Module 1: Introduction to Software Testing

Demo Document

edureka!

edureka!

Title:- Social Networking site- Account creation page

Demo 1- Statement

A MNC company ABC.NET is about to launch a social networking site and it has a requirement of account creation page. User will be allowed to access the site after creating account.

An IT company ABC.CORP gets the requirement from the Client and agrees to provide a Signup page.

In this demo, we go through different phases of SDLC and explain testing types with examples.

Requirement Collection: -

1) User Name input for account. First Name and Surname
2) Account credentials: - mobile/email address and password
3) Birthday: - DD/MM/YYYY (particular age group is allowed to create account based on this input)
4) Gender (M or F)
5) Signup Button (Submit action)
6) Hover wherever necessary for extra information

Design: - According to Client requirement statements, the pictorial representation of Signup page is done.

Development: - The programming language, libraries, databases, caching, framework will be decided by the development team.

Testing Phase: - Once the Signup page is ready. It is available on testing environment.

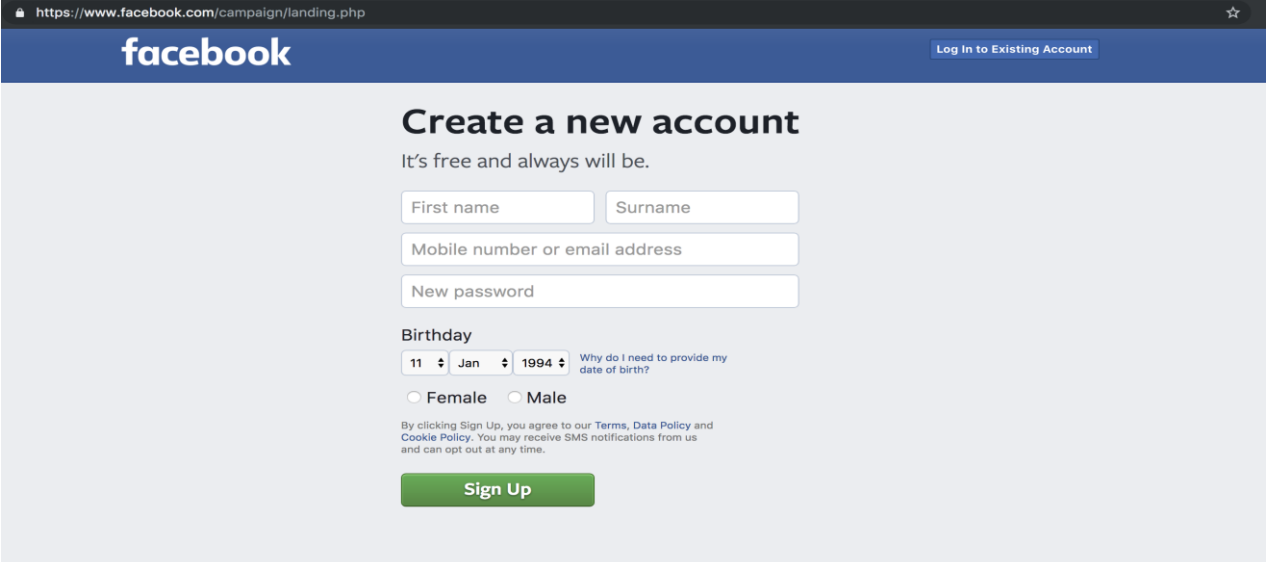The final signup page looks like this. --->Image

Test Web Page link:
https://www.facebook.com/campaign/landing.php

Solution

Topic 1: Smoke Testing

a) Click on the test link and Verify that the application launches successfully.
Link provided should lead to the application landing page.

b) Verify Graphical User Interface(GUI) input fields. All web fields should be responsive and user entered input should get filled.

