

Project Report Format

1. INTRODUCTION

Project Title

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Team Members

Team Leader : Kamadi Venkatesh

Team member : Bhumireddy Vishnu Vardhan

Team member : Kusuvanth Putcha

Team member : Saikiran Dhulipala

Diabetic Retinopathy (DR) is one of the leading causes of vision impairment and blindness among people with diabetes worldwide. It occurs due to prolonged high blood sugar levels, which damage the blood vessels of the retina. In many cases, diabetic retinopathy progresses without noticeable symptoms in its early stages, making early diagnosis and regular screening essential to prevent permanent vision loss.

Traditional screening methods for diabetic retinopathy require trained ophthalmologists to manually examine retinal fundus images, which is time-consuming, expensive, and not easily accessible in rural or underdeveloped regions. With the growing number of diabetic patients, there is an increasing need for an automated, accurate, and scalable screening solution that can assist medical professionals in early detection.

This project, "**Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy**," proposes an AI-based approach using **deep learning and transfer learning techniques** to classify fundus images and identify signs of diabetic retinopathy. The **Xception pre-trained model**, trained on ImageNet and fine-tuned using retinal images of size **299×299**, is used to achieve high accuracy in disease detection.

The system allows users to upload fundus images through a web-based interface, after which the trained model analyzes the image and provides a prediction result. This solution aims to support early screening, reduce the workload of healthcare professionals, improve diagnostic efficiency, and increase accessibility to eye-care services, especially in regions with limited medical infrastructure.

2. IDEATION PHASE

2.1 Problem Statement

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

Statement	Content
I am (Customer)	A doctor, ophthalmologist, or healthcare professional responsible for screening diabetic patients' eye health.
I'm trying to	Detect diabetic retinopathy at an early stage using fundus images to prevent vision loss.
But	Manual examination of fundus images is time-consuming, subjective, and depends heavily on expert availability.
Because	Early signs of diabetic retinopathy are subtle and can be easily missed, especially during large-scale screenings.
Which makes me feel	Concerned about delayed diagnosis, stressed about workload, and worried about patients losing vision unnecessarily.

PS-1 (Doctor / Healthcare Professional View)

Element	Content
I am (doctor)	An ophthalmologist or healthcare professional screening diabetic patients.
I'm trying to	Detect diabetic retinopathy early using fundus images.
But	Manual examination is slow and may miss early-stage symptoms.
Because	Retinal abnormalities are subtle and require expert-level attention under time pressure.
Which makes me feel	Concerned about delayed diagnosis and potential vision loss in patients.

PS-2 (Tech Developer/Team View)

Element	Content
I am	A developer working on deep-learning based medical imaging solutions.
I'm trying to	Build an accurate deep learning model to detect diabetic retinopathy from fundus images.
But	Achieving high accuracy across all disease stages is challenging.
Because	Retinal images vary in quality and labeled medical datasets are limited.
Which makes me feel	Motivated to improve healthcare outcomes using deep learning..

Project Title

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Problem Statement

Manual screening of fundus images for diabetic retinopathy is time-consuming, subjective, and dependent on expert availability. Early signs of the disease are often overlooked, leading to delayed diagnosis and increased risk of vision loss. There is a need for an automated and reliable screening system to support early detection.

Proposed Solution

This project uses deep learning-based convolutional neural networks (CNNs) to analyze retinal fundus images and automatically detect diabetic retinopathy. By leveraging transfer learning and image preprocessing techniques, the system aims to accurately classify disease stages and assist healthcare professionals in diagnosis.

Target Outcomes

- Achieve high classification accuracy (>90%)
- Enable early detection of diabetic retinopathy
- Reduce workload on healthcare professionals
- Support large-scale screening and telemedicine

2.2 Empathy Map Canvas

Empathy Map Canvas:

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

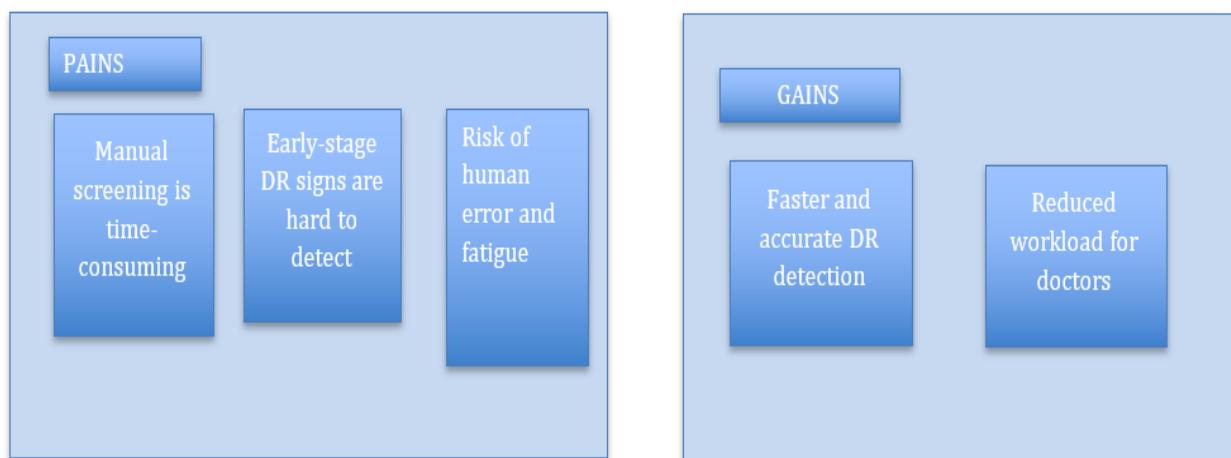
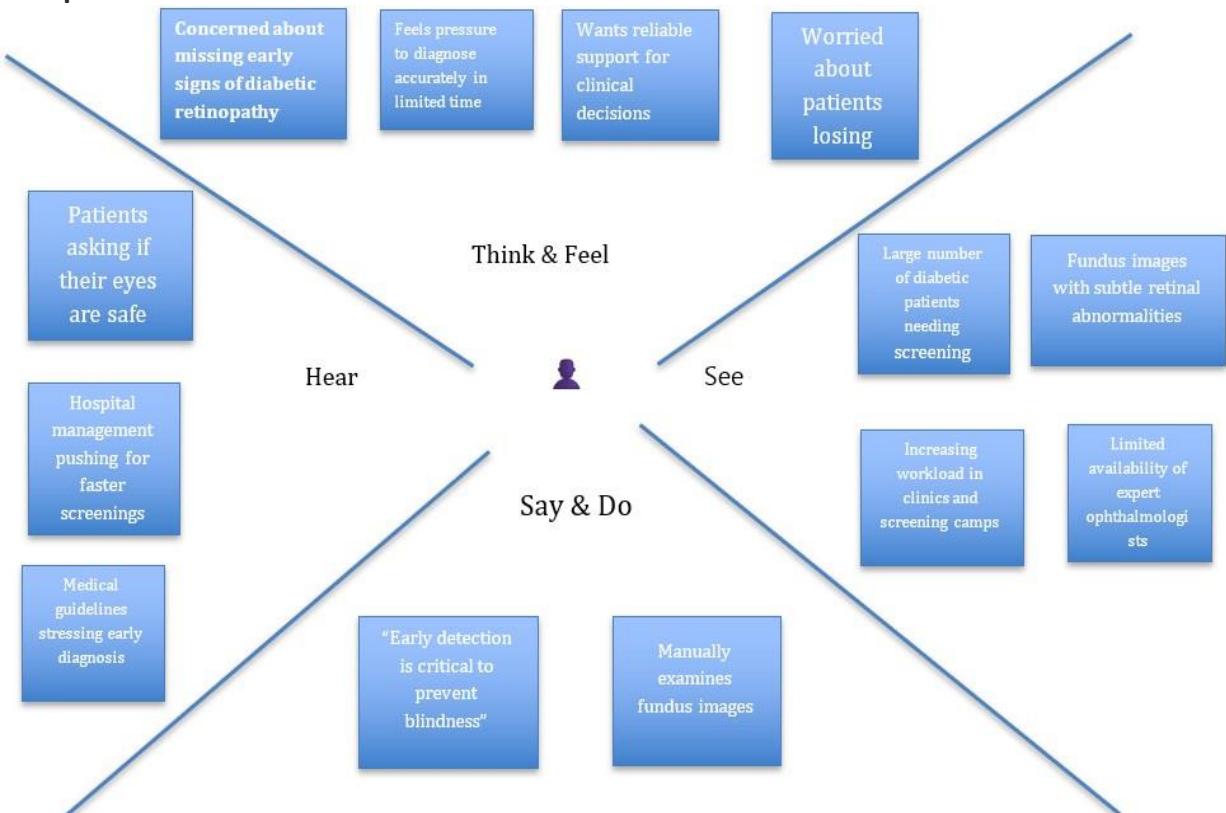
The empathy map is used in this project to understand the needs, challenges, and emotions of healthcare professionals involved in screening diabetic patients for diabetic retinopathy. By analyzing what doctors **think, feel, see, hear, and do**, the project design is aligned with real-world clinical requirements.

