

**E-Voting Portal**

Software Test Plan

**Group Members -**

**1.G Srisha Anagh (IIT2016030)**

**2.Sai Charan Teja (IIT2016039)**

**3.Anmol Singh Sethi (IIT2016040)**

**4.Manavdeep Singh (IIT2016042)**

**5.Nairit Banerjee (IIT2016505)**

**Table of Contents**

1. INTRODUCTION
2. OBJECTIVE
3. SCOPE
4. Features to be tested.
5. Featured not to be tested.
6. TESTING PROCESS OVERVIEW
7. TEST STRATEGY
8. GUI Testing
9. Unit Testing
10. Integration Testing
11. System Testing
12. Validation Testing
13. User Acceptance Testing
14. TEST ENVIRONMENT
15. TEST SCHEDULE
16. CONTROL PROCEDURE
17. ROLES AND RESPONSIBILITIES
18. DELIVERABLE
19. ENTRY CRITERIA
20. SUSPENSION CRITERIA
21. RESUMPTION CRITERIA
22. EXIT CRITERIA

**1. INTRODUCTION**

This document will address the different standards that will apply to the unit, integration and system testing of the specified application.

**2. OBJECTIVE**

Objective of Test plan is to define the various Testing strategies and testing tools used for complete Testing life cycle of this project.

**3. SCOPE**

The document mainly targets the Front end testing and validating data in report output as per Requirements Specifications provided by Users.

**3.1 Features to be tested:**

1. The system should be able to take input from user which includes inputting the details to register to create the profile in the online voting portal.
2. Then further, the portal should be able to insert the values into the database. We must check the validity of data entered.
3. The System should be able to add an entry as a candidate once the user nominates himself and the election officer validates them.
4. The routes should be clear for the functionalities to make sure the user is not “by mistake” directed to electionofficer page or admin.

**3.2 Features not to be tested:**

1. Since this a portal which can be used from anywhere and anytime in the world, so the server should be well managed for such kind of requirements.
2. The users using the portal should read the tips displayed while first use in order to get familiar with it and get the best performance out of it.

**4. TESTING PROCESS OVERVIEW**

**Test Process :**

* Understanding of requirements from the requirements specifications.
* Preparing test cases based on the requirement specifications. This will cover all scenarios for requirements.
* Reviewing test cases
* Creating test data based on Test scenarios and Test cases.
* Executing Test Cases
* Update test result (Actual Result, Pass/Fail) in test case document.

**5. TESTING STRATEGY**

Software testing is a process of running with intent of finding errors in software.Software testing assures the quality of software and represents final review of other phases of software like specification, design, code generation etc.

**5.1 GUI Testing**

**(i) Checking for display of graphic elements of app on devices of different screen sizes**

**Test Id** - GUI01

**Purpose** - To check that the relative display layout of the application.

**Steps** -

1. Verify that all the display elements are placed correctly as expected on the screen.

**Expected Outcomes** - Relative GUI layout is as expected on devices of different screen sizes.

**(ii). Colour and theme of the Device**

**Test ID**- GUI02

**Steps**:-

1. Verify that the colour and theme of the application is consistent with the different colours and theme schemes of the phone.
2. Verify that colour and theme is consistent for the whole app.

**(iii). Font Style, Type and Size**

**Test ID**- GUI03

**Purpose:**- Make sure the style, size and font type of text is same throughout the app.

**Steps:-**

1. Verify that visual colour themes are consistent..
2. Verify usage of standard and professional font.

**(iv) User Friendliness**

**Test ID**- GUI04

**Purpose**:- User is able to understand the purpose of different elements of the app easily.

**Steps**:-

1. Verify that error messages are conveyed in an understandable and informative manner.
2. Verify that The icons of the buttons and text fields are self explanatory.

**(v) GUI compatibility**

**Test ID**- GUI05

**Purpose**:- To check if GUI of app is compatible and adaptable to different APIs.

**Steps**:-

1. Verify that the graphical elements and themes are working as expected in different levels of APIs.

**5.2 Unit Testing**

Unit testing is a technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules.Each Module is checked for expected functionality by testing them for basic working.

Each of the modules was tested as separate units. In each module all the functionalities were tested in isolation.

**(i) Insertion in Database**

**Test ID**- UNIT01

**Assumptions** : A particular module is already selected.

**Purpose:**- To check that voter details are being inserted into respective tables in the database.

**Steps:-**

1. Verify that the data added by the user is inserted correctly in corresponding table of the database file in MobileVoter.
2. Verify that data format is as desired by developer.

**(ii) Viewing of Results**

**Test ID**- UNIT02

**Purpose:**- To check that all the results are displayed in a sorted manner as expected by developer.

**Steps:-**

1. Verify that all the candidates are displayed as required on the results page.
2. Verify the details of each candidate is correctly shown.

