Test Plan

Objective

The primary objective of this test plan is to ensure that the web application is production-ready. This includes verifying functionality, performance, security, and user experience while identifying and mitigating potential risks before deployment.

Scope

- Login functionality: Username, password, login button.
- Field validation: Validate username and password fields
- Dashboard page: Post-login navigation.
- 404 error handling: For invalid login attempts.
- Include accessibility, performance, and security testing.

Assumptions

- The application is bug-free.
- The environment is set up locally and mirrors production.

Types of Testing

Functional Testing

- Verify that all features (e.g., login functionality) work as intended

Non-Functional Testing

- **Performance Testing**: Ensure the application performance
- **Security Testing**: Protect against vulnerabilities like SQL injection
- **Accessibility Testing**: Confirm WCAG compliance for all UI components

Compatibility Testing:

- Verify compatibility across various browsers and devices
  - Browsers (e.g., Chrome, Firefox, Safari, Edge)
  - Devices (e.g., desktop, mobile, tablets)
  - Operating Systems (e.g., Windows, macOS, Linux, iOS, Android)

Regression Testing

- Ensure that new changes do not impact existing functionality

End-to-End Testing

- Simulate user workflows from start to finish, e.g., logging in and navigating to the dashboard

CI/CD Pipeline:

- Setup automation tests to be triggered as part of CI/CD pipeline

User Acceptance Testing (UAT)

- Verify that the application meets business requirements and expectations

Post-Deployment Validation

- Test in the production environment to confirm successful deployment

Testing Environments

Environment	Purpose	Setup
Local	Initial development and unit testing	Local machine setup
Dev	Testing of new features	Pre-configured environment
QA/Staging	Comprehensive functional testing	Production-like environment
Production	Final validation after deployment	Live environment with real data

Testing Requirements

- Access to test environments
- Test data resembling real-world scenarios
- Testing tools (Cypress, Lighthouse etc)
- CI/CD pipeline integration for automated testing

Test Data Examples

Data Type	Example Values
Valid Username	test, adminUser
Invalid Username	unknownUser, @invalid!name
Valid Password	test, Password123!
Invalid Password	12345, pass
Edge Cases	Empty username/password fields

Tools and Frameworks

- **Test Automation Framework**: Cypress (for E2E and integration testing)
- **API Testing:** Postman, Cypress
- **Performance Testing**: K6 for Backend APIs and Lighthouse for frontend.
- **Security Testing**: OWASP ZAP
- **Monitoring and Logging**: New Relic or Grafana with Prometheus
- **CI/CD Pipeline**: Jenkins, GitHub Actions, or CircleCI
- **Code Quality and Coverage**: SonarQube
- **Browser Testing**: BrowserStack or Sauce Labs for cross-browser compatibility
- **Accessibility testing:** WAVE API or Wick A11y Testing Environments
- **Bug Reporting & Tracking**: JIRA

Test Plan Phases

Phase 1: Requirement Gathering and Test Planning

- Review product requirements and gather requirements
- Define objectives, scope, and testing criteria.
- Identify risks and mitigation strategies.

Phase 2: Test Case Design

- Create test cases and scripts for functional, performance, and security tests.

Phase 3: Test Execution

- Run automated and manual tests in all environments.
- Identify and log defects, and work on resolutions.

```gherkin
Feature: Login functionality

  Scenario Outline: Test login with valid and invalid credentials
    Given the login page is displayed
    When I enter "<username>" and "<password>"
    And I click the login button
    Then I should see "<expectedMessage>" page

    Examples:
      | username     | password     | expectedMessage |
      | test         | test         | THE BAND        |
      | invalidUser  | validPass123 | 404 Not Found   |
```
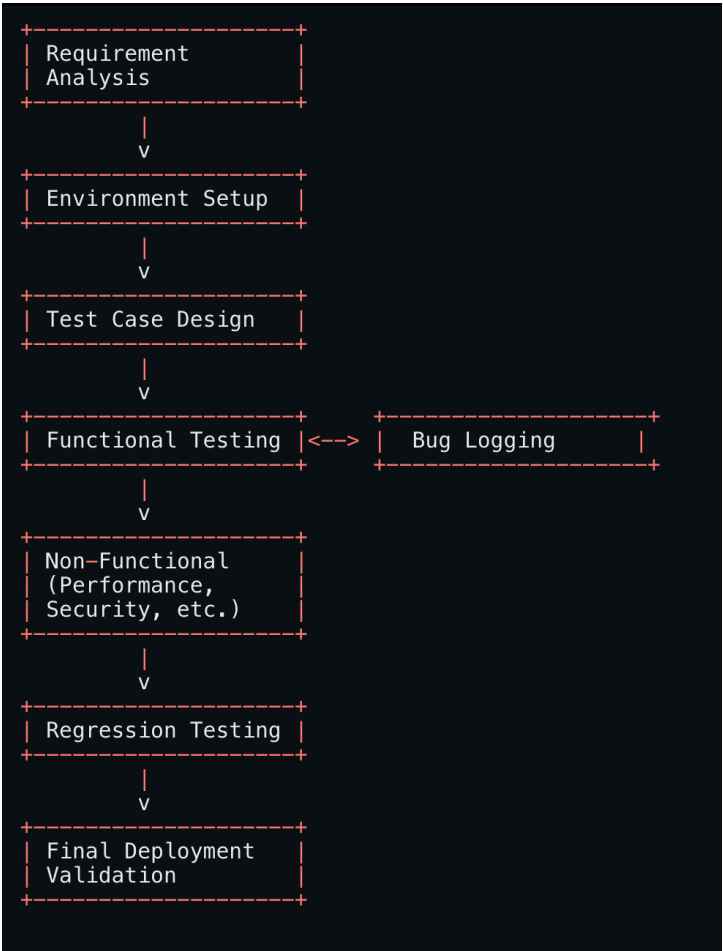
Phase 4: Reporting

