

# Quality Assignment Test Plan

## Testing methodology:

The testing will include:

- Ensures that all API routes are working as expected.
- Checks that database connections are working and data is correct.
- Ensures that LLM integration is working correctly and returning expected responses.
- Tests for errors and ensures that error messages are clear.

Scope of testing the testing covers the following components:

### **API routes:**

- /messages (CRUD operations for user and system messages).
- /responses (CRUD operations for LLM responses).
- /llms (listing available LLM templates).
- /prompt (calling LLM and saving responses).

### **Database:**

- Connecting to MongoDB and creating collections.
- Storing, retrieving, updating and deleting data.

### **LLM integration:**

- prompt\_llm function functioning and generating the correct response.
- Middleware: CORS settings functioning.

## Test Types

### Unit Testing :

Individual functions and methods are tested.

Components to be tested:

- prompt\_llm (src/llm\_implementation.py)
- get\_database and init\_db (src/db.py)
- CRUD functions in routes (/messages, /responses).

### Integration Testing :

Testing that the different components work together.

API routes and database: Testing that messages are stored and retrieved correctly from MongoDB.

LLM integration: Testing that LLM returns a response and it is stored in the database.

Error situations: Testing that invalid requests return the correct error messages.

Test Case ID	Description	Input	Expected Output	Testing Type
TC-01	Test /prompt endpoint with valid input	{"system_message": "Explain FastAPI", "user_message": "What is FastAPI?"}	Response with a valid response_id and response content	Unit
TC-02	Test /prompt endpoint with missing fields	{"user_message": "What is FastAPI?"}	HTTP 422 error with validation message	Robustness
TC-03	Test /messages CRUD operations	Create, read, update, and delete operations on <u>MessageRequest</u>	Messages are correctly created, retrieved, updated, and deleted	Integration
TC-04	Test database connection initialization	Call init_db()	MongoDB client and database instance are successfully initialized	Unit

<b>TC-05</b>	Test LLM response generation	Query: "What is FastAPI?", System Message: "Explain FastAPI"	Valid response generated by the LLM	Unit
<b>TC-06</b>	Test /responses endpoint for retrieving responses	Valid response_id	Response content is retrieved correctly	Integration
<b>TC-07</b>	Test /prompt endpoint with invalid LLM integration	LLM service unavailable	Graceful failure with error message	Robustness
<b>TC-08</b>	Test /prompt endpoint performance	High volume of concurrent requests	Responses are generated within acceptable time limits	Performance
<b>TC-09</b>	Test CORS middleware	API request from a different origin	Request is allowed with appropriate CORS headers	Integration
<b>TC-10</b>	Test /prompt endpoint with large input	Large <u>system_message</u> and <u>user_message</u>	Response is generated without errors	Performance

### Tools and Technologies

- **Unit Testing:** Pytest
- **Integration Testing:** Postman, Pytest
- **Performance Testing:** SonarCube and CodeScene
- **Robustness Testing:** Pytest with mocking (e.g., pytest-mock)