**(iii) Delete Candidate**

**Test ID**- UNIT03

**Purpose:**- To check that user can delete individual nomination.

**Steps:-**

1. Verify that the nomination is updated correctly in database.
2. Verify that the deleted candidate details are removed from database.

**5.3 Integration Testing**

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs afterunit testing .

We approached integration testing in a bottom up way. One by one modules were integrated and modifications were made to make them compatible and run the application successfully.

**(i) Compatibility after Integration**

**Test ID**- INT01

**Assumptions** : Code is integrated .

**Purpose:**- To check that application runs properly as expected after integration.

**Steps:-**

1. Verify that different pages are triggered by specific actions and their functionality is maintained.
2. Verify that functionalities of pages are as per expected.

**5.4 System Testing**

System testing means testing the system as a whole. All the modules/components are integrated in order to verify if the system works as expected or not.

System testing is done after integration testing. This plays an important role in delivering a high-quality product.

**(i) Launching Web App -**

**Test Id** - SYS01

**Assumptions** - The code for frontend and backend integrates well.

**Purpose** - To verify that the web app launches properly without any problems.

**Steps** -

1. Verify that the web app is launched successfully and all the GUI components of the modules are visible to user as expected.

**Expected Outcome** - The web app opens properly and user is able to see all the GUI components with notifications appearing on screen properly.

**Actual Outcome** - The web app launches successfully and all components appears even though app is not currently in use.

**Status** - Pass.

**5.5 Validation Testing**

**(i) Email ID validity**

**Test ID**- VAL01

**Purpose:**- To check if the data is valid that is entered by the user.

**Steps:-**

1. Enter the data wrong for example wrong email ID etc. Hence, the account would not be verified.

**Expected Outcome** - Account not verified. Hence not able to login into account.

**Actual Outcome** - Validity tested as desired.

**Status** - Pass

**(ii) Phone Number validity**

**Test ID**- VAL02

**Purpose:**- To check that the date input taken by user is correct according to standards.

**Steps:-**

1. Write wrongly interpreted data. For example in phone number, the number of digits if less than 10.

**Expected Outcome** -Error message in sending OTP.

**Actual Outcome** - Validity tested as desired.

**Status** - Pass

**(iii) Necessary Fields validity**

**Test ID**- VAL03

**Purpose:**- To check that all the fields necessary if not filled, would result in non submission of the user details to database.

**Steps:-**

1. Attempt to submit the form without all the necessary fields filled out.

**Expected Outcome** - Error message thrown.

**Actual Outcome** - Validity tested as desired.

**Status** - Pass

**5.6 User Acceptance Testing**

**(i) Web App able to take input from user**

**Test ID**- UAC01

**Purpose:**- To check that the web app is able to take input from user including the signature file, photo link etc.

**Steps:-**

1. Verify that the website is taking all the files with varied files size into the database.

**Expected Outcome** - Input taken correctly and according to different module requirements.

**Actual Outcome** - Input taken properly.

**Status** - Pass

**(ii) Proper functioning of election time testing.**

**Test ID**- UAC02

**Purpose:**- To check if the website application is able to properly execute the elections with the start and stop function allowed by the election officer.

**Steps:-**

1. Verify for all the cases where the election is started and the election officer is deleted.

**Expected Outcome** - Election remains functioning

**Actual Outcome** - Election remains functioning as the attribute of the constituency is set. Hence for that constituency the elections will be open till another election officer stops it.

