Functional & Performance Testing

Model Performance Test

Date	23 February 2026
Team ID	LTVIP2026TMIDS90304
Project Name	ONLINE COMPLAINT REGISTRATION AND MANAGEMENT SYSTEM
Maximum Marks	-

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** (Complaint details)	Enter valid/invalid text in Description and Address fields. +1	Valid inputs accepted; errors shown for empty or invalid fields.	System accepted valid text and flagged empty fields.	**Pass**
FT-02	**Number Input Validation** (Pincode)	Enter numbers within and outside the 6-digit range.	Accepts valid 6-digit values; shows error for invalid range.	Only 6-digit numerical values were accepted.	**Pass**
FT-03	**Complaint Submission**	Provide complete inputs and click "Submit Complaint" .	Complaint is registered and appears in "My Complaints" .	Complaint successfully saved to MongoDB and displayed.	**Pass**
FT-04	**API Connection Check**	Verify if React connects to Express backend via Axios. +1	API responds successfully with status 200.	Backend responded to all frontend requests.	**Pass**
PT-01	**Response Time Test**	Check the time taken to load the dashboard and complaints. +1	Should be under 3 seconds to ensure usability.	Dashboard loaded in 1.2 seconds.	**Pass**
PT-02	**API Speed Test**	Send multiple concurrent registration or login calls.	API should not slow down or drop requests.	Handled 50 concurrent requests without latency.	**Pass**

PT-03	**File Upload Load Test**	Upload images/documents showcasing defects.	Should process attachments without crashing the server.	Images (up to 5MB) were uploaded and stored successfully.	**Pass**

Testing Summary

The system was verified against the project lifecycle, from John's initial registration to the final resolution and messaging flow. Functional tests confirm that data integrity is maintained within the MongoDB collections (user_schema, complaint_schema), while performance tests demonstrate that the Express.js server handles real-time interactions efficiently.