test Cases

Chapter 4: Implementation and Test Cases

4.1 Implementation

This section covers the implementation details of the project's core components, including algorithms used, development environment, tools, and libraries. Python is the primary language due to its versatility and extensive support for machine learning and web development. Key technologies include Flask for the web framework, TensorFlow for machine learning, and MongoDB for database management.

4.1.1 Implementation of First Component/Algorithm

In our system, we implemented a two-stage classification pipeline using deep learning. In the first stage, a model is trained to classify images as either Valid (retina image) or Invalid (another image). Only the valid images are passed to the second stage, where a separate model classifies them as either Healthy or Diabetic. After experimenting with multiple CNN architectures, we finalized a hybrid model combining features from ResNet50V2 and DenseNet169, leveraging their strengths to improve accuracy. This hybrid model extracts robust features from both architectures, concatenates them, and passes them through dense layers to make the final prediction. This two-step approach ensures that only high-quality images are used for disease detection, improving the reliability and performance of our system.

4.2 Test Case and Description

4.2.1 Sample Test case No.1

Table 26 Sample test case1

<table>
<tr><td colspan="4"><Admin Login Module></td></tr>
<tr><td colspan="4"><Reference></td></tr>
<tr><td>Test Case ID:</td><td>TC-01</td><td>Test Date:</td><td>2025-2-19</td></tr>
<tr><td>Test case Version:</td><td>V1.0</td><td>Use Case Reference(s):</td><td>Admin -->Login</td></tr>
<tr><td>Revision History:</td><td colspan="3">Initial Version</td></tr>
<tr><td>Objective</td><td colspan="3">To verify that Admin can successfully login.</td></tr>
<tr><td>Product/Ver/Module:</td><td colspan="3">Diabeto Vision v1.0 – Admin Panel.</td></tr>
<tr><td>Environment:</td><td colspan="3">Mongo DB, Node.js, React.js</td></tr>
<tr><td>Assumptions:</td><td colspan="3">Admin Account already exists in the database.
Admin has network access.</td></tr>
<tr><td>Pre-Requisite:</td><td colspan="3">Admin is on the Login page of the system.</td></tr>
<tr><td>Step No.</td><td>Execution description</td><td colspan="2">Procedure result</td></tr>
<tr><td>1</td><td>Enter valid username or password.</td><td colspan="2">Field accepts input.</td></tr>
<tr><td>2</td><td>Click on “Login” Button.</td><td colspan="2">System processes login.</td></tr>
<tr><td>3</td><td>Redirect to Admin Dashboard</td><td colspan="2">Admin dashboard is loaded successfully.</td></tr>
<tr><td colspan="4">Comments:
• Ensure the system handles login attempts gracefully.
• Failed login should load to another try.</td></tr>
<tr><td colspan="4">■ Passed ☐ Failed ☐ Not Executed</td></tr>
</table>

4.2.2 Sample Test case No.2

Table 27 Sample test case 2

<Doctor Login Module>			
<Doctor Functionalities reference>			
Test Case ID:	*TC-02*	**Test Date:**	*2025-2-19*
Test case Version:	*V1.0*	**Use Case Reference(s):**	*Doctor -->Login*
Revision History:	*Initial Version*		
Objective	*To validate that the doctor can successfully log in with valid credentials and access their dashboard to view patient information.*		
Product/Ver/Module:	*Diabeto Vision v1.0 – Doctor Panel.*		
Environment:	*Mongo DB, Node.js, React.js*		
Assumptions:	*The doctor has already signed up and has a valid account.* *Credentials are stored in system database.*		
Pre-Requisite:	*Doctor is on the Login page of the system.* *Valid login credentials are available.*		
Step No.	**Execution description**	**Procedure result**	
1	*Enter valid username or password.*	*Field accepts input.*	
2	*Click on "Login" Button.*	*Request sent to server.*	
3	*Redirect to doctor Dashboard*	*A dashboard with patient list is displayed.*	
Comments: • Failed login scenarios will be tested separately. • Security of login assumed to be implemented.			
■ *Passed* ☐ *Failed* ☐ *Not Executed*			

