**Project Design Phase**

**Solution Architecture**

|  |  |
| --- | --- |
| Date | 28 June 2025 |
| Team ID | LTVIP2025TMID40163 |
| Project Name | grainpalette - a deep learning odyssey in rice type classification through transfer learning |
| Maximum Marks | 4 Marks |

## 🧠 ****Goal****

Bridge the gap between manual rice variety identification and fast, accurate, AI-powered classification using a lightweight, scalable deep learning model.

## ⚙️ ****Core Components****

* **🔍 Model:**  
  Transfer learning using **ResNet50** (or **MobileNetV2**) trained to classify **5 rice types** – Basmati, Arborio, Ipsala, Jasmine, and Karacadag.
* **💻 Frontend:**  
  HTML/CSS-based responsive web interface for users to **upload rice grain images** and view predictions.
* **🔁 Backend:**  
  **Flask-based Python server** to handle image input, run model inference, and return prediction results with confidence scores.
* **🚀 Deployment:**  
  Designed to work **locally or on lightweight cloud services** like Replit or PythonAnywhere. Easy to deploy without GPU.

## 🔄 ****System Flow****

1. 📸 User uploads or drags-and-drops an image of rice grains via the web interface.
2. 🧼 Image is resized and preprocessed to fit model input shape.
3. 🧠 Model predicts the rice variety + returns confidence %.
4. 📊 Result is displayed instantly in a clean UI with image preview and class name.

## 🧩 ****Development Phases****

* **📦 Data Collection & Preprocessing:**  
  Image gathering, resizing, normalization, and labeling.
* **📈 Model Training & Optimization:**  
  Using transfer learning on ResNet/MobileNet with frozen base layers + dense classifier layers.
* **🎨 UI Design:**  
  Responsive, device-friendly, colorful UI with visual feedback.
* **🧪 Testing & Deployment:**  
  Local testing > Flask setup > GitHub version control > Web deployment (e.g., Replit/Render).

## ✅ ****Technical & Business Requirements****

| **Category** | **Requirement** |
| --- | --- |
| 🎯 Usability | Must work smoothly on basic devices (no GPU). |
| 🌐 Accessibility | Designed for both rural and urban users. |
| 📶 Connectivity | Should support offline inference once model is loaded. |
| 📈 Scalability | Easily extendable to classify other grains (wheat, maize, etc.). |
| 💼 Business | Open-source for students; monetizable via SaaS/web integrations for farms. |

****