

Project Design Phase-II Technology Stack (Architecture & Stack)

Date	31 January 2025
Team ID	LTVIP2026TMIDS76616
Project Name	HematoVision – Advanced Blood Cell Classification Using Transfer Learning
Maximum Marks	4 Marks

Technical Architecture:

HematoVision follows a three-layer architecture:

1 Presentation Layer

- Web interface built using HTML, CSS, and JavaScript.
- Allows lab technicians to upload blood smear images.
- Displays prediction results clearly.

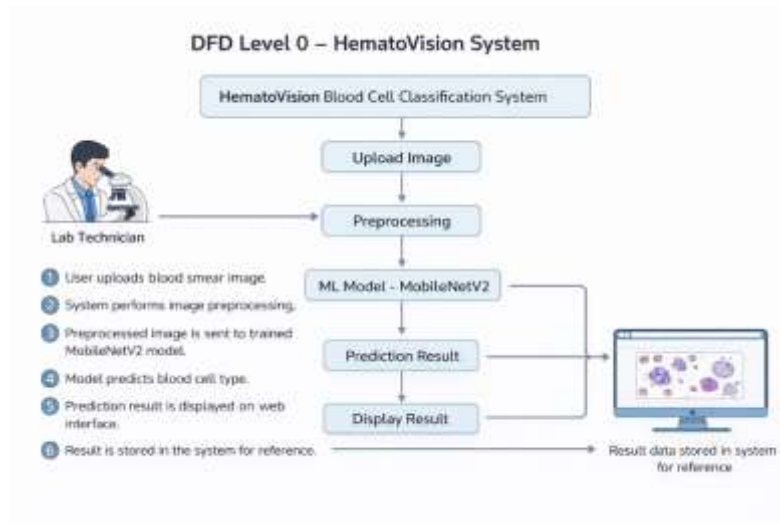
2 Application Layer

- Backend developed using Python and Flask.
- Handles image upload and validation.
- Performs image preprocessing (resize, normalization).
- Sends image to trained MobileNetV2 model.
- Generates prediction results with confidence score.

3 Data Layer

- Stores trained model file (.h5).
- Stores uploaded images temporarily.
- Stores prediction logs (if required).

Reference: <https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>



Guidelines:

Include all the processes (As an application logic / Technology Block)
 Provide infrastructural demarcation (Local / Cloud)
 Indicate external interfaces (third party API's etc.)
 Indicate Data Storage components / services
 Indicate interface to machine learning models (if applicable)

Table-1 : Components & Technologies:

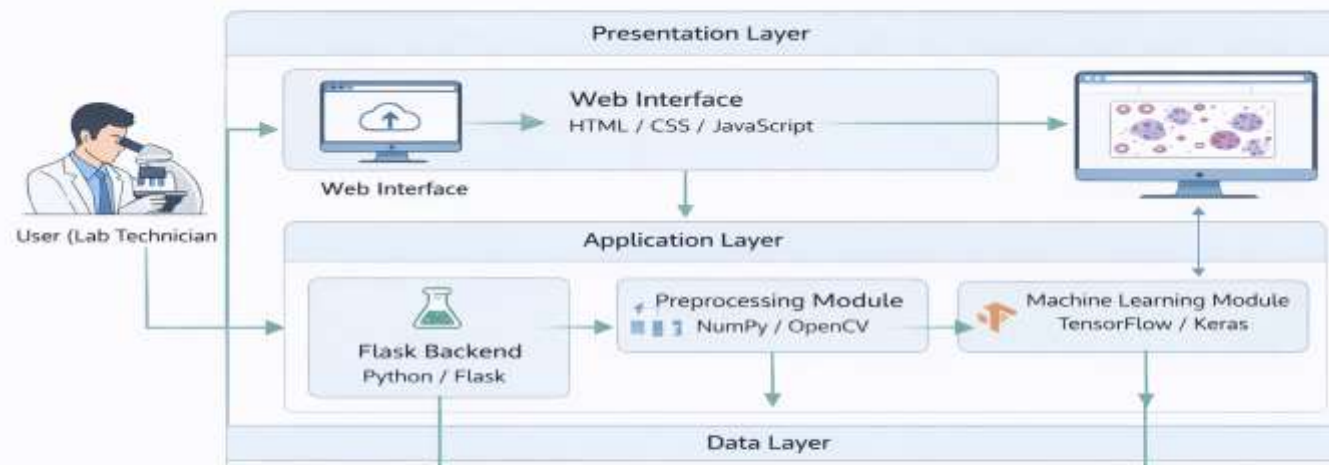
S.No	Component	Description	Technology
1.	User Interface	Web interface for uploading images and viewing results	HTML, CSS, JavaScript
2.	Application Logic-1	Backend processing and routing	Python, Flask
3.	Application Logic-2	Image preprocessing logic	NumPy, OpenCV
4.	Application Logic-3	Machine learning inference logic	TensorFlow / Keras
5.	Database	Store prediction results	SQLite / MySQL
6.	Cloud Database	Database service (if deployed on cloud)	Render Database / PostgreSQL
7.	File Storage	Store uploaded images and model files	Local File System

8.	External API-1	(Not required in this project)	N/A
9.	External API-2	(Not required in this project)	N/A
10.	Machine Learning Model	Blood cell classification using transfer learning	MobileNetV2
11.	Infrastructure (Server / Cloud)	Deployment environment	Local System / Render Cloud

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Web and ML frameworks used	Flask, TensorFlow, Keras
2.	Security Implementations	Basic input validation and secure file handling	Flask validation, HTTPS
3.	Scalable Architecture	Can handle increasing image uploads	Cloud deployment (Render)
4.	Availability	Accessible via web browser anytime	Cloud hosting
5.	Performance	Fast prediction within few seconds	Optimized ML model

HematoVision Blood Cell Classification – Architecture



References:

<https://c4model.com/>

<https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>

<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>