**Phase-6: Functional and Performance Testing**

**I : Test Cases Executed:**

| **Test Case ID** | **Test Description** | **Input** | **Expected Output** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- |
| TC-01 | Check if web app loads successfully | URL: localhost:5000/ | Home page loads without error | Home page displayed successfully | Pass |
| TC-02 | Image upload functionality test | Valid rice grain image (jpg/png) | Image uploads successfully and moves to server directory | Upload successful, file saved | Pass |
| TC-03 | Image preprocessing verification | Uploaded image | Image resized to 224×224, normalized | Image preprocessing done correctly | Pass |
| TC-04 | Model prediction accuracy test | Known test image (rice type A) | Correct rice variety prediction | Correct prediction returned | Pass |
| TC-05 | Invalid image format upload | Upload a .txt file | Error message displayed | Proper validation and error handled | Pass |
| TC-06 | Large image file handling | Upload a 5MB+ rice image | Image accepted, processed without crashing | Handled successfully | Pass |
| TC-07 | Multiple consecutive predictions | Upload multiple images one after another | Each prediction handled correctly, results displayed | No crashes, accurate predictions | Pass |
| TC-08 | UI responsiveness test | Access web app on mobile browser | UI adapts and works without layout issues | UI responsive and accessible | Pass |
| TC-09 | Invalid file path access | Access non-existing route /predict123 | Custom 404 page or error message | Error message shown | Pass |
| TC-10 | Model file loading test | Start Flask server | Model loads without errors | Model loaded successfully | Pass |

**II : Bug Fixes and Improvements:**

**Bug Fixes:**

| **Bug Description** | **Cause** | **Fix/Resolution** |
| --- | --- | --- |
| **Image not resizing correctly during prediction** | Incorrect input shape expected by MobileNetV4 | Added a dedicated image preprocessing function to resize images to **224×224** and normalize pixel values before prediction |
| **Flask app crash on uploading unsupported file formats (.txt, .pdf)** | No file type validation during upload | Added file type validation using Flask’s allowed\_extensions check to restrict uploads to **.jpg, .jpeg, .png** |
| **Model loading error when restarting Flask app** | Incorrect file path to saved .h5 model | Fixed by using absolute/relative path properly and verifying the correct model filename in Flask backend |
| **Web page layout breaking on mobile devices** | Missing responsive CSS rules | Applied **CSS media queries** and simplified layout structure for mobile screens |
| **Incorrect predictions on low-light or blurry images** | Model overfitting to high-quality, ideal dataset images | Improved by applying **data augmentation** during training: rotation, zoom, brightness adjustments |

**Improvements Made:**

| **Improvement** | **Reason** | **Benefit** |
| --- | --- | --- |
| Added **data augmentation** during model training | To improve model generalization on real-world images | Increased model robustness and accuracy on varied images |
| Implemented **simple, clean UI design with mobile responsiveness** | Original UI was cluttered and non-responsive | Improved usability for farmers, students, and mobile users |
| Integrated a **prediction log feature (optional)** | To track model predictions for future analysis | Helps in monitoring AI decisions and retraining needs |
| Optimized model by freezing **base layers in MobileNetV4** | To reduce training time on CPU-only systems | Faster model training without significant loss in accuracy |
| Created **incremental project documentation templates** | To avoid end-stage reporting delays | Organized and time-efficient documentation process |

**III : Final Validation:**

**Project Successfully Meets All Initial Requirements**

**-**AI-based rice variety prediction model works reliably

-Web application is functional, responsive, and user-friendly

-All planned features were implemented, tested, and validated