- Share test results and coverage metrics.
- Highlight risks and unresolved issues.

Phase 5: Post-Deployment Validation

- Execute smoke tests in the production environment.
- Monitor application logs and metrics for anomalies.

```
+---------------------+
| Requirement         |
| Analysis            |
+---------------------+
          |
          v
+---------------------+
| Environment Setup   |
+---------------------+
          |
          v
+---------------------+
| Test Case Design    |
+---------------------+
          |
          v
+---------------------+        +---------------------+
| Functional Testing  |<-->|    Bug Logging      |
+---------------------+        +---------------------+
          |
          v
+---------------------+
| Non-Functional      |
| (Performance,       |
| Security, etc.)     |
+---------------------+
          |
          v
+---------------------+
| Regression Testing  |
+---------------------+
          |
          v
+---------------------+
| Final Deployment    |
| Validation          |
+---------------------+
```

Test Case Design

SR NO	TEST CASE	TEST STEPS	EXPECTED RESULTS	AUTOMATED
1	Login with valid credentials	**Given** the login page is displayed **When** the user enters valid username and password **When** I click on the login button **Then** I should see the dashboard page	The user is able to login successfully User is navigated to the dashboard	Yes
2	Login with invalid credentails	**Given** the login page is displayed **When** an invalid username and valid password are entered **When** I click on the login button **Then** I should see the 404 error page	The user is not able to login User is navigated to the 404 error page	Yes
3	Login with invalid password	**Given** the login page is displayed **When** the user enter username and invalid password **When** I click on the login button **Then** I should see the 404 error page	The user is not able to login User is navigated to the 404 error page	Yes
4	Login with invalid username	**Given** the login page is displayed When the user enters invalid username and valid password **When** I click on the login button **Then** I should see the 404 error page	The user is not able to login User is navigated to the 404 error page	Yes
5	Login with empty username	**Given** the login page is displayed **When** the user enters empty username and password **When** I click on the login button **Then** I see pop up `Please fill out this field`	The user is not able to login A pop appears `Please fill out this field`	No
6	Login with empty password	**Given** the login page is displayed **When** the user enters username and empty password **When** I click on the login button **Then** I see pop up `Please fill out this field`	The user is not able to login A pop appears `Please fill out this field`	No
		Given the login page is displayed		

7	Login with empty username and password	**When** the user pass empty username and password **When** I click on the login button **Then** I see pop up `Please fill out this field`	The user is not able to login A pop appears `Please fill out this field`	Yes
8	Case Sensitivity for Username and Password	**Given** the login page is displayed **When** the user enters random case username and password **When** I click on the login button **Then** I should see the 404 error page	The user is not able to login User is navigated to the 404 error page	No
9	Password Field Masking	**Given** the login page is displayed **When** the user enters username and password **Then** the password field should be masked	The password field is masked	No

Risk Assessment

Risk	Likelihood	Impact	Mitigation
Bugs in critical workflows	High	High	Prioritize regression and E2E tests.
Performance degradation under load	Medium	High	Run performance tests with realistic load.
Security vulnerabilities	Medium	High	Perform comprehensive security testing.
Deployment failure	Low	High	Validate deployments in staging first.

Entry and Exit Criteria

Entry Criteria

- Development is complete

- Unit and integration tests pass in the local environment
- Test environments are stable and accessible

Exit Criteria

- All critical and high-severity defects are resolved
- Test coverage meets predefined benchmarks
- Stakeholder sign-off is obtained for production deployment

Report and Documentation

- **Test Execution Report**: Includes pass/fail status, defect logs, and coverage metrics.
- **Risk Assessment Document**: Details identified risks and mitigation strategies.
- **Deployment Checklist**: Ensures that all prerequisites are met before production release.

Post-Deployment Validation

- Run smoke tests in production to verify critical functionality (e.g., login, navigation)
- Monitor application logs for errors
- Validate rollback mechanisms in case of failure

Conclusion

This test plan ensures the web application is robust, secure, and meets user expectations before deployment. It incorporates functional, non-functional, and post-deployment validation for comprehensive testing coverage.