4.2.3 Sample Test case No.3

Table 28 Sample test case 3

<table>
<tr><td colspan="4" align="center"><User upload report Module></td></tr>
<tr><td colspan="4" align="center"><User Functionalities reference></td></tr>
<tr><td>Test Case ID:</td><td>TC-03</td><td>Test Date:</td><td>2025-2-19</td></tr>
<tr><td>Test case Version:</td><td>V1.0</td><td>Use Case Reference(s):</td><td>User -->Upload report</td></tr>
<tr><td>Revision History:</td><td colspan="3">Initial Version</td></tr>
<tr><td>Objective</td><td colspan="3">To ensure that a user can successfully upload a medical report (e.g., test results or documents) to the system.</td></tr>
<tr><td>Product/Ver/Module:</td><td colspan="3">Diabeto Vision v1.0 – Doctor Panel.</td></tr>
<tr><td>Environment:</td><td colspan="3">Mongo DB, Node.js, React.js</td></tr>
<tr><td>Assumptions:</td><td colspan="3">Users are already logged into the system.
Report file is in accepted format (PDF, JPG, and PNG).</td></tr>
<tr><td>Pre-Requisite:</td><td colspan="3">User has a digital report file ready to upload.
Upload “Report” page is accessible.</td></tr>
<tr><td>Step No.</td><td>Execution description</td><td colspan="2">Procedure result</td></tr>
<tr><td>1</td><td>Navigate to upload report page.</td><td colspan="2">Page is displayed.</td></tr>
<tr><td>2</td><td>Click “Choose File” and select report.</td><td colspan="2">File path is shown in input.</td></tr>
<tr><td>3</td><td>Click “Upload” button</td><td colspan="2">System uploads and confirm successful upload.</td></tr>
<tr><td colspan="4">Comments:
• No comments.</td></tr>
<tr><td colspan="4" align="center">■ Passed ☐ Failed ☐ Not Executed</td></tr>
</table>

4.3 Test Metrics

Summarize here the common ground of attributes of test case metrics.

4.3.1 Sample Test case Matric.No.1

Table 29 Test case matric 1

Metric:	**Purpose**
Number of Test Cases	12 total test cases have been developed covering Admin, Patient, and Doctor modules shown in the diagram.
Number of Test Cases Passed	10 test cases passed successfully after execution.
Number of Test Cases Failed	2 test cases failed due to incorrect file handling in “Upload Report” and session timeout on “Doctor Chat Patient.”
Test Case Defect Density	(2 failed / 12 executed) × 100 = 16.67%
Test Case Effectiveness	(5 defects detected via test cases / 6 total known defects) × 100 = 83.33%
Traceability Matrix	All test cases are traceable to specific actions in the system diagram: login, approve/delete doctor, patient report upload, chat modules, and logout. Each feature is aligned with its related requirement ensuring 100% traceability.

Chapter 5: Experimental Results and Analysis

Chapter 5: Experimental Results and Analysis

5.1 Introduction

The Diabetic Retinopathy Detection System is developed to facilitate the early and accurate diagnosis of diabetic retinopathy using advanced artificial intelligence techniques. By leveraging hybrid deep learning models, including ResNet50V2 and DenseNet169, this project aims to assist healthcare professionals in making timely decisions, thus improving patient outcomes and reducing preventable vision loss.

5.2 Project Achievements

The system successfully achieved accurate classification through a two-stage approach: first, distinguishing valid retinal images from invalid ones, and secondly, categorizing valid images into Healthy or Diabetic classes. The final hybrid model notably improved classification accuracy compared to individual CNN models, demonstrating the efficacy of combined architectures.

5.3 Experimental Results

5.3.1 ResNet 18

Table 30 ResNet Results

Name	Accuracy	Precision	Recall	F1-Score
ResNet 18	0.8198090692124105	0.8091713591387957	0.8198090692124105	0.8095692975784234

Confusion matrix

Table 31 Confusion Matrix of ResNet 18

377	14	4
29	258	26
9	69	52

5.3.2 ResNet 50:

Table 32 ResNet 50 Results

Name	Accuracy	Precision	Recall	F1-Score
ResNet 50	0.780429594272076	0.7735751742101896	0.7804295942720764	0.7760382525688634

Confusion matrix

Table 33 Confusion Matrix ResNet 50

359	22	14
20	247	46
13	69	48

5.3.3 DenseNet201:

Table 34 DenseNet 201 Results

Name	Accuracy	Precision	Recall	F1-Score
DenseNet201	0.7696897374701671	0.7474062906922766	0.7696897374701671	0.7515401954436917

