**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 15 February 2025 |
| Team ID | LTVIP2025TMID45617 |
| Project Name | CleanTech: Transforming Waste Management with Transfer Learning |
| Maximum Marks | 4 Marks |

**Solution Architecture:**

### **Solution Architecture Overview**

Solution architecture serves as the bridge between the real-world waste management problem and the AI-based technical solution. It outlines how the project is structured technically, ensuring that all components work together efficiently to meet business and user needs.

### **Goals of the Solution Architecture**

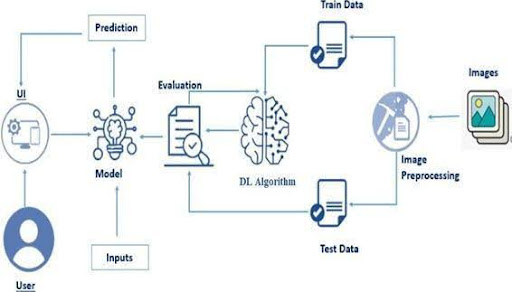
* Identify the best AI/ML solution (transfer learning) to improve waste classification.
* Clearly define components such as the user interface, backend model, and database.
* Break down development phases including training, testing, deployment, and user access.
* Provide technical specifications and data flow for building and managing the solution.

### **Solution Architecture Description**

Key Components:

1. Frontend (Web Interface)
   * Built using HTML/CSS/JavaScript or Streamlit (if using Python).
   * Allows users to upload images and view classification results.
2. Backend (Model Server)
   * Flask or FastAPI-based API server.
   * Hosts the waste classification model using a pre-trained CNN (e.g., VGG16, ResNet).
   * Accepts uploaded images and returns the predicted waste class.
3. Model (Transfer Learning)
   * Transfer learning applied using a pre-trained model (like VGG16 or MobileNet).
   * Fine-tuned on a custom dataset with three classes: biodegradable, recyclable, and trash.
   * High accuracy due to smaller training time and better generalization.
4. Database (Optional)
   * Stores image logs, predictions, and metadata.
   * Can be implemented with SQLite or Firebase.
5. Visualization Dashboard
   * Displays class-wise counts, accuracy, confusion matrix, etc.
   * Tools: Power BI, Tableau, or Matplotlib/Seaborn for basic reporting.
6. Deployment
   * Hosted on a cloud platform (e.g., Heroku, AWS EC2, or Google Cloud).
   * API endpoints integrated into the frontend for real-time classification.

### **Solution Architecture Diagram**



Development Phases

| **Phase** | **Description** |
| --- | --- |
| Phase 1: Data Setup | Collect and label image dataset. |
| Phase 2: Model Building | Implement transfer learning, train and validate the model. |
| Phase 3: API & UI | Build API for prediction and frontend interface. |
| Phase 4: Deployment | Host application and dashboard on a cloud platform. |
| Phase 5: Testing & Reporting | UAT, dashboard creation, and performance reporting. |