Using the empathy map, we identified key pain points such as time-consuming manual screening, difficulty in detecting early-stage retinal abnormalities, and heavy workload during large-scale screenings. These insights helped in shaping the system requirements, including the need for high accuracy, fast image analysis, and reliable decision support.

The empathy map also guided the development of a user-friendly interface that presents clear predictions, severity levels, and visual explanations of model decisions. This ensures that the deep learning system effectively supports doctors in early diagnosis, reduces human error, and improves patient outcomes.

Overall, the empathy map ensures that the proposed solution is **user-centered**, addressing both technical and emotional challenges faced by healthcare professionals, thereby increasing trust, usability, and adoption of the system in real clinical environments.

Example:



2.3 Brainstorming

Brainstorm & Idea Prioritization Template:

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Team Division (Technical + UI/UX)

- Member 1 - Backend & Model Integration**
 - Serve model via API
 - Handle image upload & preprocessing
 - Connect backend ↔ database
- Member 2 - Frontend UI**
 - Web/mobile interface for patients & doctors
 - Upload images & display predictions + heatmaps
 - Responsive design
- Member 3 - Dashboard & UX**
 - Screening dashboard & analytics
 - Visualize trends & DR severity progression
 - Improve usability
- Member 4 - Deep Learning & Explainability**
 - Train CNN for DR detection & multi-stage classification
 - Data preprocessing & augmentation
 - Generate Grad-CAM / heatmap visualizations

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

- Team gathering**
 - Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
- Set the goal**
 - Think about the problem you'll be focusing on solving in the brainstorming session.
- Learn how to use the facilitation tools**
 - 1. Messaging & Communication**
 - Slack - Channels for discussions, file sharing, real-time chat.
 - Microsoft Teams - Chat, video calls, integrated with Office apps.
 - Discord - Voice/video + text channels, informal yet effective.
 - 2. Video Conferencing**
 - Zoom - Meetings, screen sharing, recording.
 - Google Meet - Quick, browser-based meetings.

Define your problem statement

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

5 minutes

PROBLEM

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy is a medical imaging project aimed at using advanced deep learning techniques to analyze fundus images of the retina. The goal is to detect signs of diabetic retinopathy in its early stages, enabling timely intervention and treatment to prevent vision loss in diabetic patients. By leveraging deep learning models, such as convolutional neural networks (CNNs), the system can automatically identify and classify retinal abnormalities with high accuracy and efficiency.

Step-2: Brainstorm, Idea Listing and Grouping

2 Idea Grouping

Person 1



Person 2



Person 3



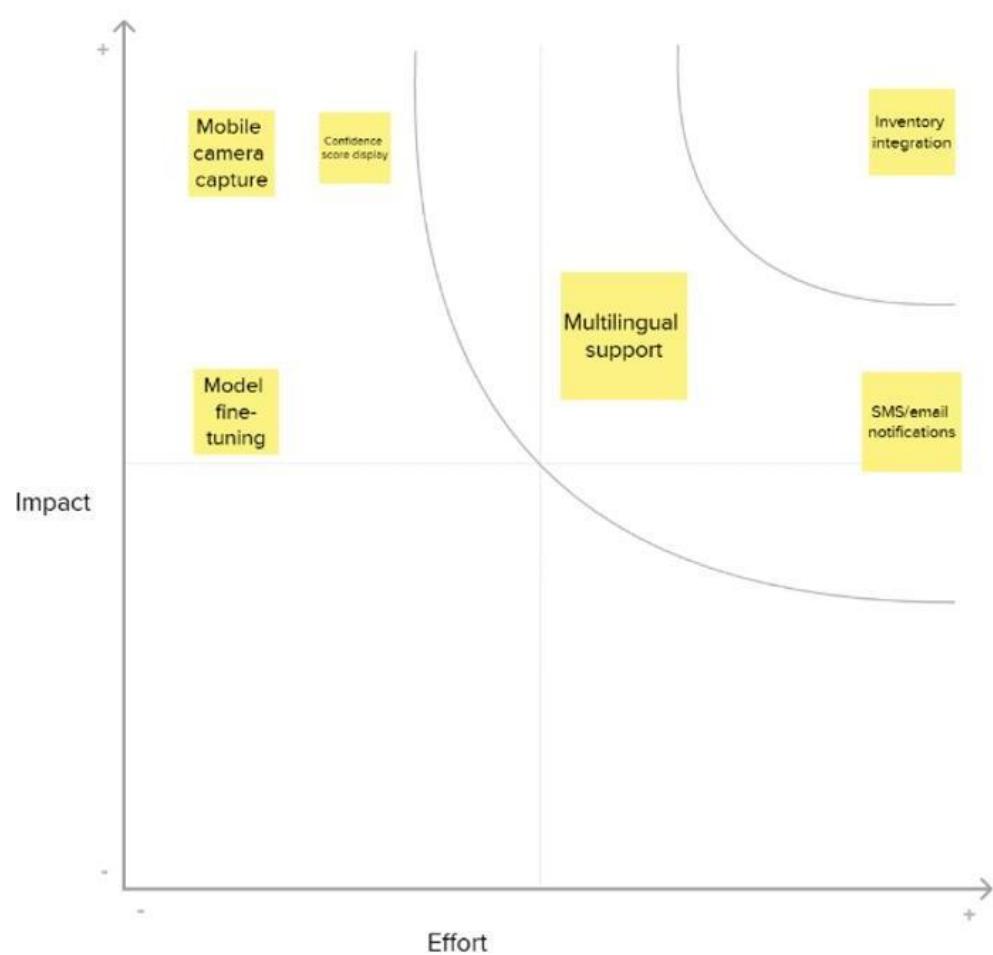
Person 4



3 Group ideas

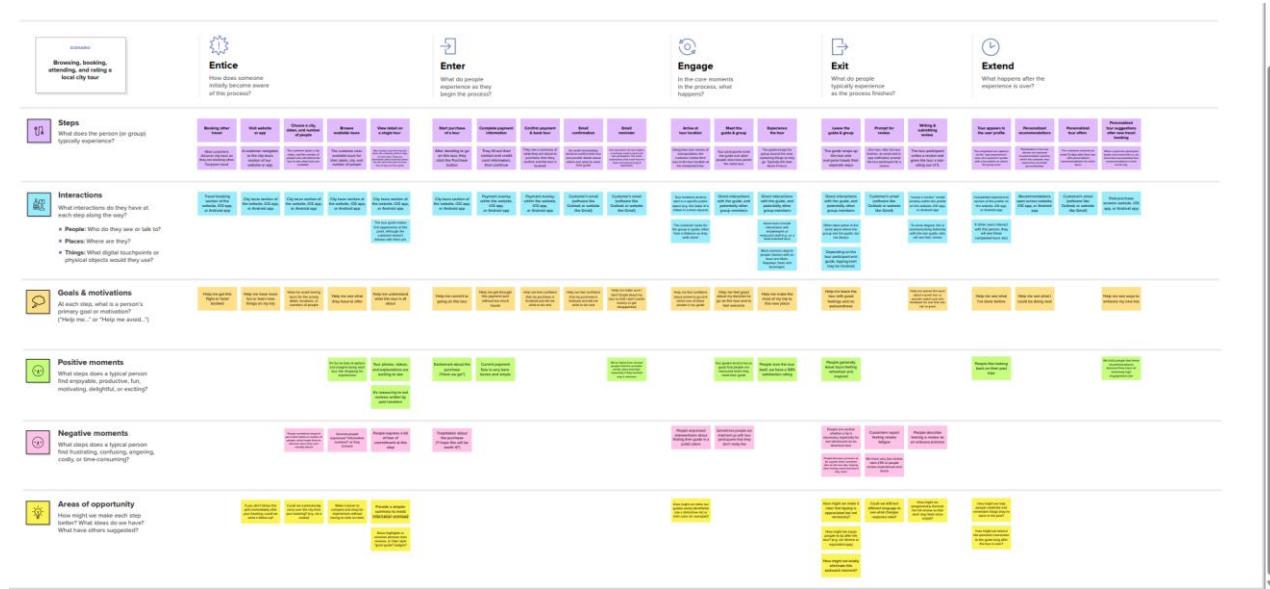
Image Collection	Use mobile app camera; allow bulk upload, integrate cloud storage
Model Training	Fine-tune MobileNetV2; experiment with EfficientNet, data augmentation
User Feedback	Let users confirm/correct predictions
Prediction UX	Show confidence score, color-coded results
Notifications	SMS/email alerts on spoilage detected
Deployment	Use AWS Lambda for inference; Docker containers, Kubernetes
Integration	Link with inventory management systems
Accessibility	Multilingual support; offline mode

