Functional & Performance Testing Template

Model Performance Test

Date	29 june2025
Team ID	LTVIP2025TMID59847
Project Name	Hematovision: Advanced Blood Cell Classification Using Transfer Learning
Maximum Marks	4 Marks

Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
HV-01	Image Upload Functionality	Upload a blood smear image in the UI	Image should upload without error and preview correctly		
HV-02	Input Image Validation	Try uploading non-image files (e.g., .txt, .docx) or corrupted images	App should reject unsupported or invalid files with error message		
HV-03	Blood Cell Classification Accuracy	Upload clear images of lymphocyte, neutrophil, etc. and check predictions	Model should return correct blood cell type with high confidence		
HV-04	Transfer Learning Model Loading	Start the app and observe if the pre-trained model loads without issues	Model should load into memory and be ready for predictions		
HV-05	Performance on Low-Resolution Images	Upload blurry or low-res blood smear images	System should still provide prediction or show warning if unusable		
HV-06	Web Interface Responsiveness	Test interface on desktop, tablet, and mobile	UI should adapt to different screen sizes and remain usable		

HV-07	Multiple File Upload	Upload several images simultaneously	System should accept and process all valid image files		
HV-08	Error Handling on Prediction Failures	Simulate a crash during prediction (e.g., break model loading temporarily)	Error message should display instead of crashing the app		
HV-09	Accuracy Report Generation (if available)	Check if accuracy/confusion matrix or logs are downloadable after predictions	Report should generate and download correctly		
HV-10	Flask API Endpoint Functionality	Send test POST request with image to the prediction endpoint	JSON response should include correct class label and probability score		

Performance Testing Scenarios

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
PT-01	Disease Prediction Response Time	Use a stopwatch or log time taken for prediction after symptom input	Should respond in **under 3 seconds**		
PT-02	Chat API Load Test	Simulate 10+ users chatting with AI simultaneously	Chatbot remains responsive with no timeouts		
PT-03	Vitals Data Upload Load Test	Upload multiple vital logs (e.g., 50 records from Excel or sensors)	Upload and visualization should work without lag		
PT-04	Dashboard Load Performance	Open dashboard with multiple charts and recent health records	Loads within 2–3 seconds on stable network		
PT-05	Backend Model Throughput	Send 100 requests per minute to disease predictor API	No crashes, responses maintained under 5s		

PT-06	File Upload Resilience	Upload multiple reports (PDF, images) at once	All uploads complete without crashing the application		