**Project Design Phase-II**

**Technology Stack (Architecture & Stack)**

|  |  |
| --- | --- |
| Date | 31 January 2025 |
| Team ID | LTVIP2025TMID36223 |
| Project Name | GRAINPALETTE |
| Maximum Marks | 4 Marks |

**Technical Architecture:**

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

**GrainPalette  
Table-1: Application Components**

| **S.No** | **Component** | **Description** | **Technology** |
| --- | --- | --- | --- |
| **1** | **User Interface** | **Web UI to upload rice image and display prediction result** | **HTML, CSS, JavaScript** |
| **2** | **Application Logic-1** | **Core backend logic including file handling and prediction routing** | **Python, Flask** |
| **3** | **Application Logic-2** | **Image preprocessing and integration with trained ML model** | **OpenCV, NumPy, TensorFlow** |
| **4** | **Application Logic-3** | **Model architecture using pre-trained MobileNetV2 for feature extraction** | **TensorFlow Keras** |
| **5** | **Database** | **Not required for this version (can be added for user tracking/history)** | **—** |
| **6** | **Cloud Database** | **Not used currently (could use Firebase/Cloudant in future for user logs)** | **—** |
| **7** | **File Storage** | **Stores uploaded rice images temporarily** | **Local File System** |
| **8** | **External API-1** | **Not used in current version** | **—** |
| **9** | **External API-2** | **Not used in current version** | **—** |
| **10** | **Machine Learning Model** | **Deep learning model classifies rice type using image input** | **MobileNetV2, TensorFlow** |
| **11** | **Infrastructure** | **Runs locally during development; can be deployed to cloud like AWS/Render** | **Localhost (Flask), Cloud-ready deployment** |

**Table-2: Application Characteristics**

| **S.No** | **Characteristics** | **Description** | **Technology Used** |
| --- | --- | --- | --- |
| **1** | **Open-Source Frameworks** | **Web & ML Frameworks** | **Flask, TensorFlow, Keras, OpenCV** |
| **2** | **Security Implementations** | **Basic input validations for file upload; no auth used in current version** | **Flask File Security, MIME filtering** |
| **3** | **Scalable Architecture** | **Can be adapted to microservices (UI, model, logic separated); ML model can be containerized** | **Flask microservice, Docker** |
| **4** | **Availability** | **Can be deployed with load balancer support and failover services** | **Render, AWS EC2, Azure App Service** |
| **5** | **Performance** | **Image preprocessed before prediction; uses pre-trained MobileNetV2 for fast inference** | **TensorFlow Lite (future), OpenCV, CDN (optional)** |