**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

| Date | 20June 2025 |
| --- | --- |
| Team ID | LTVIP2025TMID39799 |
| Project Name | Clean Tech: Transforming waste into 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: HematoVision: Transforming Waste Management with Transfer Learning**

**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/)





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

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | User Interface | Web interface for uploading images and viewing reports | HTML, CSS, JavaScript/ React Js etc. |
|  | Application Logic-1 | Handles image upload, validation, and preprocessing | Python |
|  | Application Logic-2 | Model inference and classification logic | TensorFlow, Keras |
|  | Application Logic-3 | Diagnostic report generation and feedback loop | Python, Pandas, ReportLab |
|  | Database | Stores user data and report logs | PostgreSQL or Firebase Realtime DB |
|  | Cloud Database | Cloud-hosted structured database | IBM DB2, IBM Cloudant etc. |
|  | File Storage | Storage for uploaded blood cell images | IBM Block Storage or Other Storage Service or Local Filesystem |
|  | External API-1 | Optional: Email notification services | IBM Weather API, etc. |
| 9. | Machine Learning Model | Pretrained CNN model for blood cell classification | Object Recognition Model, etc. |
| 10. | nfrastructure (Server / Cloud) | Deployment environment | Local, Cloud Foundry, Kubernetes, etc. |
|  |  |  | , . |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Frameworks used for web, backend, and ML modeling | Flask, React, TensorFlow, Keras |
|  | Security Implementations | Securing user uploads and data access with authentication | SHA-256, JWT, HTTPS, IAM Controls |
|  | Scalable Architecture | Microservices + cloud deployment enables easy horizontal scaling | Docker, Kubernetes, AWS Lambda |
|  | Availability | Ensured by using cloud storage and distributed deployment | Docker, Kubernetes, AWS Lambda |
|  | Performance | Fast response enabled by model caching and asynchronous requests | Redis, Celery, CDN for static assets |

**References:**

[**https://c4model.com/**](https://c4model.com/)

[**https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/**](https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/)

[**https://www.ibm.com/cloud/architecture**](https://www.ibm.com/cloud/architecture)

[**https://aws.amazon.com/architecture**](https://aws.amazon.com/architecture)

[**https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d**](https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d)