**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 31 January 3035 |
| Team ID | LTVIP2025TMID35810 |
| 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: Order processing during pandemics for offline mode**

**Reference:** [**https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/**](https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/)



**1. Frontend**

* **Framework: Streamlit (for web-based user interface)**
* **Alternative Options: React.js / Angular (for advanced dashboards)**
* **Features:**
  + **Image upload interface**
  + **Result visualization (charts, predictions)**
  + **Report download option**

**2. Backend**

* **Language: Python**
* **Frameworks: Flask or FastAPI (for REST API endpoints)**
* **Features:**
  + **Handles user requests**
  + **Communicates with ML model**
  + **Generates reports**

**3. Machine Learning / AI**

* **Libraries:**
  + **TensorFlow or PyTorch (for deep learning models)**
  + **OpenCV (image preprocessing)**
  + **Scikit-learn (metrics, evaluation)**
* **Models:**
  + **Pre-trained CNN models for transfer learning (ResNet, EfficientNet, InceptionV3, VGG16)**
* **Workflow:**
  + **Data preprocessing → Feature extraction → Fine-tuning → Prediction**

**4. Database**

* **Options:**
  + **PostgreSQL or MySQL (structured data like user profiles, logs)**
  + **MongoDB (for unstructured data, optional)**

**5. Cloud & Deployment**

* **Cloud Providers: AWS / GCP / Azure**
* **Deployment:**
  + **Docker (containerization)**
  + **Kubernetes (scaling, optional)**
  + **Streamlit Cloud or Heroku for rapid deployment**

**6. Storage**

* **Image Storage: AWS S3 / Google Cloud Storage**
* **Model Storage: TensorFlow Serving / TorchServe**

**7. Security**

* **Data Encryption: HTTPS, SSL Certificates**
* **Authentication: OAuth 2.0 / JWT**