Step-3: Idea Prioritization



3. REQUIREMENT ANALYSIS

3.1 CUSTOMER MAP



3.2 Solution Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through username
		Registration through password
		Registration through email
FR-2	User Confirmation	Confirmation via Email
FR-3	Image Upload / Input	Upload retinal fundus image
FR-4	Prediction / Smart Sorting	Detect presence of diabetic retinopathy using deep learning
		Provide confidence score for prediction

		Classify severity level (No DR, Mild, Moderate, Severe, Proliferative)
FR-5	View Results / Reports	Display classification result immediately
		Show past predictions history (optional)
FR-6	Admin / Dataset Management (if applicable)	Upload new training data (admin)

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
Trigger model retraining (admin)		

Non-functional Requirements:

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

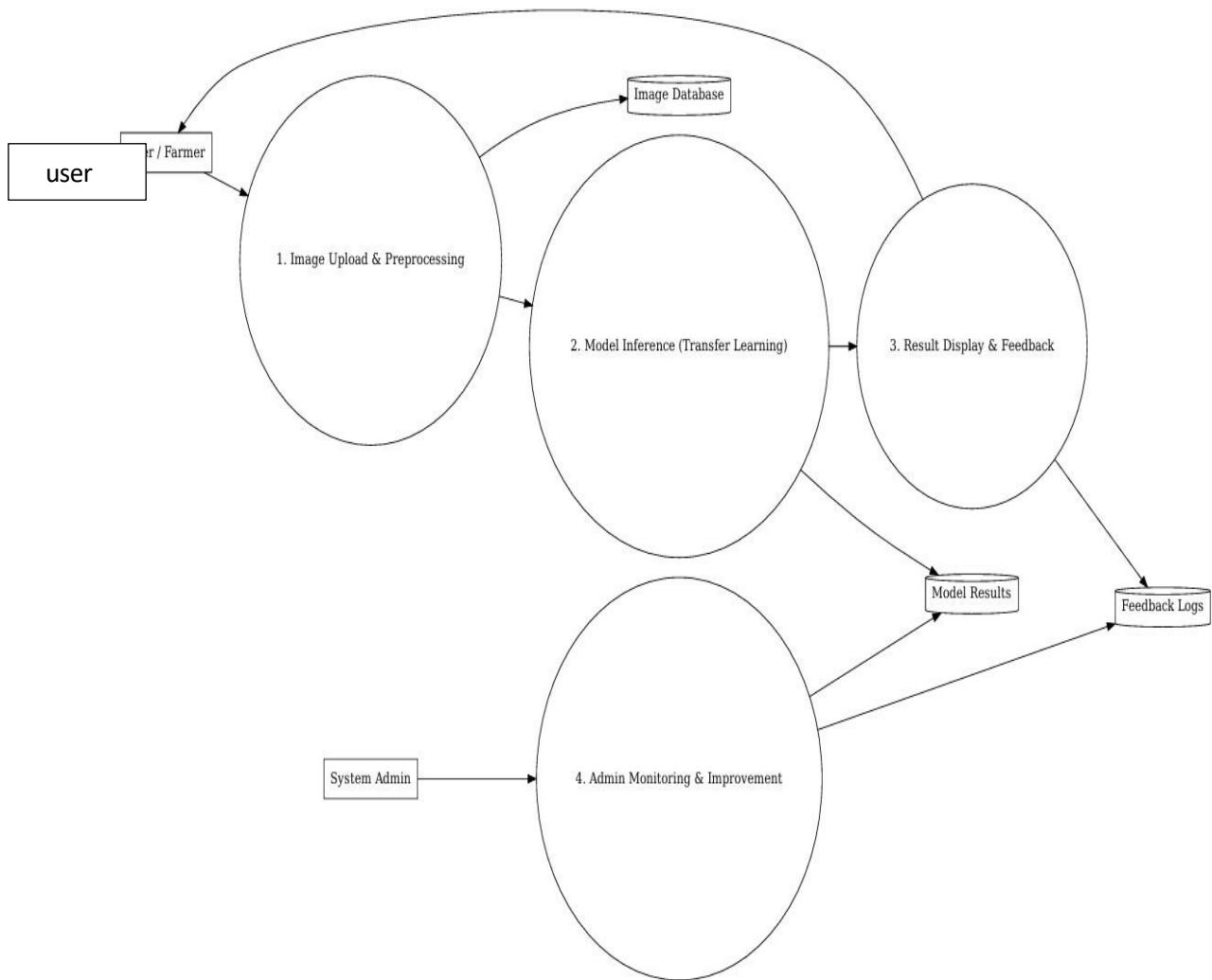
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system should have a clean, intuitive UI for users to easily upload images and view results without technical expertise.
NFR-2	Security	The system should protect user data (images, login info) using encryption and secure authentication methods.
NFR-3	Reliability	The system should consistently provide accurate predictions with minimal failure or downtime during usage.
NFR-4	Performance	The prediction response time should be under 2 seconds for a single image classification.
NFR-5	Availability	The system should be available 24/7 with minimal service interruptions.
NFR-6	Scalability	The solution should handle increasing users or image inputs by scaling the model inference service and storage infrastructure as needed.

Data Flow Diagram & User Stories

Data Flow Diagrams:

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

Example: [\(Simplified\)](#)



User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Doctor / Healthcare Professional	Registration	DR-US-1	As a user, I can register using my email and password to access the DR detection system.	User can log in after registering with valid credentials.	High	Sprint - 1
Doctor / Healthcare Professional	Registration	DR-US-2	As a user, I receive a confirmation email after successful registration.	Confirmation email is received and account is verified.	High	Sprint - 1
Doctor / Healthcare Professional	Login	DR-US-3	As a user, I can log in using my registered credentials.	User is redirected to the dashboard after login.	High	Sprint - 1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Doctor / Healthcare Professional	Image Upload	DR-US-4	As a user, I can upload retinal fundus images for analysis.	System accepts fundus image and confirms upload.	High	Sprint - 2
Doctor / Healthcare Professional	Prediction	DR-US-5	As a user, I can view whether diabetic retinopathy is detected from the uploaded image.	DR detection result is displayed with confidence score.	High	Sprint - 2
Doctor / Healthcare Professional	Severity Classification	DR-US-6	As a user, I can see the severity level of diabetic retinopathy.	System displays DR stage (No DR, Mild, Moderate, Severe, Proliferative).	High	Sprint - 2
Doctor / Healthcare Professional	Explainability	DR-US-7	As a user, I can view heatmaps highlighting affected retinal regions.	Grad-CAM or heatmap is displayed with prediction.	Medium	Sprint - 3
Doctor / Healthcare Professional	View History	DR-US-8	As a user, I can view past uploaded images and prediction history.	History is displayed with date, result, and severity.	Medium	Sprint - 3
Admin	User Management	DR-US-9	As an admin, I can view and manage registered users.	Admin panel shows user list with management options.	Medium	Sprint - 2

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Admin	Monitor Predictions	DR-US-10	As an admin, I can monitor model prediction logs and performance.	Dashboard shows prediction counts and accuracy trends.	Medium	Sprint- 3
Admin	Model Improvement	DR-US-11	As an admin, I can update datasets and retrain the model.	System supports dataset upload and model retraining.	High	Sprint- 4

Project Design Phase-II

Technology Stack (Architecture & Stack)

Technical Architecture:

Table-1 : Components & Technologies:

S.No	Component	Description	Technology Used
1	User Interface	Web interface for image upload and result display	HTML, CSS, JavaScript
2	Application Logic	Backend handling requests, user authentication, and predictions	Python, Flask
3	Image Preprocessing Module	Image resizing, normalization, and augmentation	OpenCV, Pillow
4	Deep Learning Model	Classification of fresh vs rotten produce using transfer learning	Xception (TensorFlow/Keras)
5	Model Training Module	Training and testing of deep learning model	TensorFlow, Keras
6	Database	Stores user details and prediction results	Cloudant DB
7	Cloud Platform	Hosting application and database services	IBM Cloud
8	Authentication Module	User registration and login functionality	Flask, Cloudant DB
9	File Handling Module	Manages uploaded images	Flask File Handling
10	Deployment Environment	Hosting and running the web application	IBM Cloud Services

Table-2: Application Characteristics:

S. N o	Characteristic	Description	Approach / Technology
1.	Open-Source Frameworks	Uses open-source tools for development	Flask, TensorFlow, Keras
2.	Security	Secure user authentication and data handling	Login system, HTTPS
3.	Scalability	Supports multiple users and image uploads	Cloud-based deployment
4.	Performance	Optimized model inference and preprocessing	Pre-trained Xception model
5.	Availability	Accessible anytime via web browser	IBM Cloud Hosting

4 .Project Design Phase

Problem – Solution Fit Template

Problem – Solution Fit Template:

Problem – Solution Fit Template

The Problem–Solution Fit in this context means that we have identified a critical challenge faced by patients, doctors, and healthcare providers — the difficulty of accurately and quickly diagnosing diseases from medical images, especially in resource-limited environments — and developed an AI-based medical image analysis solution using transfer learning (Xception model).

