**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 27 June 3035 |
| Team ID | LTVIP2025TMID36584 |
| Project Name | cleantech: transforming waste management with 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: cleantech: transforming waste management with transfer learning**

* **User Interface:** User interacts with the web application by uploading images.
* **Backend:** Flask backend receives and preprocesses the uploaded images.
* **Model Prediction:** The trained deep learning model (CNN based on MobileNetV2) classifies the image.
* **Result Display:** The classification result is returned to the frontend and displayed to the user.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web UI for uploading waste images and displaying results.. | HTML, CSS |
|  | Application Logic-1 | Receives and preprocesses image, returns result to frontend. | Python, Flask |
|  | Machine Learning Model | Classifies images into Biodegradable, Recyclable, and Trash. | TensorFlow, Keras, MobileNetV2 |
|  | Infrastructure (Server / Cloud) | Application deployment for web-based access. | Local, Cloud Foundry, Kubernetes (Future: Firebase + Google Cloud Run) |
|  | Data Storage (Logs) | Stores prediction results. | Local Filesystem (for terminal logs). |

**Table-2: Application Characteristics:**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Libraries used for model development and application. | TensorFlow, Keras, Flask, OpenCV, NumPy |
|  | Security Implementations | Measures to secure image handling and data. | To be implemented based on specific requirements, e.g., HTTPS for web communication |
|  | Scalable Architecture | The web-based architecture allows for potential scaling. | Micro-services (Future consideration for deployment via Firebase + Google Cloud Run) |
|  | Availability | The web application ensures accessibility for users. | Future consideration: Load balancers for distributed servers if deployed at scale |
|  | Performance | Designed for fast image classification. | Prediction speed: -0.2 seconds/image on CPU. |