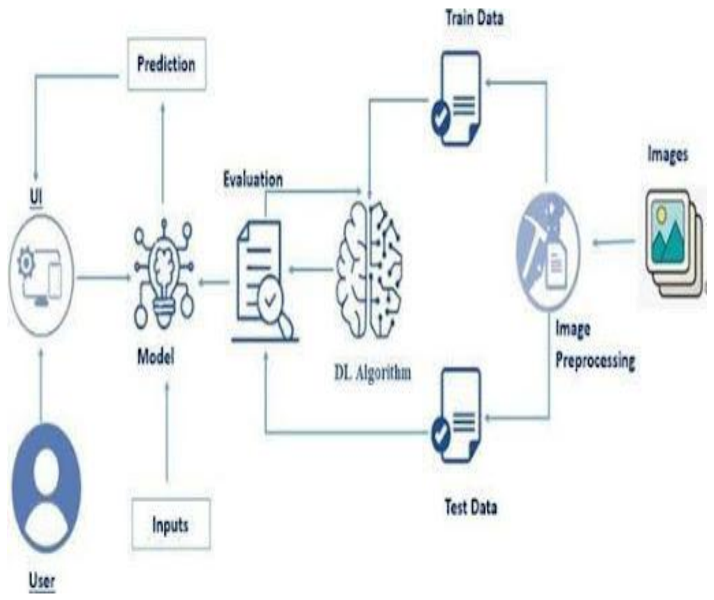
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	28 June 2025
Team ID	LTVIP2025TMID60119
Project Name	Hematovision
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



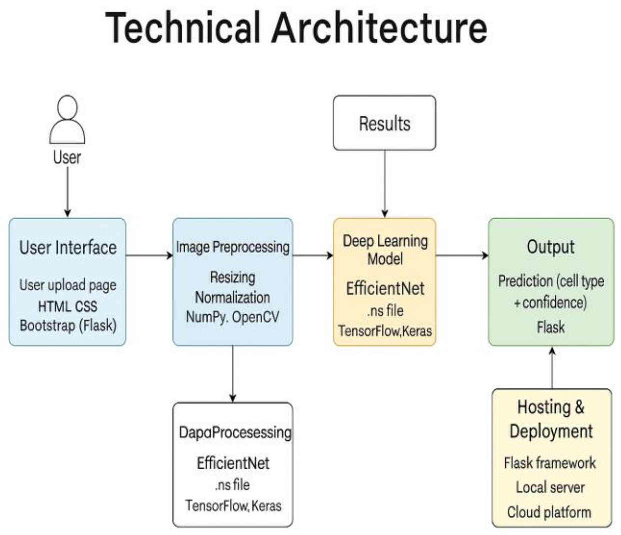


Table-1:

S.No	Component	Description	Technology
1	Data Collection	Collect 12,000 labeled blood cell images (eosinophils, lymphocytes, etc.)	Image Datasets, Kaggle, BCCD
2	Data Preprocessing	Resize, normalize, augment images before training	NumPy, OpenCV, TensorFlow ImageDataGen
3	Model Architecture	Load pre-trained CNN for feature extraction and classification	Transfer Learning, ResNet50 (Keras)
4	Model Training	Train the classifier on blood cell dataset using fine-tuning	TensorFlow, Keras
5	Prediction Module	Predict cell type with confidence score from uploaded image	NumPy, TensorFlow, Softmax
6	Flask Backend	Handles HTTP requests, file uploads, and prediction logic	Flask (Python Web Framework)
7	Frontend UI	Allows users to upload image and view prediction result	HTML, CSS, Jinja2 (Flask Templating)
8	Static File Handling	Saves uploaded images to server and serves them back to user	Flask static/ directory
9	Output Display	Shows predicted label, confidence %, and image preview	HTML, Flask response rendering
10	Platform Support	Runs locally (CMD, PyCharm) or deploys online (Streamlit/Render)	Localhost, Render, Streamlit Cloud

S. No.	Characteristic	Technology Used	Description
1	AI-Driven Classification	Transfer Learning (EfficientNet/ResNet), TensorFlow/Keras	Enables high-accuracy blood cell classification using deep learning.
2	Web-Based Interface	Flask, HTML, CSS, Bootstrap	Provides a user-friendly web portal to upload images and display results.
3	User-Friendly	Simple UI/UX Design	Designed for non-technical users like lab technicians and students.
4	Lightweight & Fast	Flask Backend, Optimized Model	Delivers predictions quickly with low latency (3–5 seconds per image).
5	Portable	Local Machine, Cloud Hosting, TensorFlow Lite	Runs on desktops, web servers, and future mobile apps.
6	Secure & Private	File Validation, Auto-deletion, Flask	Ensures user images are not stored permanently; privacy-focused.
7	Scalable	Modular Python Code, Reloadable Models	Can be expanded to include more cell types or disease detection.
8	Low Resource Requirements	EfficientNet Model, Minimal Hardware Load	Functions well on basic systems without requiring GPU.
9	Educational & Diagnostic Use	Real-Time Inference + Classification Display	Useful for both teaching hematology and real-world diagnostics.
10	Real-Time Prediction	Flask API, Pre-loaded \`.h5\` Model	Returns blood cell type and confidence score instantly.

Table-2

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