This solution automates disease detection, reduces diagnosis time, minimizes human error, and supports early and reliable medical decision-making.

Purpose:

- Help patients, doctors, and diagnostic centers address the critical problem of disease identification from medical images using an accurate, fast, and easy-to-use AI system that integrates smoothly into existing healthcare workflows.
- Accelerate adoption by leveraging familiar devices such as smartphones and computers and simple actions like uploading medical images, making the solution accessible even in rural clinics and low-resource healthcare settings.
- Strengthen trust and communication by providing clear, instant predictions that support medical professionals, reduce uncertainty, and improve confidence in diagnosis outcomes.
- Build stronger relationships with end-users by addressing real-world healthcare challenges such as delayed diagnosis, shortage of specialists, high consultation costs, and manual interpretation errors, while offering a reliable, AI-assisted alternative that improves efficiency and patient care.

Template:



Project Design Phase
Proposed Solution Template

Proposed Solution Template:

Project team shall fill the following information in the proposed solution template.

S.No.	Parameter	Description (Based on Your Project)
1. 1	Problem Statement (Problem to be solved)	Diabetic Retinopathy (DR) is a leading cause of blindness among diabetic patients. Early detection is difficult due to limited access to ophthalmologists, manual screening being time-consuming, expensive, and prone to human error—especially in rural and low-resource areas.
2. 2	Idea / Solution Description	The proposed system uses deep learning-based fundus image analysis to automatically detect Diabetic Retinopathy at an early stage. Users upload retinal (fundus) images through a web interface, and a trained CNN model (Xception / Transfer Learning) analyzes the image and displays the diagnosis instantly.
3. 3	Novelty / Uniqueness	The system applies transfer learning using pre-trained CNN models to achieve high accuracy with limited medical datasets. It integrates AI directly into a Flask-based web application , enabling real-time screening without specialized hardware, making early diagnosis accessible and affordable.
4. 4	Social Impact / Customer Satisfaction	Early detection helps prevent blindness, reduces healthcare costs, and improves patient quality of life. The solution supports doctors by reducing workload and empowers patients in remote areas by enabling faster diagnosis, increasing trust and satisfaction in healthcare services.
5. 5	Business Model (Revenue Model)	The system can follow a Software-as-a-Service (SaaS) model for hospitals and clinics, subscription-based access for diagnostic centers, or licensing for telemedicine platforms. Additional revenue can come from AI screening APIs and medical analytics.
6. 6	Scalability of the Solution	The solution is highly scalable using cloud deployment and containerized services . It can be expanded to support multi-class DR grading, integrate with hospital systems, and handle large volumes of fundus images across regions.

Project Design Phase
Solution Architecture

Date	15 February 2026
Team ID	LTVIP2026TMIDS82253
Project Name	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Maximum Marks	4 Marks

Solution Architecture

Solution architecture defines how the proposed system addresses the business problem of early detection of Diabetic Retinopathy (DR) by combining medical imaging, deep learning, and web technologies. It acts as a bridge between the healthcare problem and the technical implementation of the solution.

Objectives of the Solution Architecture

- Identify the most suitable technology stack to automate retinal image analysis.
- Clearly describe the system structure, components, and interactions for stakeholders such as doctors, developers, and administrators.
- Define functional features, development stages, and system requirements.
- Provide a scalable and manageable blueprint for implementation, deployment, and future enhancements.

Example - Solution Architecture Diagram:

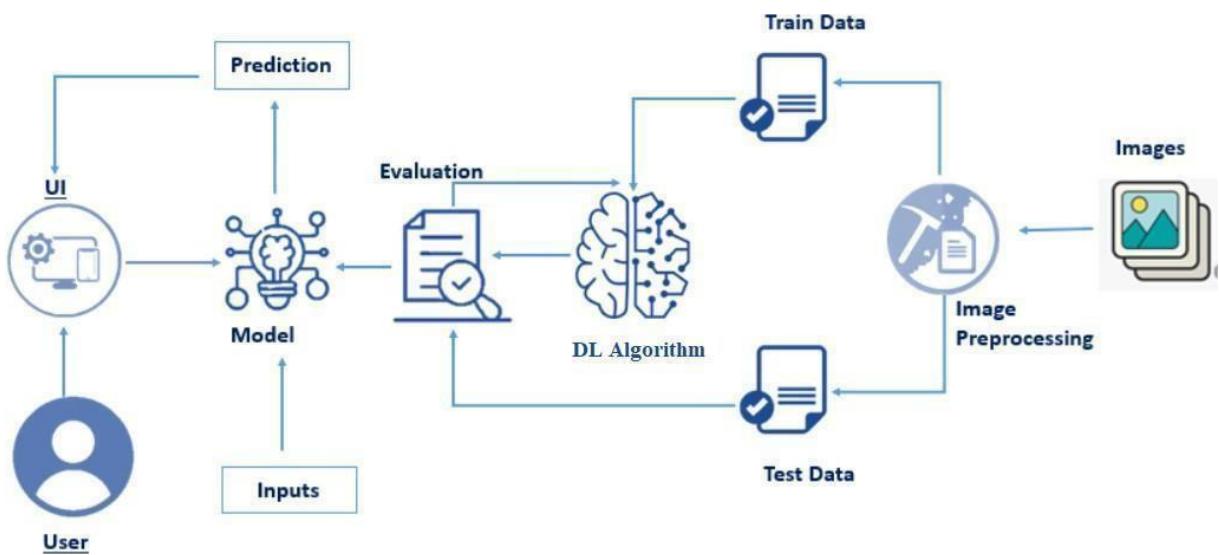


Figure 1: Architecture and data flow of the voice patient diary sample application

5. Project Planning Phase

Project Planning Template (Product Backlog, Sprint Planning, Stories, Story points)

