**Test Plan for Web Portal**

**1. Introduction** This test plan outlines the strategy and approach for testing the web portal, focusing on onboarding and sign-in modules. The key testing areas include functional, performance, API, database, automation, and security testing.

**2. Scope** The scope of this testing plan covers the following modules:

* **Onboarding (Sign Up)**
  + Business Email
  + First Name
  + Last Name
  + Phone Number with Country Code
  + Password Field
* **Sign In Module**
  + Email ID and Password
  + Sign in via Social Media

**3. Testing Objectives**

* Ensure the onboarding and sign-in functionalities work as expected.
* Validate data integrity and security of user credentials.
* Assess performance under different loads.
* Ensure APIs function correctly and return expected responses.
* Validate database transactions and consistency.
* Automate repetitive test cases to enhance efficiency.

**4. Test Strategy** The testing strategy for this web portal involves a combination of manual and automated testing techniques. The approach includes:

* **Requirement Analysis:** Understanding business and technical requirements to design appropriate test cases.
* **Test Planning:** Identifying test scenarios, defining scope, and determining testing tools.
* **Test Case Design:** Writing functional and non-functional test cases covering all edge cases.
* **Test Execution:** Executing manual and automated test cases in different environments.
* **Defect Management:** Logging defects, tracking them to closure, and performing re-testing.
* **Test Reporting:** Providing comprehensive reports on test execution, defects, and overall application quality.

**5. Testing Approach**

### **5.1 Functional Testing**

* Verify that all form fields accept valid inputs.
* Validate mandatory fields and error handling for invalid/missing data.
* Test different combinations of email, phone number, and password inputs.
* Validate the sign-in process using email/password and social media.
* Ensure password complexity requirements are enforced.
* Check session management and logout functionality.

### **5.2 Performance Testing**

* Load Testing: Assess the response time of sign-up and sign-in with multiple users.
* Stress Testing: Determine the system’s behavior under extreme load conditions.
* Scalability Testing: Validate system performance when scaling up user registrations and sign-ins.

### **5.3 API Testing**

* Verify API request and response for sign-up and sign-in.
* Check response time, status codes, and data validation.
* Test authentication token generation and expiration.
* Validate API security, including unauthorized access handling.

### **5.4 Database Testing**

* Verify data storage and retrieval for user details.
* Ensure data consistency across multiple user sessions.
* Check encryption and security of stored passwords.
* Validate constraints such as unique email IDs and phone numbers.

### **5.5 Automation Testing**

* Automate sign-up and sign-in workflows.
* Implement UI automation using Selenium or similar tools.
* Use API automation to validate backend responses.
* Integrate automated test cases with CI/CD pipelines.

### **5.6 Security Testing**

* Test against SQL injection, XSS, and other web vulnerabilities.
* Validate secure password storage and encryption techniques.
* Check for session management vulnerabilities.
* Perform penetration testing to identify security loopholes.

**6. Test Environment**

* **Operating Systems:** Windows, macOS, Linux
* **Browsers:** Chrome, Firefox, Edge, Safari
* **Devices:** Desktop, Mobile, Tablet
* **Database:** MySQL / PostgreSQL
* **Automation Tools:** Selenium, Postman, JMeter, Cypress
* **Security Tools:** OWASP ZAP, Burp Suite
* **Test Management Tools:** JIRA, TestRail

**7. Entry and Exit Criteria**

* Entry Criteria: Test environment setup, test cases documented, application ready for testing.
* Exit Criteria: All critical test cases executed, major defects resolved, performance benchmarks met.

**8. Deliverables**

* Test cases
* Test execution reports
* Defect reports
* Performance and security assessment reports
* Automation scripts

**9. Risk Analysis and Mitigation**

* **Unstable application:** Continuous communication with developers to fix issues early.
* **Security vulnerabilities:** Conduct periodic penetration testing.
* **Scalability concerns:** Regular load testing to identify bottlenecks.
* **Incomplete requirements:** Maintain close collaboration with stakeholders for requirement clarification.
* **Third-party dependency failures:** Implement fallback mechanisms and conduct integration testing.

**10. Defect Reporting Procedure**

* **Defect Identification:** Testers identify defects during test execution and log them in the defect tracking system (e.g., JIRA, TestRail).
* **Defect Logging:** Each defect should include:
  + Defect ID
  + Summary
  + Description with steps to reproduce
  + Expected vs. actual results
  + Screenshots/logs (if applicable)
  + Severity and priority
  + Assigned developer
* **Defect Triage:** Regular meetings with stakeholders to prioritize defects based on impact and urgency.
* **Defect Fixing:** Developers address defects based on priority and assign them for retesting.
* **Retesting and Regression Testing:** Testers validate fixed defects and conduct regression testing to ensure no new issues have been introduced.
* **Closure:** Defects are marked as closed once validated by the test team.

**11. Approvals**

| **Role** | **Name** | **Designation** | **Date** | **Signature** |
| --- | --- | --- | --- | --- |
| Test Manager | [Name] | Test Manager | [Date] | [Signature] |
| QA Lead | [Name] | QA Lead | [Date] | [Signature] |
| Project Manager | [Name] | Project Manager | [Date] | [Signature] |
| Business Analyst | [Name] | Business Analyst | [Date] | [Signature] |

**12. Conclusion** This test plan ensures a structured approach to verifying the web portal’s functionality, performance, security, and reliability, ensuring a seamless user experience.