**Test Plan for Facebook Login Page**

**1. Introduction**

**Objective:**  
To ensure the Facebook login page functions correctly and securely across various browsers and devices.

**Scope:**

* Testing login functionality.
* Verifying input field validations.
* Ensuring proper error message handling.
* Assessing UI/UX consistency.
* Security and performance testing.

**2. Features to be Tested**

1. **Login Form:**
   * Email/Phone number input field.
   * Password input field.
   * "Log In" button.
   * "Forgotten password?" link.
   * "Create New Account" button/link.
2. **Input Validations:**
   * Correct format for email/phone number.
   * Password complexity requirements.
   * Error messages for incorrect inputs.
3. **Error Handling:**
   * Incorrect email/phone number or password.
   * Blank input fields.
4. **UI/UX:**
   * Consistency across different browsers and devices.
   * Accessibility features (screen readers, keyboard navigation).
5. **Security:**
   * Prevention of SQL injection, XSS, and other vulnerabilities.
   * Secure transmission of credentials.
6. **Performance:**
   * Page load time.
   * Response time after form submission.

**3. Testing Strategy**

* **Functional Testing:** To verify each feature works as expected.
* **Usability Testing:** To ensure the interface is user-friendly.
* **Security Testing:** To identify and mitigate security vulnerabilities.
* **Performance Testing:** To check the page's responsiveness and load times.
* **Cross-browser Testing:** To verify functionality across different web browsers.
* **Cross-device Testing:** To ensure compatibility with various devices (desktops, tablets, smartphones).

**4. Test Environment**

**Browsers:**

* Chrome
* Firefox
* Safari
* Edge
* Internet Explorer 11

**Devices:**

* Desktop (Windows, macOS)
* Tablet (iOS, Android)
* Smartphone (iOS, Android)

**5. Resources**

**Testers:**

* Manual Testers
* Automation Engineers

**Tools:**

* Selenium for automation testing.
* BrowserStack for cross-browser testing.
* OWASP ZAP for security testing.
* JMeter for performance testing.

1. **Schedule**

|  |  |  |
| --- | --- | --- |
| **Task** | **Start Date** | **End Date** |
| Test Planning | 2024-05-24 | 2024-05-26 |
| Test Case Design | 2024-05-27 | 2024-05-29 |
| Test Environment Setup | 2024-05-30 | 2024-05-31 |
| Test Execution (Functional) | 2024-06-01 | 2024-06-05 |
| Test Execution (Security) | 2024-06-06 | 2024-06-07 |
| Test Execution (Performance) | 2024-06-08 | 2024-06-09 |
| Test Execution (Cross-browser) | 2024-06-10 | 2024-06-12 |
| Test Execution (Cross-device) | 2024-06-13 | 2024-06-14 |
| Bug Reporting and Re-testing | 2024-06-15 | 2024-06-17 |
| Test Closure | 2024-06-18 | 2024-06-19 |

**7. Entry and Exit Criteria**

**Entry Criteria:**

* Test environment set up.
* Test cases reviewed and approved.
* Application build is ready for testing.

**Exit Criteria:**

* All planned test cases executed.
* Critical and major bugs fixed and verified.
* Test summary report prepared.

**8. Risk and Mitigation**

**Risks:**

* Delay in test environment setup.
* Incomplete test data.
* Unexpected application downtime.

**Mitigation:**

* Prepare test environment in advance.
* Use realistic and comprehensive test data.
* Coordinate with development team for stable build.

**9. Test Deliverables**

* Test Plan
* Test Cases
* Test Scripts (for automation)
* Test Summary Report
* Bug Reports