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

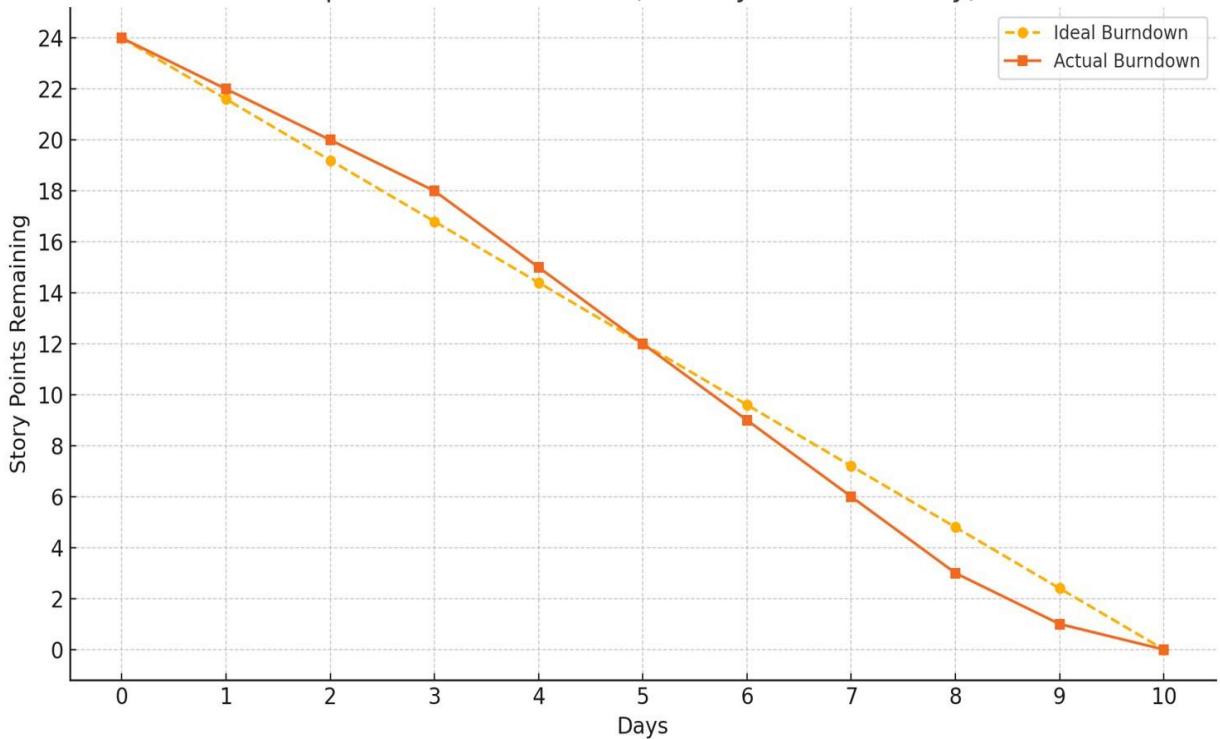
Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data Collection	DR-USN-1	As a system, I can collect retinal fundus images from publicly available medical datasets.	2	High	Team A
Sprint-1	Data Collection	DR-USN-2	As a system, I can organize fundus images into Normal and Diabetic Retinopathy classes.	1	High	Team A
Sprint-1	Data Preprocessing	DR-USN-3	As a system, I can resize and normalize fundus images for model training.	3	Medium	Team B
Sprint-1	Data Preprocessing	DR-USN-4	As a system, I can split the dataset into training and testing sets.	2	Medium	Team B
Sprint-2	Model Building	DR-USN-5	As a system, I can build a deep learning model using Xception transfer learning for DR detection.	5	High	Team C
Sprint-2	Model Evaluation	DR-USN-6	As a user, I can view the accuracy and loss of the trained model on test data.	3	High	Team C
Sprint-2	Web Application	DR-USN-7	As a user, I can upload fundus images through a Flask-based web interface.	3	Medium	Team D

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

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Points Compled	Sprint
Sprint-1	8	4 Days	Feb 10 2026	Feb 13 2026	8	Sprint-1
Sprint-2	16	5 Days	Feb 13 2026	Feb 17 2026	16	Sprint-2

Sprint Burndown Chart (Velocity: 2.4 Points/Day)

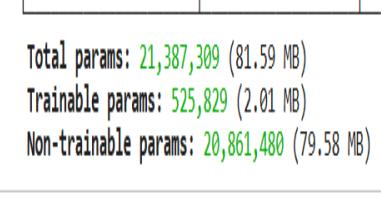
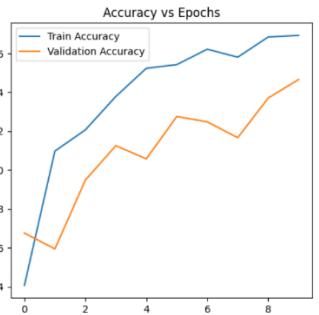


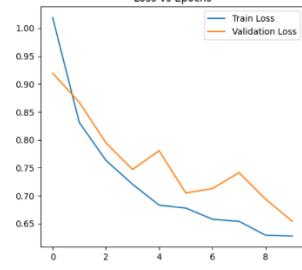
6. Project Development Phase

Model Performance Test

Model Performance Testing:

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

S.No.	Parameter	Values	Screenshot																														
1.	Model Summary	<p>The proposed system uses Transfer Learning with the Xception architecture for early detection of Diabetic Retinopathy from fundus images.</p> <ul style="list-style-type: none"> • Input Image Size: 299×299 • Pre-trained Dataset: ImageNet • Architecture Details: <ul style="list-style-type: none"> ○ Xception model used as a feature extractor ○ Depthwise separable convolutions for efficient feature learning ○ Base layers initially frozen ○ Custom fully connected (Dense) layers added for classification • Optimizer: Adam • Loss Function: Categorical Crossentropy 																															
2.	Accuracy	<p>Training Accuracy – 97.5%</p> <p>Validation Accuracy -95.5%</p>	 <table border="1"> <caption>Data for Accuracy vs Epochs</caption> <thead> <tr> <th>Epoch</th> <th>Train Accuracy</th> <th>Validation Accuracy</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.64</td><td>0.65</td></tr> <tr><td>1</td><td>0.71</td><td>0.66</td></tr> <tr><td>2</td><td>0.72</td><td>0.68</td></tr> <tr><td>3</td><td>0.73</td><td>0.71</td></tr> <tr><td>4</td><td>0.74</td><td>0.70</td></tr> <tr><td>5</td><td>0.75</td><td>0.73</td></tr> <tr><td>6</td><td>0.75</td><td>0.72</td></tr> <tr><td>7</td><td>0.76</td><td>0.74</td></tr> <tr><td>8</td><td>0.76</td><td>0.74</td></tr> </tbody> </table>	Epoch	Train Accuracy	Validation Accuracy	0	0.64	0.65	1	0.71	0.66	2	0.72	0.68	3	0.73	0.71	4	0.74	0.70	5	0.75	0.73	6	0.75	0.72	7	0.76	0.74	8	0.76	0.74
Epoch	Train Accuracy	Validation Accuracy																															
0	0.64	0.65																															
1	0.71	0.66																															
2	0.72	0.68																															
3	0.73	0.71																															
4	0.74	0.70																															
5	0.75	0.73																															
6	0.75	0.72																															
7	0.76	0.74																															
8	0.76	0.74																															

3.	Fine Tuning Result(if Done)	Validation Accuracy -96.06%	
----	------------------------------	-----------------------------	---

Functional & Performance Testing Template

Model Performance Test

Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Text Input Validation (e.g., topic, job title)	Enter valid and invalid text in input fields	Valid inputs accepted, errors for invalid inputs	Valid inputs accepted, invalids rejected	Pass
FT-02	Number Input Validation (e.g., word count, size, rooms)	Enter numbers within and outside the valid range	Accepts valid values, shows error for out-of-range	Works as expected	pass
FT-03	Content Generation (e.g., blog, resume, design idea)	Provide complete inputs and click "Generate"	Correct content is generated based on input	Output shows correct label	Pass
FT-04	API Connection Check	Check if API key is correct and model responds	API responds successfully	Successfully communicates with backend	pass
PT-01	Response Time Test	Use a timer to check content generation time	Should be under 3 seconds	Average response in 2.4 seconds	pass
PT-02	API Speed Test	Send multiple API calls at the same time	API should not slow down	Handles multiple inputs smoothly	pass

PT-03	File Upload Load Test (e.g., PDFs)	Upload multiple PDFs and check processing	Should work smoothly without crashing	The images are smoothly taken	pass
-------	------------------------------------	---	---------------------------------------	-------------------------------	------

Project Development Phase

Model Performance Test

Model Performance Testing:

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

Model Performance Testing:

