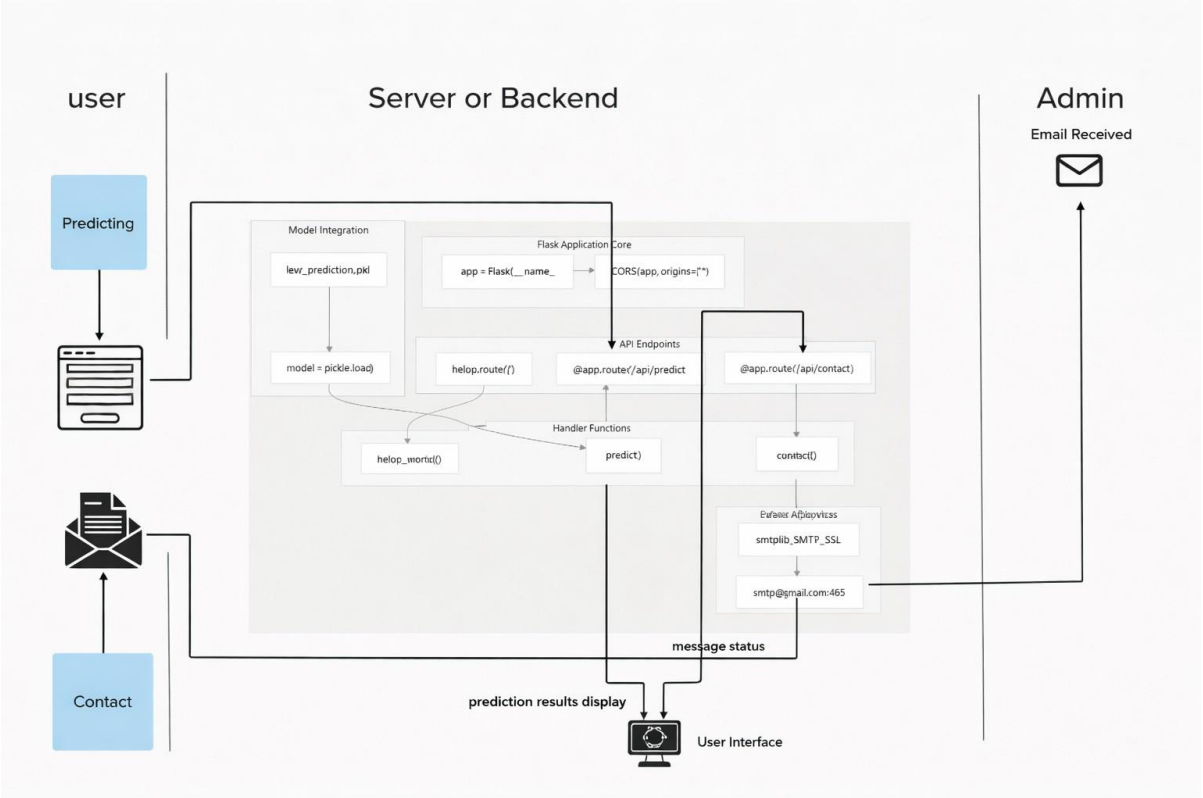
Project Design Phase – II

Technology Stack (Architecture & Stack)

Date:	02 Feb 2026
Team ID:	LTVIP2026TMIDS75194
Project Name:	Deep Learning Fundus Image Analysis for Early Detection of Diabetic Retinopathy
Maximum Marks:	5 Marks

Technical Architecture:



The technical architecture of the proposed system follows a **three-tier architecture** consisting of the **User Interface layer**, **Backend/Application layer**, and **Deep Learning Model layer**. This architecture ensures modularity, scalability, and ease of maintenance.

The user interacts with the system through a web-based interface to upload retinal fundus images and submit prediction requests. The backend server processes these requests using REST APIs and communicates with the trained deep learning model to generate diabetic retinopathy predictions. Administrative communication is handled through email services for user contact requests.

According to the architecture diagram in the reference PDF (page 2), the system separates user interactions, backend processing, and model inference, which improves system reliability and scalability.

Table-1: Components & Technologies

S.No	Component	Description	Technology
1	User Interface	Web-based user interface for uploading retinal fundus images, viewing prediction results, and contacting the administrator	HTML, CSS, JavaScript
2	Application Logic-1	Backend logic to manage routing, request handling, and REST API endpoints	Python (Flask)
3	Application Logic-2	Contact form handler that sends user messages to the administrator via email service	Python `smtplib`, `email.message`
4	Application Logic-3	Deep learning model prediction handler that preprocesses images, performs inference, and returns results	Flask, NumPy, Pandas
5	External API-1	Not applicable in the current version	–
6	External API-2	Email service used to send user queries and notifications to the administrator	Google Gmail SMTP
7	Deep Learning Model	Detects diabetic retinopathy from retinal fundus images using image-based learning	Convolutional Neural Network (TensorFlow / PyTorch)
8	Infrastructure (Server / Cloud)	Application deployed on local server and cloud platforms; container-ready architecture	Flask, Docker-ready, Localhost / Cloud

Explanation of Components

- The **User Interface** provides an easy and intuitive way for users to upload images and view predictions.
- **Application Logic-1** handles API routing and request processing.
- **Application Logic-2** manages user communication through secure email services.
- **Application Logic-3** connects the backend with the trained deep learning model and handles image preprocessing and prediction.
- The **Deep Learning Model** uses CNNs to automatically extract retinal features and classify diabetic retinopathy.
- The **Infrastructure layer** supports scalable deployment using containerized environments.

Table-2: Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	All frameworks and libraries used are open-source	Flask, TensorFlow/PyTorch, Pandas, NumPy, Matplotlib
2	Security Implementations	Input validation, secure email handling, and backend security headers	Gmail Authentication, Flask-CORS
3	Scalable Architecture	Backend is modular; deep learning model can be containerized and scaled	3-tier architecture (Frontend, API, Model)
4	Availability	Backend API deployable using Docker ensures high availability on cloud platforms	Docker, Flask
5	Performance	Optimized image preprocessing and fast inference with response time less than 2 seconds	Flask API, optimized CNN model

Summary

The selected technology stack provides a robust foundation for building a scalable, secure, and high-performance diabetic retinopathy detection system. The combination of a web-based interface, Flask backend, and CNN-based deep learning model ensures accurate medical image analysis and real-time predictions. The architecture supports future enhancements such as cloud deployment, mobile integration, and multi-class disease severity detection.

References

- https://c4model.com/
- https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/
- https://www.ibm.com/cloud/architecture
- https://aws.amazon.com/architecture
- https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d