Functional & Performance Testing

Model Performance Test

Date	27 June 2025
Team ID	LTVIP2025TMID29288
Project Name	SmartSDLC – AI-Enhanced Software Development Lifecycle
Maximum Marks	

Test Scenarios & Results (based on SmartSDLC features)

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., prompts, requirements)	Enter valid/invalid requirements or prompts	Valid inputs accepted, errors shown for invalid formats	As expected	Pass
FT-02	PDF Upload & Extraction	Upload requirement PDF with mixed content	Extracts structured text using PyMuPDF	As expected	Pass
FT-03	AI Code Generation	Input user story and trigger Watsonx model	Generates production-quality code	As expected	Pass
FT-04	Bug Fixer Functionality	Input buggy code and receive corrected output	Returns fixed version with explanation	As expected	Pass
FT-05	Test Case Generator	Input code block or requirement and click “Generate Tests”	Outputs valid unit/pytest test cases	As expected	Pass
FT-06	Code Summarizer	Input a source code block	Returns natural language summary of function/purpose	As expected	Pass
FT-07	Floating AI Chatbot (LangChain)	Ask questions like “What is unit testing?”	Returns helpful AI-based SDLC answers	As expected	Pass

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
PT-01	Content Generation Response Time	Measure time for Watsonx to return code/test after input	Should respond in under 3 seconds	2.4 seconds	Pass
PT-02	Concurrent API Load (Parallel Requests)	Simulate 10 simultaneous user requests	No significant slowdown or failure	As expected	Pass
PT-03	PDF Upload Stress Test	Upload 5–10 large requirement PDFs in succession	Handles uploads without crash	As expected	Pass
PT-04	Multi-Feature Usage Performance	Use code generation, test generator, summarizer simultaneously	System remains stable and responsive	As expected	Pass