**✅ Functional and Performance Testing – HematoVision**

|  |  |
| --- | --- |
| **Date** | **21 February 2025** |
| **Team ID** | **LTVIP2025TMID35810** |
| **Project Name** | **Hemotovision** |
| **Maximum Marks** | **10** |

**🔧 1. Functional Testing Overview**

Functional testing ensures that every feature in the HematoVision system works as intended. The goal was to validate user inputs, application responses, model prediction flow, and result presentation in a controlled environment.

**📋 Tested Functional Areas:**

| **Feature/Function** | **Description** | **Status** |
| --- | --- | --- |
| **Image Upload** | Allows users to upload .jpg, .jpeg, or .png microscope images | ✅ Pass |
| **Invalid File Handling** | Rejects unsupported files (e.g., .txt, .pdf, .exe) with clear message | ✅ Pass |
| **Model Prediction** | Successfully classifies the uploaded image into one of four blood cell types | ✅ Pass |
| **Confidence Score Display** | Shows softmax-based probability score alongside prediction | ✅ Pass |
| **User Interface** | UI is simple, responsive, and provides visual feedback after upload | ✅ Pass |
| **Prediction Result Page** | Clean layout showing class name and confidence | ✅ Pass |
| **Error Messages** | Shows proper messages for blank upload, incorrect file, or system crash | ✅ Pass |
| **Cross-Browser Compatibility** | Tested on Chrome, Firefox, and Edge | ✅ Pass |
| **Mobile Responsiveness** | Successfully used on Android phone browser | ✅ Pass |
| **Reset and Re-Upload Functionality** | Users can upload a new image after viewing previous results | ✅ Pass |

**🚀 2. Performance Testing Overview**

Performance testing evaluates the system's **speed, stability, and accuracy** under different conditions. It ensures that HematoVision performs well not just in ideal setups but also in practical, day-to-day usage.

**📊 Key Performance Metrics:**

| **Metric** | **Result** | **Notes** |
| --- | --- | --- |
| **Model Accuracy** | 94.6% | Tested on 20% hold-out test data |
| **Prediction Time** | ~0.4 seconds per image | Tested on i5 CPU with 8GB RAM |
| **Memory Usage** | ~200–300 MB | Includes Flask app and model in RAM |
| **Model Size** | ~100 MB .h5 file | Easy to deploy and load |
| **Confidence Score Range** | 0.87–0.99 (in most cases) | High confidence, minimal false positives |
| **False Positive Rate** | <5% | Mainly due to overlapping features in cells |
| **Flask API Response Time** | ~350–500 ms | Fast enough for real-time single image use |
| **Uptime in Local Testing** | 100% | No crashes in continuous 3-hour test cycle |

**🧪 3. Testing Environment:**

* **Hardware**: Intel Core i5, 8GB RAM, Windows 10
* **Software**:
  + Python 3.8
  + Flask 2.x
  + TensorFlow 2.x
  + OpenCV, Pillow
* **Testing Tools**:
  + Manual UI testing
  + Postman for Flask API testing
  + Browser dev tools for performance analysis

**🛠 4. Recommendations for Further Testing**

* Integrate automated unit testing for each module
* Add logging and monitoring for backend inference times
* Perform stress testing with multiple images concurrently
* Deploy model in a container (Docker) and evaluate in cloud environments
* Compare performance with other CNNs (e.g., MobileNet, EfficientNet)

## Test Scenarios & Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Scenario (What to test)** | **Test Steps (How to test)** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| **FT-01** | Text Input Validation (e.g., topic, job title) | Enter valid and invalid text in input fields | Valid inputs accepted, errors for invalid inputs | Work As expected | Pass |
| **FT-02** | Number Input Validation (e.g., word count, size, rooms) | Enter numbers within and outside the valid range | Accepts valid values, shows error for out-of-range | Accepts correctly | PASS |
| **FT-03** | Content Generation (e.g., blog, resume, design idea) | Provide complete inputs and click "Generate" | Correct content is generated based on input | Functions correctly | Pass |
| **FT-04** | API Connection Check | Check if API key is correct and model responds | API responds successfully | Predicted correctly | Pass |
| **PT-01** | Response Time Test | Use a timer to check content generation time | Should be under 3 seconds | Succesful responce | Pass |
| **PT-02** | API Speed Test | Send multiple API calls at the same time | API should not slow down | Avg 2.4 sec | PAss |
| **PT-03** | File Upload Load Test (e.g., PDFs) | Upload multiple PDFs and check processing | Should work smoothly without crashing | Handled 100 requests/sec | Pass |