CHAPTER 7: TESTING

##### **7.1 TESTING PLAN**

**Objectives of Testing**

* The key objectives of the testing phase are:
* To identify and fix functional and non-functional issues before deployment.
* To ensure that the software meets client and user expectations.
* To validate system performance, security, and scalability.
* To verify integration with third-party services and APIs.
* To ensure smooth user experience and usability across different devices.

**Testing Methodologies**

To comprehensively assess the software, a combination of testing methodologies will be employed:

* **Manual Testing** – Test cases are executed manually to ensure correct behavior.
* **Automated Testing** – Scripts are developed to automate repetitive test scenarios.
* **Unit Testing** – Individual components and functions are tested to ensure they work as expected.
* **Integration Testing** – Ensures that different modules interact seamlessly.
* **System Testing** – Validates the complete software system as a whole.

**Testing Scope**

The testing plan will cover:

* **Functionality Testing**: Checking all features and workflows.
* **Performance Testing**: Evaluating response time, load handling, and scalability.
* **Security Testing**: Ensuring the application is protected from vulnerabilities.
* **Compatibility Testing**: Checking the application on different browsers and devices.
* **Database Testing**: Ensuring data integrity and query performance.
* **API Testing**: Validating the correctness and efficiency of API endpoints.

**Testing Environment**

The software will be tested in different environments to mimic real-world scenarios:

* **Development Environment**: Used for initial unit and integration testing.
* **Staging Environment**: A pre-production setup for system testing.
* **Production Environment**: Final testing before deployment to end users.

**Test Case Development**

Test cases will be documented for each module, including:

* **Test Case ID** – Unique identifier for each test case.
* **Description** – Summary of what the test case covers.
* **Preconditions** – Necessary setup or dependencies.
* **Steps to Execute** – Step-by-step process to perform the test.
* **Expected Outcome** – The anticipated result of the test.
* **Actual Outcome** – The observed result during execution.
* **Status** – Pass/Fail result of the test case.

**Testing Tools**

To streamline the testing process, various tools will be utilized:

* **Selenium** – For automated web application testing.
* **Postman** – For API testing.
* **JMeter** – For performance and load testing.
* **JUnit/PyTest** – For unit testing.
* **SonarQube** – For code quality and security analysi

##### **TEST CASES**

Table 7.1 Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test condition** | **Expected Output** | **Actual Output** | **Remark** |
| TC- 001 | User Login with valid credentials | User successfully logs in | User successfully logged in | Pass |
| TC-002 | User Login with incorrect credentials | Display "Invalid Username or Password" message | Displayed correct error message | Pass |
| TC-003 | User tries to access dashboard without logging in | Redirected to login page | Redirected to login page | Pass |
| TC-004 | User submits an empty signup form | Display "Required fields must be filled" message | Error message displayed as expected | Pass |
| TC-005 | File upload within allowed size limit | File uploaded successfully | File uploaded successfully | Pass |
| TC-006 | File upload exceeding size limit | Display "File size exceeds limit" error | Displayed correct error message | Pass |
| TC-007 | Database connection failure scenario | System displays a "Database error" message | Displayed correct error message | Pass |
| TC-008 | Performance testing: chatbot response time under 2 seconds | Chatbot responds within 2 seconds | Response time: 2.8 seconds | Partially correct |