**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 31 January 3035 |
| Team ID | LTVIP2025TMID35810 |
| Project Name | Hemotovision |
| Maximum Marks | 4 Marks |

**🧱 Technical Architecture Diagram**

📌 The diagram will illustrate the following flow:

* User uploads blood smear image via **Web UI**
* Backend server (Flask) processes and routes data
* Image is passed to **CNN-based ML model (ResNet50)**
* Prediction and confidence score are returned to the UI
* Results may be saved locally or optionally to a cloud DB



**Table-1 : Components & Technologies:**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
| 1 | **User Interface** | Web interface for uploading images and viewing classification results | HTML, CSS, Bootstrap |
| 2 | **Application Logic-1** | Handles routing, image upload, error validation | Python, Flask |
| 3 | **Application Logic-2** | Handles image preprocessing and model inference | TensorFlow, NumPy, OpenCV |
| 4 | **Application Logic-3** | Logic for showing results, confidence scoring, and UI updates | Flask + Jinja2 Template Engine |
| 5 | **Database** | (Optional) Store prediction logs and metadata | SQLite / CSV File Logs |
| 6 | **Cloud Database** | (Optional enhancement) for cloud storage of patient/classification data | Firebase Realtime DB / AWS S3 |
| 7 | **File Storage** | Store uploaded images temporarily | Local Filesystem |
| 8 | **External API-1** | Not used in v1.0; optionally integrate with lab info system | (N/A) |
| 9 | **External API-2** | Not used in current version | (N/A) |
| 10 | **Machine Learning Model** | Used to classify 4 types of white blood cells | ResNet50 Transfer Learning Model |
| 11 | **Infrastructure** | Local system (for testing), scalable to cloud with Docker/Heroku | Localhost / Cloud (AWS/Heroku) |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology / Tools** |
| --- | --- | --- | --- |
| 1 | **Open-Source Frameworks** | Flask (Python), TensorFlow/Keras, Bootstrap | All open-source |
| 2 | **Security Implementations** | Input validation, file type filtering, basic HTTPS support in cloud setups | Flask-WTF, file extension filters |
| 3 | **Scalable Architecture** | Supports 3-tier architecture (UI → Server → ML Model) | Flask server + Model containerization |
| 4 | **Availability** | Can be deployed on multiple nodes/cloud; static files via CDN | AWS, Heroku, GitHub Pages (UI only) |
| 5 | **Performance** | Lightweight model, optimized image size, average prediction ~0.4s | ResNet50, Numpy, Local Caching |

**REFERENCES:**

📄 [Download HematoVision\_ProjectDesignPhaseII.docx](Download%20HematoVision_ProjectDesignPhaseII.docx)

It includes:

* Architecture image
* Tables for Components, Technologies, and Characteristics
* Proper references