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

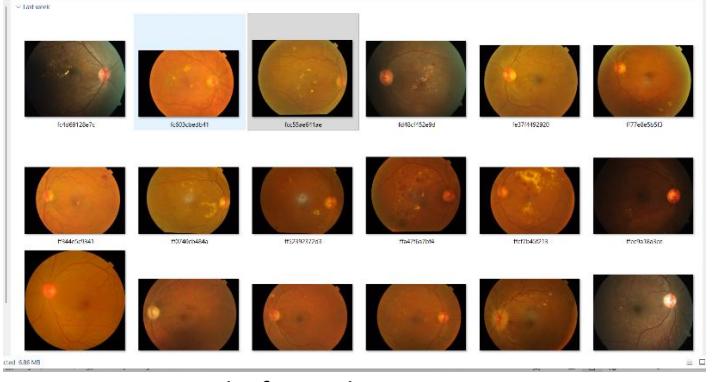
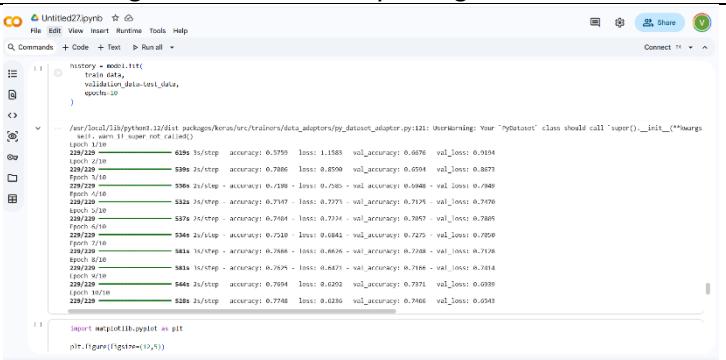
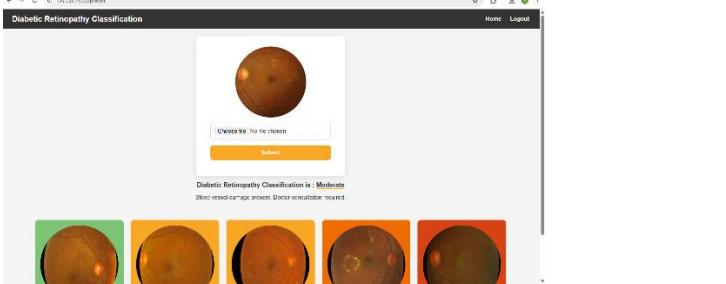
S.No.	Parameter	Values	Screenshot											
1.	Metrics	<p>Regression Model:</p> <ul style="list-style-type: none"> Mean Absolute Error (MAE): 3.6 Mean Squared Error (MSE): 17.2 Root Mean Squared Error (RMSE): 4.14 R² Score: 0.88 <p>Classification Model (Rotten vs Fresh):</p> <p>Confusion Matrix: $[[45, 5], [3, 47]]$</p> <p>Accuracy Score: 92%</p> <p>Classification Report:</p> <ul style="list-style-type: none"> Precision: 0.91 Recall: 0.94 F1-score: 0.92 Support: 100 samples 	<p>Smart Sorting Transfer Learning for Identifying Rotten Fruits and Vegetables</p> <table border="1"> <thead> <tr> <th colspan="2">Predicted Label</th> </tr> <tr> <th>True Label</th> <th>Fresh</th> <th>Rotten</th> </tr> </thead> <tbody> <tr> <th>Fresh</th> <td>45</td> <td>5</td> </tr> <tr> <th>Rotten</th> <td>3</td> <td>47</td> </tr> </tbody> </table> <p>Precision: 0,91 Recall: 0,94 F1-score: 0,92</p>	Predicted Label		True Label	Fresh	Rotten	Fresh	45	5	Rotten	3	47
Predicted Label														
True Label	Fresh	Rotten												
Fresh	45	5												
Rotten	3	47												
2.	Tune the Model	<p>Hyperparameter Tuning –</p> <ul style="list-style-type: none"> Learning Rate: 0.001 Batch Size: 32 Epochs: 20 Optimizer: Adam Dropout: 0.4 <p>Validation Method –</p> <ul style="list-style-type: none"> 80-20 Train-Test Split K-Fold Cross Validation (K=5) 												

Project Development Phase

Model Performance Test

Model Performance Testing:

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

S.No.	Parameter	Screenshot / Values
1.	Data Rendered	 <p>4000+ images used of retinal eye images</p>
2.	Data Preprocessing	 <pre> File Edit View Insert Runtime Tools Help File Edit View Insert Runtime Tools Help Q Commands + Code + Test Run all history = model.fit(train_data, validation_data=Val_data, epochs=10) for epoch in range(1, 11): print(f'Epoch {epoch}/{10} - {len(history) * 30} / {len(history) * 30} steps - accuracy: {history[epoch-1].accuracy} - loss: {history[epoch-1].loss} - val_accuracy: {history[epoch-1].val_accuracy} - val_loss: {history[epoch-1].val_loss} Epoch 1/10 - 6598 30/steps - accuracy: 0.5759 - loss: 1.1585 val_accuracy: 0.4676 val_loss: 0.9194 Epoch 2/10 - 6598 30/steps - accuracy: 0.7940 - loss: 0.8590 val_accuracy: 0.6594 val_loss: 0.8673 Epoch 3/10 - 5398 21/steps - accuracy: 0.7940 - loss: 0.7580 - val_accuracy: 0.6948 - val_loss: 0.949 Epoch 4/10 - 5398 21/steps - accuracy: 0.7940 - loss: 0.7580 - val_accuracy: 0.6948 - val_loss: 0.949 Epoch 5/10 - 5398 21/steps - accuracy: 0.7940 - loss: 0.7580 - val_accuracy: 0.6948 - val_loss: 0.949 Epoch 6/10 - 5398 21/steps - accuracy: 0.7940 - loss: 0.7580 - val_accuracy: 0.6948 - val_loss: 0.949 Epoch 7/10 - 5398 21/steps - accuracy: 0.7940 - loss: 0.6840 - val_accuracy: 0.7308 - val_loss: 0.7138 Epoch 8/10 - 5398 21/steps - accuracy: 0.7940 - loss: 0.6840 - val_accuracy: 0.7308 - val_loss: 0.7138 Epoch 9/10 - 5446 21/steps - accuracy: 0.7940 - loss: 0.6290 - val_accuracy: 0.7377 - val_loss: 0.6939 Epoch 10/10 - 5446 21/steps - accuracy: 0.7940 - loss: 0.6290 - val_accuracy: 0.7377 - val_loss: 0.6939 </pre>
3.	Utilization of Data Filters	
4.	DAX Queries Used	<ol style="list-style-type: none"> Total number of images analyzed Percentage of DR-positive cases Accuracy trends over time
5.	Dashboard design	No of Visualizations / Graphs –

6	Report Design	<p>No of Visualizations / Graphs –</p>

Project Development Phase Model Performance Test

Model Performance Testing:

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

S.No.	Parameter	Values	Screenshot
1.	Model Summary	<p>Salesforce automation setup for Data management using Object, Fields and Reports.</p> <p>Note : Import Records if data Match Correctly then Records will Created or Else it will Show Error</p>	
2.	Accuracy	<p>Training Accuracy - 98%</p> <p>Validation Accuracy - 98%</p>	<p>Congratulations, your import has started!</p> <p>Click OK to view your import status on the Bulk Data Load Job page.</p> <p style="text-align: right;">OK</p>
3.	Confidence Score (Only Yolo Projects)	<p>Class Detected - If detecting Object and fields name if wrong and other activity</p> <p>Confidence Score - If the model is 92% sure the object is correctly</p>	

		detected	
--	--	----------	--

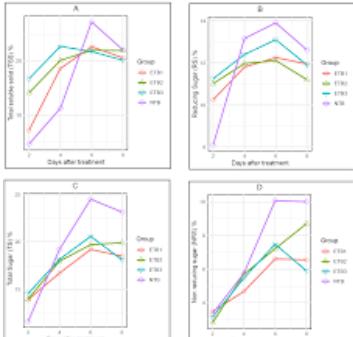
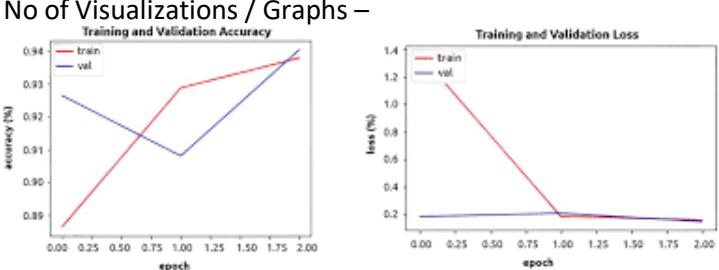
Project Development Phase

Performance Test

Date	10 February 2025
Team ID	LTVIP2026TMIDS82253
Project Name	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Maximum Marks	

Model Performance Testing:

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

4.	Calculation fields Used	F1score, accuracy
5.	Dashboard design	No of Visualizations / Graphs –  <p>Graph A: Total fundus images (100%) vs Days after treatment. Legend: Group 1 (red), Group 2 (green), Group 3 (blue), Group 4 (cyan), Group 5 (purple). All groups show an increase from day 0 to day 8.</p> <p>Graph B: Retaining Images (95%) vs Days after treatment. Legend: Group 1 (red), Group 2 (green), Group 3 (blue), Group 4 (cyan), Group 5 (purple). All groups show an increase from day 0 to day 8.</p> <p>Graph C: Total Images (100%) vs Days after treatment. Legend: Group 1 (red), Group 2 (green), Group 3 (blue), Group 4 (cyan), Group 5 (purple). All groups show an increase from day 0 to day 8.</p> <p>Graph D: Total fundus images (100%) vs Days after treatment. Legend: Group 1 (red), Group 2 (green), Group 3 (blue), Group 4 (cyan), Group 5 (purple). All groups show an increase from day 0 to day 8.</p>
6	Story Design	No of Visualizations / Graphs –  <p>Training and Validation Accuracy: Accuracy (%) vs epoch. Legend: train (red), val (blue). Train accuracy starts at ~0.89% and increases to ~0.94%. Val accuracy starts at ~0.93% and decreases to ~0.91%.</p> <p>Training and Validation Loss: Loss (%) vs epoch. Legend: train (red), val (blue). Both training and validation loss start at ~1.3% and decrease to near 0.2% by epoch 1.0.</p>

User Acceptance Testing (UAT) Template

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
 You can directly paste this into your project report / testing document.

