**Phase-2:Requirement Analysis**

**Technical Requirements:**

**Hardware:**

* A computer with at least **8 GB RAM** (16 GB recommended for faster training)
* **GPU support** (optional, for faster model training if working with large datasets)

**Software & Tools:**

* **Anaconda Navigator** for environment and package management
* **Visual Studio Code (VS Code)** and/or **Spyder** for code development
* **Python 3.7+**
* **Required Python libraries**:
  + numpy
  + pandas
  + tensorflow==2.3.2
  + keras==2.3.1
  + Flask
* **Web browser** for accessing the Flask-based web application interface

**Online Resources:**

* Tutorials and guides on **CNN**, **MobileNet**, and **Flask** for foundational understanding
* Pretrained MobileNetV4 model for transfer learning

**Functional Requirements:**

**Image Upload and Processing**

* Users should be able to upload images of rice grains through the web interface.
* Images must be preprocessed (resizing, normalization) before being passed to the AI model.

**Rice Variety Prediction**

* The trained CNN with MobileNetV4 transfer learning should classify rice images into **five predefined rice types**.

**Display Prediction Results**

* The web application should display the predicted rice type clearly to the user.

**Accuracy Reporting (Optional for Users)**

* The backend should calculate and log the model’s prediction accuracy during testing.

**Web Application Deployment**

* A functional, user-friendly web interface built with **Flask**, enabling easy interaction with the AI model.

## Constraints and Challenges:

**Dataset Limitations**

* Availability of a sufficiently large and balanced dataset of high-quality rice grain images is critical.
* Data imbalance between rice types could lead to biased predictions.

**Hardware Limitations**

* Training deep learning models, especially CNNs, is resource-intensive.
* Lack of GPU or insufficient RAM could increase training time significantly.

**Model Generalization**

* The AI model might struggle with unseen rice types, low-quality images, or images taken in varying lighting conditions.

**Software Version Compatibility**

* Compatibility issues might arise due to the use of specific package versions (e.g. TensorFlow 2.3.2 and Keras 2.3.1).

**Web Deployment**

* Deploying AI models in lightweight web applications like Flask requires careful optimization to handle image uploads, predictions, and display results in real time.

**User Accessibility**

* Ensuring the web interface is intuitive and accessible to users with minimal technical skills (especially farmers and home growers) is essential.