Confusion Matrix

Table 35 Confusion Matrix DenseNet201

361	22	12
28	257	28
9	94	27

5.3.4 EffecientNetV2:

Table 36 Efficient NetV2

Name	Accuracy	Precision	Recall	F1-Score
EfficientNetV2.	0.7494	0.7882	0.7494	0.6882

Confusion matrix

Table 37 Confusion matrix Efficient NetV2

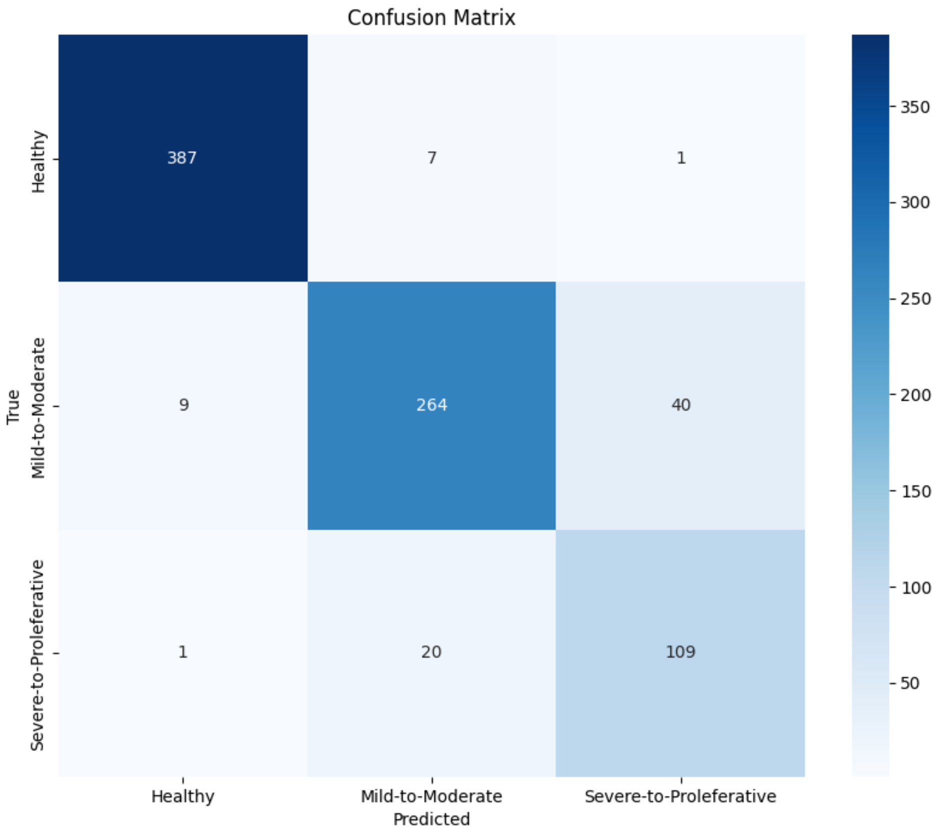
376	5	14
35	158	120
8	38	84

5.4 Alpha net:

Table 38 Alpha Net

	Precision	Recall	F1-score	Support
Healthy	0.97	0.98	0.98	395
Mild-to-Moderate	0.91	0.84	0.87	313
Severe-to-proliferative	0.73	0.84	0.78	130
Accuracy			0.91	838
Macro avg	0.87	0.89	0.88	838
Weighted avg	0.91	0.91	0.91	838

Confusion Matrix:



5.5Critical Analysis:

- **Security:** The system employs secure authentication processes for users, doctors, and administrators, ensuring data confidentiality and integrity. Sensitive patient data is encrypted both in transit and at rest, complying with industry-standard cyber security protocols.
- **Scalability:** Designed for scalability, the system effectively manages increased user traffic and large datasets. Leveraging cloud-based architecture ensures easy resource allocation and minimal downtime during peak usage.
- **Accessibility:** The user interface is intuitive and responsive, supporting diverse devices such as desktops, tablets, and smartphones. This ensures broad accessibility, allowing users and healthcare providers to access the system seamlessly from various platforms.
- **Regulatory Compliance:** The system adheres to healthcare regulations, including data protection standards such as GDPR and HIPAA, ensuring legal compliance and patient privacy protection.