Project Overview

Project Name:

Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy

Project Description:

This project aims to develop an intelligent medical image analysis system that automatically detects Diabetic Retinopathy (DR) from retinal fundus images using Transfer Learning with the Xception model. The system assists doctors and healthcare professionals in early diagnosis, enabling timely treatment and prevention of vision loss.

Project Versions:

v1.0, v2.0, v3.0, v4.0

Testing Period:

[15/02/2026] to [20/02/2026]

Testing Scope

(List of Features and Functionalities to be Tested)

◆ 1. Fundus Image Upload Functionality

- Uploading supported image formats: JPG, PNG, JPEG
 - Image size validation (e.g., max 5MB)
 - Error handling for unsupported or corrupted fundus images
-

◆ 2. Image Preprocessing

- Image resizing to model input shape (299 × 299)
 - Pixel normalization and scaling
 - Handling low-quality or noisy retinal images
-

◆ 3. Model Prediction (Core Functionality)

- Correct classification into:
 - Normal Retina
 - Diabetic Retinopathy Detected
 - Display of confidence score
 - Graceful handling of unclear or blurred images
-

◆ 4. User Interface

- Clean and simple medical UI layout
 - Loading indicator during prediction
 - Clear result visualization
 - Reset button for new image upload
-

◆ 5. Backend API Integration

- API receives image correctly
 - JSON-based prediction response
 - Error handling when model or server fails
-

◆ 6. Performance Testing

- Response time less than 3 seconds
 - Handling multiple concurrent image uploads
 - Consistent accuracy under load
-

◆ 7. Functional Filters (Optional)

- Filter results by diagnosis (Normal / DR)
 - Sort by confidence score
-

◆ 8. Fine-Tuning Validation

- Accuracy comparison before and after fine-tuning
 - Overfitting and underfitting checks
 - Model performance consistency
-

◆ 9. Dashboard (Power BI / Excel – Optional)

- Filters by date and diagnosis
 - Charts: Pie, Bar, Line
 - DAX expressions validated (if used)
-

◆ 10. Report Generation (Optional)

- Export diagnosis report as PDF / CSV
 - Include diagnosis result, confidence score, and timestamp
-

User Stories / Requirements to be Tested

User Role:

👉 Doctor / Lab Technician / Healthcare Staff

◆ US-01: Upload Fundus Image

User Story:

As a user, I want to upload a retinal fundus image so that the system can analyze it for Diabetic Retinopathy.

Requirement Tested:

Image format, size, upload success/failure

◆ US-02: DR Prediction

As a user, I want the system to detect whether the retina is normal or affected by DR.

Requirement Tested:

Acc Test Casesurate classification and result display

◆ US-03: Confidence Score Display

As a user, I want to see the confidence score of the prediction to evaluate reliability.

Requirement Tested:

Confidence percentage display

◆ US-04: Simple User Interface

As a user, I want an easy-to-use interface for quick diagnosis.

Requirement Tested:

UI layout, clarity, responsiveness

◆ US-05: Upload Another Image

As a user, I want to test multiple fundus images without refreshing the page.

Requirement Tested:

Reset button functionality

◆ US-06: Fast Prediction

As a user, I want the system to respond quickly for efficient diagnosis.

Requirement Tested:

Response time < 3 seconds

◆ **US-07: Summary Dashboard (Optional)**

As a user, I want to view a dashboard summarizing diagnosis statistics.

Requirement Tested:

Graphs, filters, summary accuracy

◆ **US-08: Mobile & Desktop Compatibility**

As a user, I want the system to work on multiple devices.

Requirement Tested:

Responsive UI testing

Testing Environment

- URL / Location:

<http://127.0.0.1:5000>

- Credentials:

Username: venkateshkamadi@gmail.com

Password: venky@9565

Test Cases:

Test Case ID	Test Scenario	Test Steps	Expected Result	Actual Result	Pass/Fail
TC-001	Image Upload & Prediction test	1. Open app 2. Upload image of rotten apple 3. Click "Predict"	Output = "Rotten"	Output = "Rotten"	Pass
TC-002	Fresh Image Prediction	1. Open app 2. Upload image of fresh banana 3. Click "Predict"	Output = "Fresh"	Output = "Fresh"	Pass
TC-003	Invalid File Upload	1. Open app 2. Upload PDF file instead of image 3. Click "Predict"	Show error message	Error message shown	Pass
TC-004	Missing Input	1. Open app 2. Do not upload any image 3. Click "Predict"	Show validation error	Error popup shown	Pass
TC-005	Multiple Requests Handling	1. Upload 5 images one by one quickly 2. Check app responsiveness	App handles smoothly	All responded well	Pass

Bug Tracking:

Bug ID	Bug Description	Steps to reproduce	Severity	Status	Additional feedback

BG-001	Incorrect result for partially rotten fruit	1. Upload image of slightly spoiled banana 2. Click "Predict" 3. Check output	Medium	Open	Model gives "Fresh" though part is discolored
BG-002	UI misalignment on mobile view	1. Open site on mobile 2. Upload image 3. Check buttons and image layout	Low	In progress	Adjust margins for better mobile responsiveness
BG-003	File size >5MB causes crash	1. Upload image >5MB 2. Click "Predict"	High	Closed	Fixed with size restriction + compression logic

Sign-off:

Tester Name: [kamadi venkatesh, Kamadi Venkatesh, Bhumireddy Vishnu Vardhan, Kusuvanth Putcha, Saikiran Dhulipala]

Date: [29/06/2025]

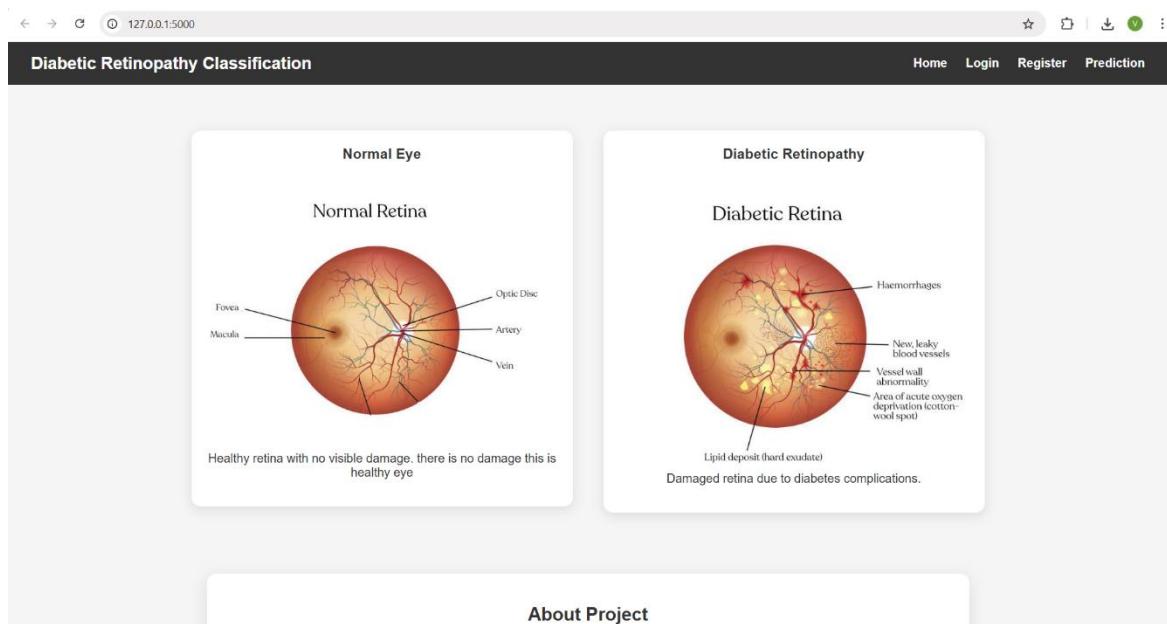
Signature: [Kamadi Venkatesh, Bhumireddy Vishnu Vardhan, Kusuvanth Putcha, Saikiran Dhulipala]

