Functional & Performance Testing Template

Model Performance Test

Date	08 February 2026
Team ID	LTVIP2026TMIDS73975
Project Name	Intelligent SQL Querying with LLMs Using Gemini Pro
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
FT-01	Text Input Validation	Enter valid and invalid natural language queries	Valid text accepted, empty input rejected with error message	Valid queries processed, empty input shows warning	Pass
FT-02	SQL Injection Prevention	Enter malicious input (e.g., DROP TABLE)	System prevents unsafe execution	Unsafe queries blocked, error shown	Pass
FT-03	Natural Language to SQL Generation	Enter sample queries (SELECT, WHERE, GROUP BY, JOIN)	Correct SQL query generated	92% accurate SQL generation	Pass
FT-04	API Connection Check	Verify Gemini API key and send test request	API responds successfully	API responded within 2 seconds	Pass
FT-05	Query Execution	Execute generated SQL on SQLite database	Correct data returned	Results displayed correctly in tabular format	Pass
PT-01	Response Time Test	Measure time for query generation	Response < 5 seconds	Average response time = 2.8 seconds	Pass

PT-02	Concurrent API Calls	Send multiple queries simultaneously (5-10 requests)	System handles requests without crash	10 requests handled in ~12 seconds (~1.2 sec/query)	Pass

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
PT-03	Load Test	Execute multiple complex JOIN queries continuously	System remains stable	No crashes, minor delay on large queries	Pass
PT-04	Error Handling Stability	Trigger invalid SQL generation	System shows meaningful error message	User-friendly error message displayed	Pass

Performance Observations

- Simple SELECT queries → 100% success
- Aggregation queries → 95% accuracy
- JOIN queries → 88-92% accuracy
- Average response time → 2.8 seconds
- No system crash during testing

Security Testing Results

- SQL injection attempts blocked
- API key secured via environment variables
- Invalid query execution prevented
- Error messages do not expose sensitive system data