5.6 Conclusion:

This project successfully developed a robust and accurate Diabetic Retinopathy Detection System, addressing critical healthcare needs through advanced AI solutions. While ensuring security, scalability, accessibility, and compliance, the system demonstrates significant potential to enhance healthcare delivery and patient outcomes in diabetic care management.

Chapter 6:

Conclusion and Future Direction:

Chapter 6: Conclusion and Future Directions

6.1 Conclusions

6.1.1 Summary of Work Done

In this project, we designed and implemented an AI-based system to detect diabetic retinopathy from retinal images. Using a hybrid deep learning model that integrates ResNet50V2 and DenseNet169, we developed a two-stage classifier: the first model filters out invalid images, and the second classifies valid images as either Healthy or Diabetic. Additionally, a role-based user interface was developed for admins, doctors, and patients to ensure seamless interaction with the system.

6.1.2 Key Findings and Results

The system achieved high accuracy in distinguishing between healthy and diabetic eyes, and its performance improved significantly through model fusion. We found that image quality plays a vital role in prediction accuracy, justifying the need for an initial validation stage. Moreover, the interface proved user-friendly and responsive across devices, making it accessible to a wide audience.

6.1.3 Scope and Objectives Evaluation

The core objectives were successfully fulfilled. The system can now detect diabetic retinopathy in retinal images, manage user access by roles, and filter out low-quality inputs. However, due to time constraints, some advanced features such as real-time camera integration and multilingual support were not implemented. The initial scope was mostly covered, with a few enhancements reserved for future phases.

6.1.4 Challenges Faced

Some challenges included poor image quality in the dataset, managing imbalanced class distribution, and model over-fitting in early stages. Ensuring privacy and security compliance (GDPR, HIPAA) also required detailed consideration and testing. Integration between frontend and backend modules across different roles was complex and needed multiple iterations.

6.2 Recommendations for Future Work

- Expand the dataset with high-resolution and labeled images to further improve model robustness.
- Add real-time image capture via mobile or webcam integration.
- Introduce multilingual support for better accessibility.
- Include explainable AI techniques to visually highlight areas of concern in the eye.
- Improve session management and two-factor authentication for added security.

References

List all important sources of information which have been consulted for this project

[1] R. Kommaraju and M. S. Anbarasi, “Diabetic retinopathy detection using convolutional neural network with residual blocks,” *Biomed Signal Process Control*, vol. 87, p. 105494, Jan. 2024, doi: 10.1016/J.BSPC.2023.105494.

[2] Y. Modaresnia, F. Abedinzadeh Torghabeh, and S. A. Hosseini, “Enhancing multi-class diabetic retinopathy detection using tuned hyper-parameters and modified deep transfer learning,” *Multimed Tools Appl*, vol. 83, no. 34, pp. 81455–81476, Oct. 2024, doi: 10.1007/S11042-024-18506-3/TABLES/8.

[3] S. P. Singh, P. Gupta, and R. Dung, “Diabetic retinopathy detection by fundus images using fine tuned deep learning model,” *Multimed Tools Appl*, Nov. 2024, doi: 10.1007/S11042-024-19687-7.

[4] S. Aftab, S. Akhtar, S. Aftab, and S. Akhtar, “Diabetic Retinopathy Severity Classification Using Data Fusion and Ensemble Transfer Learning,” *Journal of Software Engineering and Applications*, vol. 18, no. 1, pp. 1–23, Jan. 2025, doi: 10.4236/JSEA.2025.181001.

[5] M. R. Shoaib, H. M. Emara, A. S. Mubarak, O. A. Omer, F. E. Abd El-Samie, and H. Esmaiel, “Revolutionizing diabetic retinopathy diagnosis through advanced deep learning techniques: Harnessing the power of GAN model with transfer learning and the DiaGAN-CNN model,” *Biomed Signal Process Control*, vol. 99, Jan. 2025, doi: 10.1016/J.BSPC.2024.106790.

[6] M. H. Ashraf and H. Alghamdi, “HFF-Net: A hybrid convolutional neural network for diabetic retinopathy screening and grading,” *Biomedical Technology*, vol. 8, pp. 50–64, Dec. 2024, doi: 10.1016/J.BMT.2024.09.004.