**TEST PLAN**

1. **Introduction**:

* Purpose of the test plan
* Scope of testing
* Overview of the application under test

1. **Test Strategy**:

* Types of testing to be performed (e.g., functional, usability, performance, security)
* Levels of testing (e.g., unit, integration, system, acceptance)

1. **Test Criteria**:

* Entry and exit criteria for each test phase
* Pass/fail criteria for test cases

1. **Test Deliverable**:

* List of documents, reports, and scripts to be delivered

1. **Schedule**:

* Testing milestones and deadlines

1. **Risks and Mitigation**:

**1. Introduction**

**1.1 Purpose**

The purpose of this test plan is to outline the approach, scope, and methodologies for testing the web application. The goal is to ensure that the application is reliable, performs well, and meets the requirements specified in the project documentation.

### ****1.2 Scope****

This test plan covers functional, non-functional, and user acceptance testing of the web application. It includes details on test objectives, test criteria, and resources required for testing.

## ****1.3 Overview of the Application Under Test****

**Application Name:** [AbleSpace]

**Key Features:**

* **User Authentication:** Secure login and user management.
* **Dashboard:** Overview of key metrics and data.
* **Data Management:** Input and manage [specific data].
* **Reporting:** Generate and view reports.
* **Notifications:** Alerts for updates and important events.

**User Roles:**

* **Admin:** Manages settings and user accounts.
* **Registered User:** Accesses core functionalities.
* **Guest:** Limited access or view-only permissions.

**Security Features:**

* **Authentication:** Secure login and role-based access.
* **Encryption:** Data encryption in transit and at rest.

## ****2. Test Strategy****

### ****2.1 Types of Testing to be Performed****

1. **Functional Testing**

* **Objective:** Ensure that each function of the application operates in with the requirement specifications.
* **Scope:** Test cases will cover all user interactions, workflows, and business logic.

1. **Performance Testing**

* **Objective:** Assess the application's responsiveness, stability, and speed under various conditions.
* **Scope:** Includes load testing (handling expected load), stress testing (beyond normal load).

1. **Usability Testing**

* **Objective:** Evaluate the user interface and overall user experience to ensure the application is intuitive and user-friendly.
* **Scope:** Focuses on ease of navigation, accessibility, and overall user satisfaction.

**4.Compatibility Testing**

* **Objective:** Verify that the application works across different browsers, devices, and operating systems.
* **Scope:** Includes cross-browser testing, mobile responsiveness testing, and compatibility with various OS versions.

**6.Integration Testing**

* **Objective:** Test interactions between integrated components or systems to ensure they work together as expected.
* **Scope:** Includes testing of API integration, third-party services, and data exchange between modules.

### ****2.2 Levels of Testing****

1. **Unit Testing**

* **Objective:** Test individual components or units of code for correctness.
* **Scope:** Focuses on verifying the functionality of small, isolated parts of the application, such as functions or methods.

1. **Integration Testing**

* **Objective:** Test the interactions and interfaces between integrated components or systems.
* **Scope:** Ensures that different parts of the application work together and external integration function correctly.

1. **System Testing**

* **Objective:** Validate the complete and integrated application against the specified requirements.
* **Scope:** Includes end-to-end testing of the entire application to ensure that it behaves as expected in a fully integrated environment.

1. **Acceptance Testing (AT)**

* **Objective:** Ensure the application meets end-user requirements and is ready for production.
* **Scope:** Performed by end-users to verify that the application fulfills their needs and expectations in real-world scenarios.

## ****3. Test Criteria****

### ****3.1 Entry and Exit Criteria for Each Test Phase****

#### ****1.. Integration Testing****

* **Entry Criteria:**
* Ft testing of individual components is complete and defects have been resolved.
* Integrated modules are available for testing.
* **Exit Criteria:**
* All integration test cases have been executed.
* All critical and major integration issues have been resolved.
* Interfaces between modules and external systems are functioning as expected.
* Integration test results meet the defined success criteria.

#### ****2.. System Testing****

* **Entry Criteria:**
* Integration testing is complete and all integration issues have been addressed.
* System test cases are prepared and reviewed.
* The complete application is deployed in a test environment that mimics the production environment.
* **Exit Criteria:**
* All system test cases have been executed.
* All critical and major defects identified have been resolved.
* Application performance, security, and usability criteria are met.
* System test results are documented and approved.

#### ****3.. User Acceptance Testing (UAT)****

* **Entry Criteria:**
* System testing is complete and all major issues have been resolved.
* UAT test cases are prepared based on user requirements.
* UAT environment is set up with the latest release.
* **Exit Criteria:**
* All UAT test cases have been executed and passed.
* Feedback from end-users has been collected and addressed.
* All critical and major issues identified during UAT are resolved or documented with a plan for resolution.