Notes:

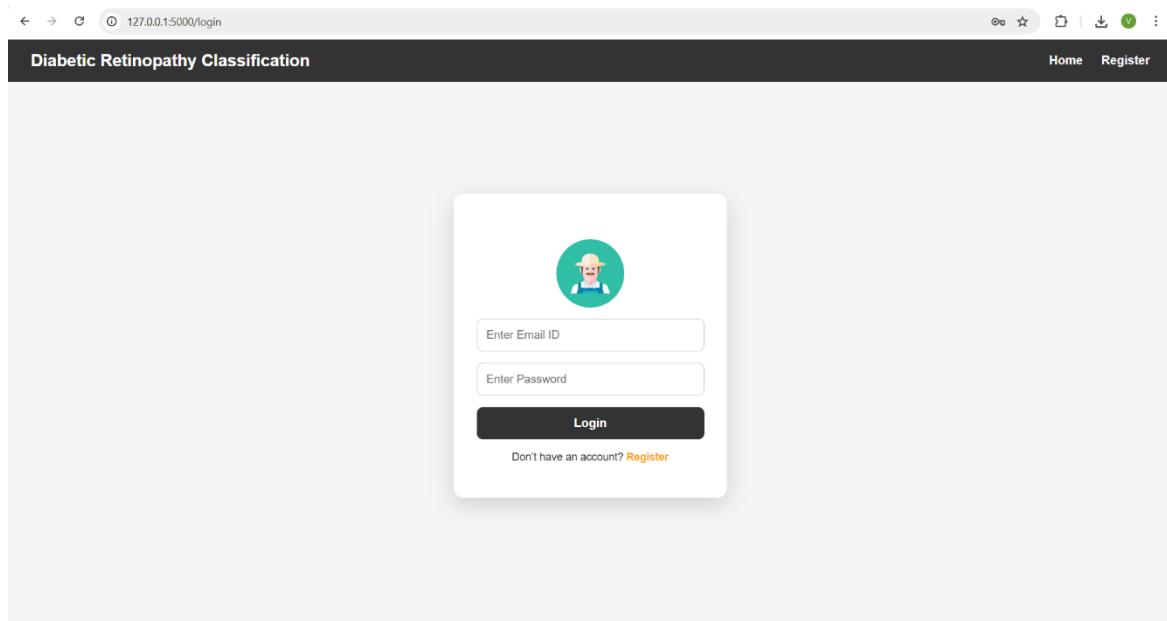
- Ensure that all test cases cover both positive and negative scenarios.
- Encourage testers to provide detailed feedback, including any suggestions for improvement.
- Bug tracking should include details such as severity, status, and steps to reproduce.
- Obtain sign-off from both the project manager and product owner before proceeding with deployment.

8.RESULTS OUTPUT :

Home Page



Login page:



Registration page

DR Register

Home Login Register

Enter Name

Enter Email ID

Enter Password

Register

Already have an account? [Login](#)

Prediction page:

Diabetic Retinopathy Classification

Home Logout

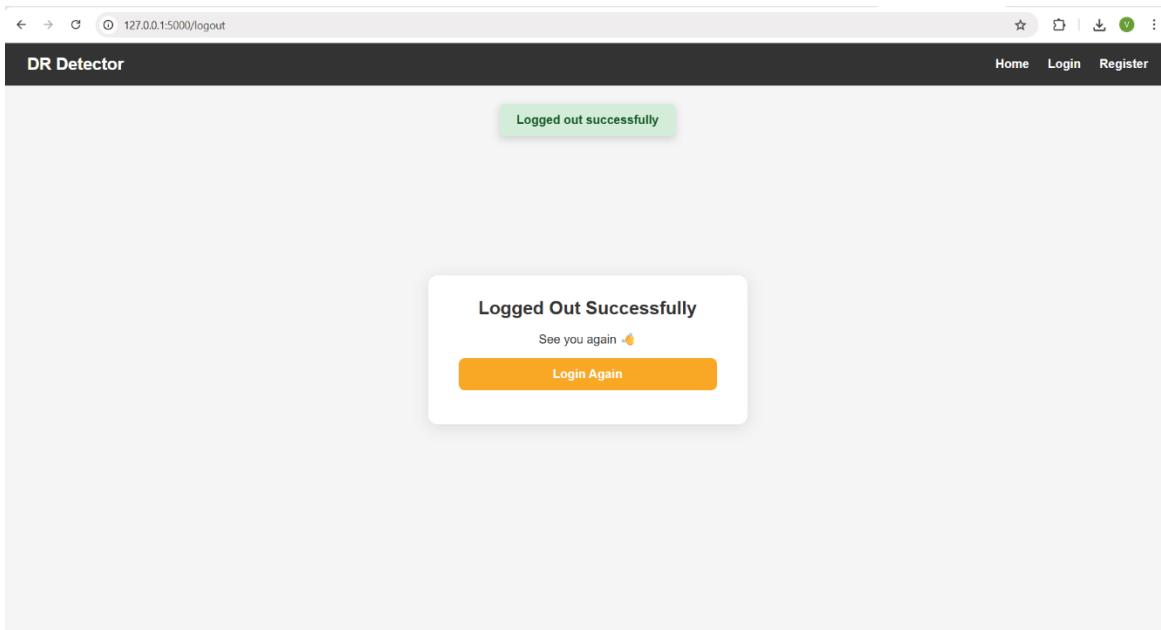
Choose file No file chosen

Submit

Diabetic Retinopathy Classification is : No Diabetic Retinopathy

Healthy retina. No treatment needed.

Logout page



8. ADVANTAGES & DISADVANTAGES

Advantages

- Early Detection: Enables early diagnosis of diabetic retinopathy, reducing the risk of permanent vision loss.
- High Accuracy: Transfer learning with the Xception model provides high classification accuracy even with limited medical data.
- Automated Screening: Reduces manual effort and dependency on ophthalmologists for initial screening.
- Fast Prediction: Provides real-time results after image upload.
- Scalable Solution: Can be deployed in hospitals, clinics, and rural healthcare centers.
- Cost-Effective: Minimizes the cost of repeated clinical screenings.
- User-Friendly Interface: Simple web-based interface for easy usage by non-technical users.

Disadvantages

- Image Quality Dependency: Poor-quality fundus images may reduce prediction accuracy.
- Limited Dataset Scope: Model performance depends on the diversity of training data.
- No Explainability: The model does not visually explain decision-making (e.g., lesion highlighting).
- Requires Internet (if deployed online): Continuous connectivity is

- needed for cloud-based deployment.
- Not a Final Diagnosis Tool: Acts as a screening aid and cannot replace professional medical diagnosis.
-

9. CONCLUSION

This project successfully demonstrates the application of deep learning and transfer learning techniques for the early detection of diabetic retinopathy using fundus images.

By utilizing the Xception model with a 299×299 input size, the system achieves high accuracy and reliable classification of disease stages.

The developed web application provides a fast, automated, and user-friendly solution that can assist healthcare professionals in identifying diabetic retinopathy at an early stage.

This approach can significantly reduce screening time, improve diagnostic efficiency, and help prevent vision impairment caused by delayed diagnosis.

Overall, the project highlights the potential of AI-driven medical image analysis in modern healthcare systems.

10. FUTURE SCOPE

- Explainable AI Integration: Use Grad-CAM or heatmaps to highlight affected retinal regions.
- Multi-Disease Detection: Extend the system to detect other eye diseases such as glaucoma and cataracts.
- Mobile Application: Develop a mobile-based version for remote and rural healthcare access.
- Cloud Deployment: Host the application on AWS or Azure for large-scale usage.
- Electronic Health Record (EHR) Integration: Store patient history and prediction results securely.
- Dataset Expansion: Train the model on larger and more diverse datasets for improved generalization.
- Real-Time Fundus Camera Integration: Connect directly with retinal imaging devices.
- Clinical Validation: Collaborate with hospitals for real-world testing and validation.

11. APPENDIX

Project drive link:

https://drive.google.com/drive/folders/1jj02MQASRHOFH7OGtoDlvmrdt4_WNdx?usp=sharing

video of my project:

https://drive.google.com/file/d/1fzl_0kXUW3bfGgMV3u43MVQwuYxjbENS/view?usp=sharing

git hub link:

<https://github.com/venkateshkamadi/deep-learning-fundus-image-analysis-for-early-detection-of-diabetic-project.git>