**Status** - Pass

**(iii) Load from database in background**

**Test ID**- UAC03

**Purpose:**- System should be able to load all the images and signatures properly in order to display for the voting

**Steps:-**

1. Verify that all the candidates images/ files necessary are loaded on the dashboard for the voter to vote for.

**Expected Outcome** - In case of the image not loaded yet, an empty container is present with alternate text.

**Actual Outcome** -Display as expected.

**Status** - Pass

**6. TEST ENVIRONMENT**

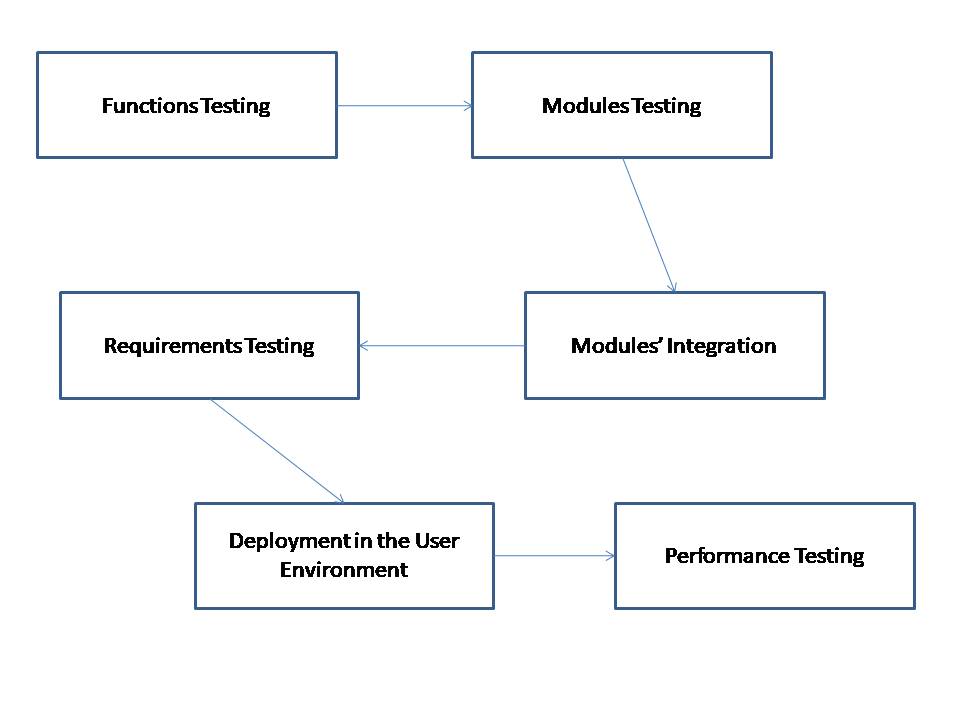
The web application is to be used by the flask backend. The minimum requirements must be:

1. A proper functioning web browser for the application to run.

We check the functionality with different systems with different operating systems and different web browsers. The effect is the same.

Outcome - The developed webapp passes all the tests in the given test environment.

Expected Environments - Different android versions.

**7. CONTROL PROCEDURE **

**8. ROLES AND RESPONSIBILITIES**

1. To create a database to store all the information of the voters to be added by the user.
2. To create a general layout of the website that would be deployed in the system.
3. To allow admin to delete an election officer
4. To allow a candidate to withdraw nomination.
5. To allow user to self nominate him/her self as a candidate for his/her constituency.
6. To create different modules for the layout of different type of users so as to take input accordingly.
7. To add DatePicker for date input of DateOfBirth, to increase user’s ease of use.
8. To check for the validity of data entered by the user.
9. To check if the email and phone number are valid as OTP and confirmation mail must be sent.

**9. SCHEDULE**

* **Feasibility report** - August 16th 2018
* **SRS report -** September 25th 2018
* **Design document -** October 12th 2018
* **Database Design -** October 21st 2018
* **GUI Design and layout -** October 26th 2018
* **Coding Functionality** - October 28th 2018

**10. DELIVERABLE**

A usable online voting portal to conduct elections pan India, where in the voters can signup and login to vote for the people in their constituency.

**11. ENTRY CRITERIA**

Entry Criteria defines the inputs (those are requirements and any data inputs from previous phases) which should be available before entering a testing phase. So we have following entry criterias for various testing phases :

1. **Gui Testing -** 
   1. Flask and Python to be installed on the server system.
   2. The developer should have an browser.
   3. GUI should be made.
   4. User’s requirement for the GUI interface of the portal should be available.
2. **Unit Testing -**
   1. Each modules’ layout should be available for testing.
   2. Test case should be available for testing the modules.
3. **Integration Testing -** 
   1. A debugging technique should be available to debug any problem faced in integration of different modules.
4. **Functional Testing -** 
   1. List of all the functions along with their specific tasks should be available.
   2. Test values of the instances to be passed as function arguments should be available.
   3. Functional requirements list should be available.
5. **System Testing -**.
   1. A computer system with internet connection and a working browser should be available.
6. **Acceptance Testing -** 
   1. The portal with complete implementation of the expected functionality should be available to be tested by user.
   2. User’s functional requirements should be available to match the actual functionality with the expected functionality.
   3. A user should be available to give feedback.

**12. SUSPENSION CRITERIA**

Suspension criteria specifies when to suspend all or a portion of the testing activities. Testing might be interrupted by various factors namely-

* Unavailability of externally dependent system-Our Project required constant availability of internet during coding and testing phase
* Overlooked Bugs in the code - Bugs hampering the basic working of the app
* Crashing of the System - Due to errors in code.

**13. RESUMPTION CRITERIA**

Resumption criteria specifies when testing can resume after it has been suspended.Different situations to resume testing are-

* Externally Dependent System becomes available again.
* Debugging the code- After looking into the bugs and resolving them,app can be tested again
* Correcting the risk posing code- After checking where the implementation went wrong and working on it accordingly.

**14. EXIT CRITERIA**

Exit Criteria specifies conditions or on-going activities that should be fulfilled prior to completing the software testing phase.

The conditions are-

* Ensuring all Critical Test Cases are passed.
* Achieving complete Functional Coverage.
* Identifying and fixing all the high-priority defects.
* User Requirements are fulfilled.