Topic 2: Functional Testing

a) Perform Positive testing by providing valid inputs to test happy flow.

b) Perform Negative testing by providing invalid inputs, for which application should return warnings and errors. Invalid inputs can be null values, invalid data type, extra characters, etc..
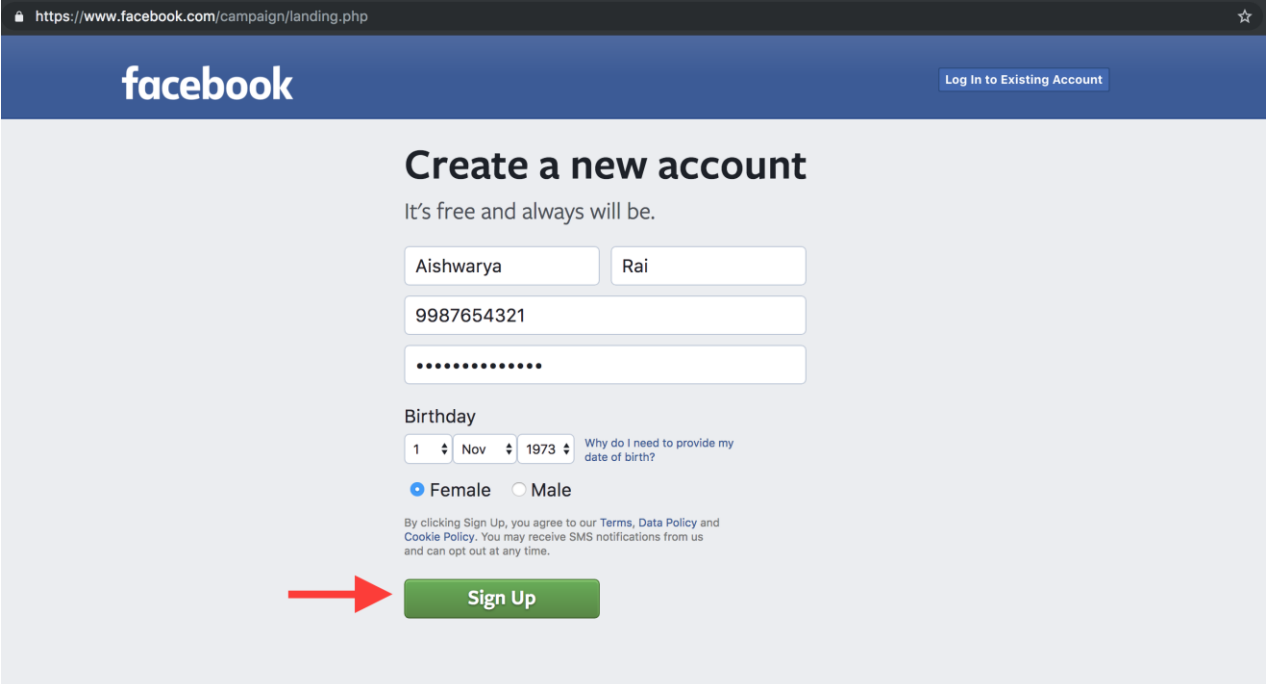
Topic 3: Sanity testing

After receiving the build from developer, which has minor changes in code or functionality, sanity testing is done to ensure that bugs fixes didn't cause new issues in existing application.

The goal is roughly test proposed functionality. If sanity fails, the build is rejected to save the time and costs involved in further testing.

As this is the first build provided by developer, sanity testing should be done for checking basic functionality.

Testing account creation after providing valid inputs and clicking on Submit/SignUp.

Topic 4: Regression Testing
To make sure none of the new changes caused new bugs. In regression testing we should make sure no old bugs appear from the addition of new software modules over time.

Example:- If Login flow is built in second phase after SignUp page.
Then in regression of the build, Signup functionality should be checked in regression testing.

Topic 5: Unit Testing

Testing performed on each single module or block of code during development by developer.

