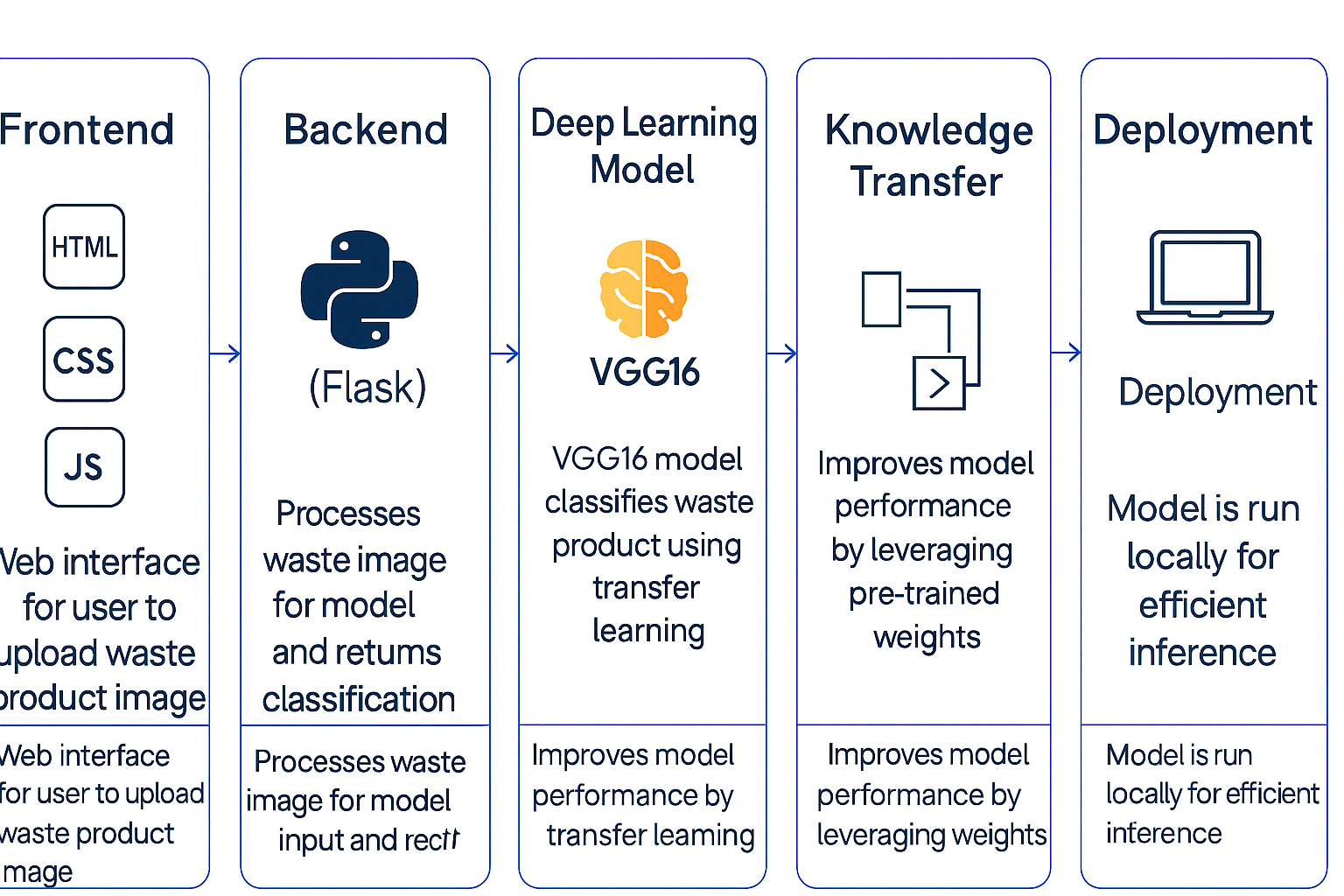
**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 6 June 2025 |
| Team ID | LTVIP2025TMID38840 |
| Project Name | CleanTech: Transforming Waste Management with Transfer Learning |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The CleanTech waste classification system is built using a full-stack approach combining web development and deep learning. The frontend is developed using HTML, CSS, and JavaScript to provide a user-friendly interface for uploading waste images. The backend is powered by Python with Flask, which handles image processing and routes data to the deep learning model. The system uses the VGG16 architecture with transfer learning, implemented using TensorFlow and Keras, to classify waste into categories such as recyclable, organic, and hazardous. The model is run locally for efficient and secure inference without relying on cloud services.



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

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
|  | User Interface | Web interface for clinicians to input data and view predictions | HTML, CSS, JavaScript |
|  | Application Logic-1 | |  | | --- | |  |  |  | | --- | | Image preprocessing and resizing for deep learning model input | | Python (OpenCV, NumPy) |
|  | Application Logic-2 | |  | | --- | |  |  |  | | --- | | Model loading, prediction, and result formatting | | Python (TensorFlow/Keras) |
|  | Application Logic-3 | Flask-based integration to link frontend and backend | Python (Flask) |
|  | File Storage | Stores pre-trained models and temporary image uploads | Local Filesystem |
|  | Deep Learning Model | Classifies waste type using VGG16 and transfer learning | VGG16 (Transfer Learning – Keras) |
|  | Infrastructure (Server / Cloud) | Application runs locally using Flask; can be extended to cloud | Local (Flask server) |

**Table-2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology** |
| --- | --- | --- | --- |
|  | Open-Source Frameworks | Utilizes open-source tools for deep learning and web development | Flask, TensorFlow/Keras, OpenCV |
|  | Security Implementations | Application runs locally; user data and images are processed offline | Local-only use; no cloud auth |
|  | Scalable Architecture | Can be modularized for future deployment on cloud or microservices | Flask + Modular Python files |
|  | Performance | Fast predictions with preloaded VGG16 model and efficient preprocessing | Flask, NumPy, Keras |