Test Plan

# Project Title:

Facebook Registration Page Testing

# 1. Introduction

## 1.1 Purpose

This test plan outlines the overall testing approach, execution strategy, and management activities for validating the functionality, usability, security, and performance of the Facebook registration page: https://www.facebook.com/r.php?entry\_point=login.

## 1.2 Project Overview

The Facebook registration page allows new users to create an account by entering personal information such as name, email/phone number, password, date of birth, and gender. This test plan focuses on verifying that the registration process works as expected under various scenarios.

# 2. Scope

## 2.1 In-Scope

- Functional testing of the registration process:  
 - Form field validation (name, email/phone, password, date of birth, gender)  
 - CAPTCHA/robot verification (if applicable)  
 - Error messages and handling of invalid inputs  
 - Link redirections (e.g., Terms & Privacy)  
 - Form submission and success/failure scenarios  
- Usability and accessibility testing  
- Compatibility testing across major browsers (Chrome, Firefox, Safari, Edge)  
- Security testing (basic: field validation, input sanitization)

## 2.2 Out-of-Scope

- Backend user account management  
- Database or backend service testing  
- Social integrations (Facebook login on third-party sites)  
- In-depth performance testing beyond form submission

# 3. Testing Strategy

## 3.1 Test Objectives

- Verify that users can successfully register on the Facebook platform.  
- Ensure that the registration form enforces correct validations.  
- Confirm that error messages are clear and actionable.  
- Validate usability and accessibility on different browsers and devices.

## 3.2 Test Assumptions

- The development team has completed unit testing.  
- Test environment mirrors the production environment in terms of configuration and data.  
- Test data is available and does not conflict with existing users.

## 3.3 Data Approach

Test data sets will include valid and invalid user registration details:  
- Valid names, email/phone numbers, passwords  
- Invalid data (empty fields, short passwords, invalid email formats)  
- Edge cases (special characters, SQL injection attempts)

## 3.4 Level of Testing

|  |  |  |
| --- | --- | --- |
| Test Type | Description | Responsible Parties |
| Unit Testing | Performed by developers to verify individual components. | Development Team |
| Functional Testing | Validate the complete registration flow end-to-end. | QA Team |
| User Acceptance Test | Ensure the page meets user expectations and is easy to use. | Product Owner, Stakeholders |
| Regression Testing | Validate that new changes do not break existing functionality. | QA Team |

## 3.5 Unit Testing

- Fields: Name, Email/Phone, Password, DOB, Gender.  
- Form submission and error message handling.

Participants:

|  |  |  |
| --- | --- | --- |
| Name | Department | Role |
| Dev Team | Engineering | Test Manager |

## 3.6 Functional Testing

- All form field validations.  
- Error messages.  
- Success scenarios.  
- Redirects and link validations.

|  |  |  |
| --- | --- | --- |
| Name | Department | Role |
| QA Team | Quality Assurance | Test Analyst |

## 3.7 User Acceptance Testing

- Confirm that registration is intuitive and straightforward.  
- Ensure compliance with accessibility guidelines.

|  |  |  |
| --- | --- | --- |
| Name | Department | Role |
| Product Owner | Product | Test Manager |
| Stakeholders | Business | Test Analyst |

## 3.8 Regression Testing

- Retest all previously tested features after any code changes or fixes.

|  |  |  |
| --- | --- | --- |
| Name | Department | Role |
| QA Team | Quality Assurance | Test Analyst |

# 4. Execution Strategy

## 4.1 Entry Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| Entry Criteria | Test Team | Technical Team | Notes |
| Test environment is available | ✓ |  |  |
| Test data is available | ✓ |  |  |
| Code has been merged successfully |  | ✓ |  |
| Development has completed unit testing |  | ✓ |  |
| Test scripts are completed, reviewed, and approved by the Project Team | ✓ |  |  |

## 4.2 Exit Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| Exit Criteria | Test Team | Technical Team | Notes |
| 100% Test Scripts executed | ✓ |  |  |
| 90% pass rate of Test Scripts | ✓ |  |  |
| No open Critical and High severity defects | ✓ |  |  |
| All remaining defects documented as Change Requests | ✓ |  |  |
| All expected and actual results documented | ✓ |  |  |
| All test metrics collected | ✓ |  |  |
| All defects logged in Defect Tracker | ✓ |  |  |
| Test environment cleanup completed | ✓ |  |  |

## 4.3 Validation and Defect Management

Test cases validated using expected vs actual results.  
Defects logged and tracked using the Defect Tracker.  
Severity Levels:  
- 1 (Critical): Registration blocked; no workaround.  
- 2 (High): Major functionality not working; workaround may exist.  
- 3 (Medium): Functionality degraded; workaround available.  
- 4 (Low): Cosmetic or minor usability issue.

# 5. Environment Requirements

## 5.1 Test Environments

- QA Environment replicating production setup  
- Browsers: Chrome, Firefox, Edge, Safari  
- Secure network environment with HTTPS enabled

# 6. Significantly Impacted Division/College/Department

|  |  |  |
| --- | --- | --- |
| Business Area | Business Manager | Tester(s) |
| Facebook Web Development | Dev Manager | QA Team |

# 7. Dependencies

- Availability of test environment and data  
- Availability of the registration page deployment  
- Developer support for fixing defects