In front end user is calling link https://www.facebook.com/campaign/landing.php ,internally it is calling to java class Signup, method createAccount

public static boolean createAccount(String firstName,String surName, String mobEmail, String password, Date birthDate, String gender)

Developer will do unit testing using JUnit/TestNg which will just validate this method creatAccount.

Assert.assetTrue(Signup.createAccount("Alex","Newton","alexnewton@gmail.com", "alex123","1981-01-01","Male"));
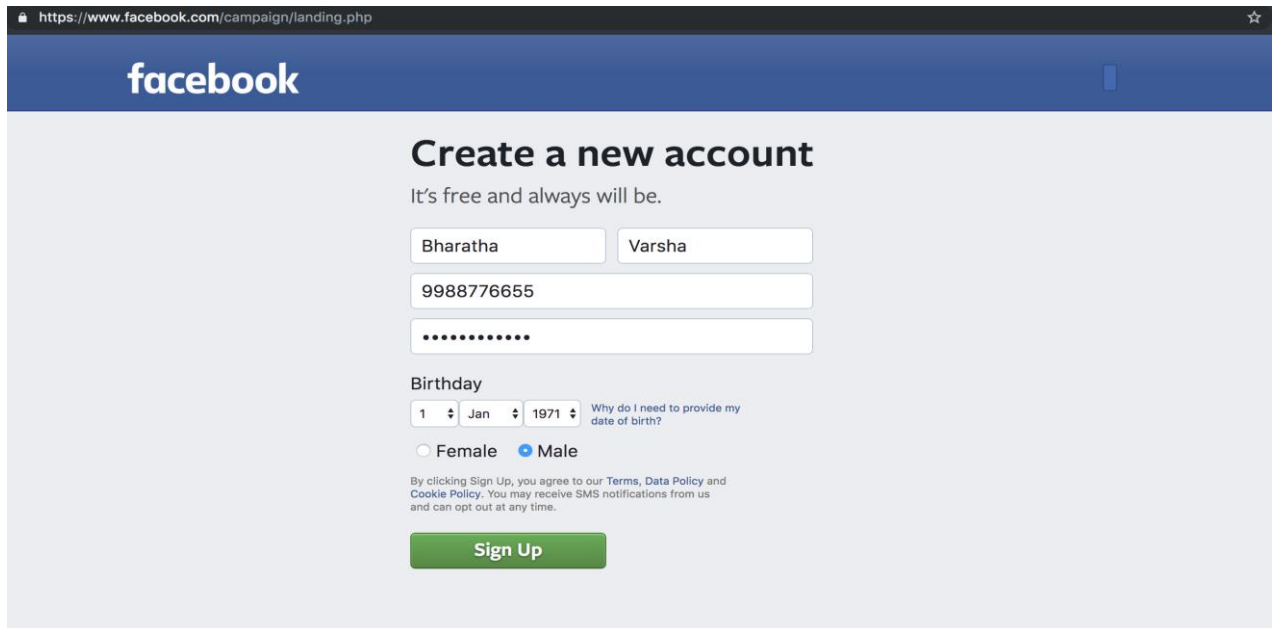//If the method returns **true** for valid inputs then testcase is passed.

Topic 6: Integration Testing/ Interface Testing(API)

Integration Testing becomes necessary to verify that the software modules work in unity. It focuses mainly on the interfaces & flow of data/information between the modules.
SignUp module and user mail/mobile are two modules here.

