Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	27 June 2025
Team ID	LTVIP2025TMID59787
Project Name	Hematovision: Advanced Blood Cell Classification using Transfer Learning
Maximum Marks	4 Marks

Technical Architecture:

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

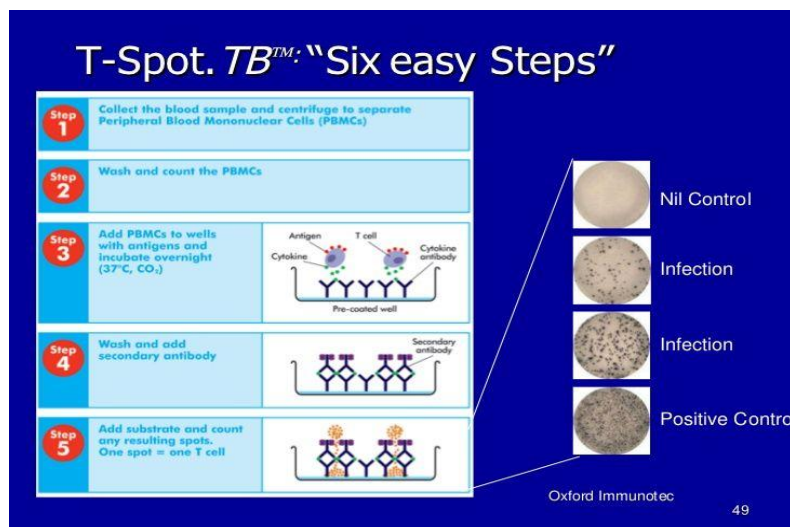**Example: Helping a Lab Technician Spot Infections Quickly**

Reference: https://in.search.yahoo.com

Table-1: Application Component:

S.No	Component	Description	Technology
1.	Image Capture & Upload	Microscope captures blood smear images and sends them to the system.	- Digital Microscope - Web upload
2.	Frontend (User Interface)	Interface for the lab technician to upload images and view results.	- React.js or Vue.js - Tailwind CSS
3.	Backend API	Handles requests from the frontend	- FastAPI or Flask (Python)

4.	Image Preprocessing	Prepares images (resize, normalize, denoise) for classification	OpenCV - Pillow (PIL) - NumPy
5.	Deep Learning Model	Classifies blood cells using a fine-tuned transfer learning model.	- PyTorch or TensorFlow - EfficientNet or ResNet pretrained on ImageNet
6.	Model Serving	Exposes the trained model to handle real-time predictions.	- TorchServe or TensorFlow Serving - ONNX Runtime
7.	Cell Detection & Annotation	Highlights and labels each blood cell in the image.	- YOLOv5/YOLOv8 (for detection) - Matplotlib / OpenCV for overlay
8.	Report Generation	Creates a summary of the cell count and any alerts for abnormal values.	Jinja2 (Python template engine) - WeasyPrint (PDF generation)
9.	Database	Stores image metadata, classification results, user actions, and reports.	PostgreSQL or MongoDB
10.	Authentication & Security	Manages user roles (e.g., technician, admin), encrypts data.	- JWT / OAuth2 for login - HTTPS (SSL/TLS)

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Real-time, High-Resolution	Captures high-quality images of blood smear slides via microscope.	- Digital Microscope - USB/HDMI interface
2.	Lightweight, Fast	Connects frontend with AI model and manages data flow.	- FastAPI or Flask (Python)
3.	Automated, Accurate	Enhances image quality, normalizes formats for ML model input.	-OpenCV - Pillow
4.	Transfer Learning, Fine-Tuned	Classifies cell types using pretrained model adapted to blood cell images.	-PyTorch or TensorFlow - EfficientNet, ResNet

References:

https://proceedings.mlr.press/

https://in.search.yahoo.com/