### ****3.2 Pass/Fail Criteria for Test Cases****

* **Pass Criteria:**
* The actual result matches the expected result as defined in the test case.
* All steps in the test case are executed without errors.
* No critical or major defects are found during the execution of the test case.
* The application behaves as specified in the requirements and design documents.
* **Fail Criteria:**
* Platform failure
* Errors or issues occur during test execution that prevent the test case from completing.
* Critical or major defects are identified that prevent the functionality from working as intended.
* The application does not meet the requirements or design specifications.

## ****4. Test Deliverable****

### ****4.1 Documents****

1. **Test Plan Document**

* **Description:** Detailed plan outlining the overall testing strategy, scope, objectives, and resources.
* **Content:** Test strategy, types of testing, test phases, roles and responsibilities, and schedule.

1. **Test Case Specifications**

* **Description:** Document detailing individual test cases, including objectives, inputs, execution steps, and expected results.
* **Content:** Test case ID, description, preconditions, test steps, expected results, and postcondition.

1. **Test Data**

* **Description:** Data sets used for executing test cases.
* **Content:** Sample data for functional tests, data for edge cases, and any specific test scenarios.

1. **Environment Setup Document**

* **Description:** Configuration and setup instructions for the test environment.
* **Content:** Hardware and software requirements, environment configuration details, and setup procedures.

### ****4.2 Reports****

1. **Test Execution Report**

* **Description:** Summary of the test execution results.
* **Content:** List of executed test cases, pass/fail status, defects logged, and overall test coverage.

1. **Defect Report**

* **Description:** Detailed report on defects identified during testing.
* **Content:** Defect ID, description, severity, status, steps to reproduce, and resolution status.

1. **Test Summary Report**

* **Description:** Comprehensive summary of the testing process and outcomes.
* **Content:** Overview of testing activities, summary of results, defect trends, and overall assessment of application quality.

### ****4.3 Scripts****

1. **Automated Test Scripts**

* **Description:** Scripts used for automated testing.
* **Content:** Test automation code/scripts for functional, regression, and performance testing. Includes test scripts for tools like Selenium etc.

1. **Test Execution Scripts**

* **Description:** Scripts or procedures for executing test cases manually or automatically.
* **Content:** Step-by-step instructions for running test cases, including setup

## ****5. Schedule****

### ****5.1 Testing Milestones****

* **Test Planning Completion**
* **Description:** Finalization of the test plan document and overall testing strategy.
* **Deadline:** 23thOct,2024
* **Test Case Design Completion**
* **Description:** Finalization of all test case documents and test data preparation.
* **Deadline:** 23thOct,2024
* **Test Environment Setup**
* **Description:** Configuration of the test environment and validation that it is ready for testing.
* **Deadline:** 23thOct,2024

### ****5.2 Testing Timeline****

* **Test Planning:** 23thOct,2024- 23thOct,2024
* **Test Case Design:** 23th Oct,2024 - 23th Oct,2024
* **Test Reporting:** 23thOct,2024 - 23thOct,2024
* **Test Script**:23th Oct,2024 - 23th Oct,2024

## ****6. Risks and Mitigation****

### ****6.1 Potential Risks and Mitigation Strategies****

#### ****1. Unclear Requirements****

* **Risk:** Ambiguous or incomplete requirements may lead to inadequate test coverage or misaligned test cases.
* **Mitigation Strategies:**
* **Requirements Review:** Engage with stakeholders to clarify and document requirements thoroughly.
* **Frequent Communication:** Schedule regular meetings with stakeholders to address any ambiguities or changes in requirements.
* **Requirements Traceability:** Use traceability matrices to ensure all requirements are covered by test cases.

#### ****2. Defect Management Issues****

* **Risk:** High volume of defects or delayed resolution may impact the testing schedule and application quality.
* **Mitigation Strategies:**
* **Defect Tracking System:** Implement a robust defect tracking system to log, prioritize, and manage defects effectively.
* **Regular Defect Reviews:** Conduct regular reviews to assess defect status and ensure timely resolution.
* **Clear Communication:** Maintain clear communication channels between testers and developers to expedite defect resolution.

#### ****3. Schedule Delays****

* **Risk:** Testing activities may fall behind schedule, impacting project timelines and release dates.
* **Mitigation Strategies:**
* **Detailed Scheduling:** Develop a realistic and detailed testing schedule with built-in buffer times for unexpected delays.
* **Regular Progress Monitoring:** Track progress against the schedule and adjust resources or plans as needed.
* **Prioritization:** Prioritize critical test cases and focus on high-impact areas to ensure essential functionalities are tested first.

#### ****4 Environment Issues****

* **Risk:** Problems with the test environment (e.g., configuration issues, downtime) may hinder testing activities.
* **Mitigation Strategies:**
* **Environment Setup:** Ensure that the test environment mirrors the production environment as closely as possible.
* **Pretest Environment Validation:** Validate the environment setup before starting the testing phase.