a)Sign up of the user through website
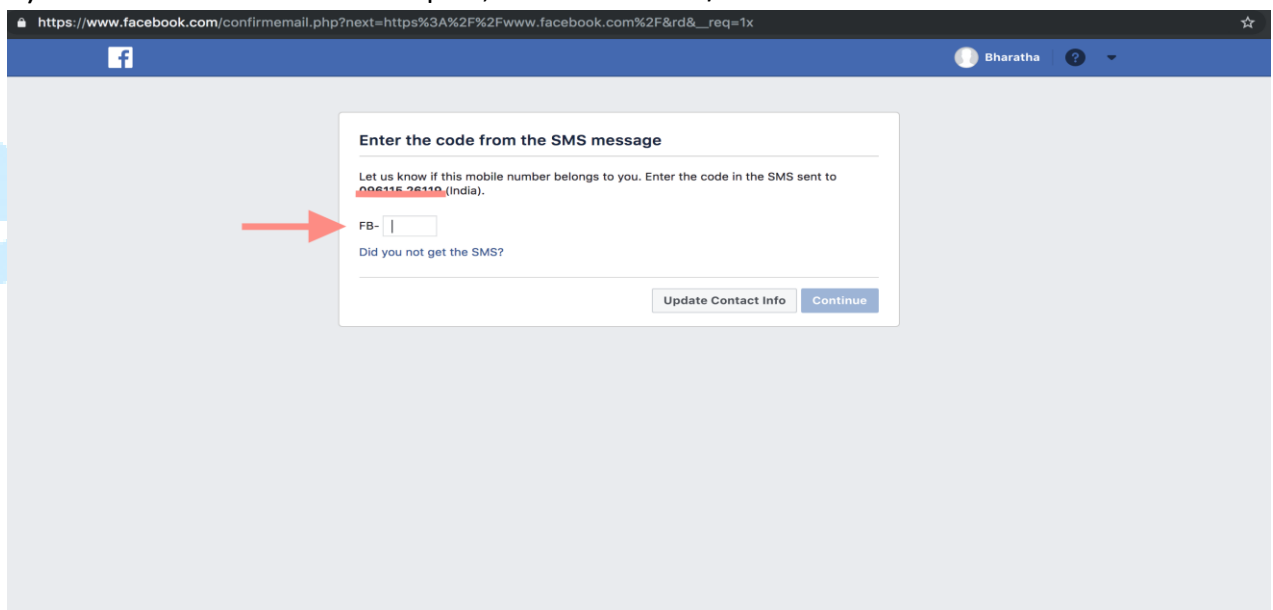
b) After authentication of the inputs, verification mail/code is sent to user.

c) Once after user authentication is done, successful account creation message is sent to user.

Topic 7: System Testing/End to End testing

Testing the fully integrated applications including external peripherals in order to check how components interact with one another and with the system as a whole.

Example:- After building of SignUp and integrating with social networking site, system testing should be carried out to test end to end flows.

- Site should be accessible after logging through created accounts.

- Notifications from site should be delivered to account.
- On Web and mobile apps sign-up and login should work as expected.

a)Testing of SignUp though web environment

b)Testing of SignUp in app environment

Topic 8: User acceptance testing

Beta testing of the product done by the actual end users. The beta version of application will be given to users to use.

Example:- For some user Ids the app version is made available in the company internally. Employees feedbacks will be collected to know first set of end user impressions.

Topic 9: Non functional- Load/performance Testing

Load/Performance testing should be carried out on software, to evaluate performance under real life load constraints.

Example:- SignUp creation link will be hit by n number of users at a time for creation of accounts, then application should work as expected under more load also.

Topic 10: Non functional- Security testing

The application is safeguarded against deliberate and sudden attacks from internal and external sources.(malware)

Example:-

- Validation of APIs by passing some security breaking inputs , due to which internal code gives some unexpected outputs.
- Identifying network and system weakness if present.
- Checking that any redirections to internal APIs from exposed APIs.
- Verifying that the response from the exposed APIs should not have confidential data.
- Storing data and password of the accounts in encrypted format.

UI vs API testing

UI testing is performed by providing inputs directly in the site page and validating the behaviour.

API Testing is performed using tools where API is called by passing inputs and response is validated. **Postman** is one of the tool which is used for API testing.

url:- http://{{hostname}}/{{api_end_point}}

headers:- Content-type: application/json (type of content being passed in request)

http-method: POST