Hopin Test Plan

**1. Introduction**

The Test Plan is designed to prescribe the scope, approach, resources, and schedule of all testing activities of the application

**1.1 Scope**

* In Scope

1. API Testing
2. UI Testing

* Out Of Scope

1. DB validation

## **1.2 Quality Objective**

1. Ensure the application under test conforms to functional and nonfunctional requirements
2. Ensure the AUT meets the quality specifications defined by the client

bugs/issues are identified and fixed before go live

**1.3 Roles & Responsibilities**

| Sr NO | Team Member | Task |
| --- | --- | --- |
|  | Manager | Signoff of Test Plan |
| 2 | Saumil Jain | Test Case Creation and Execution  API Automation  UI Automation  Test Deliverables |
|
|

**2. Test Scenarios**

| Sr No | Test Scenario | Comments |
| --- | --- | --- |
| 1 | Verify the API request and response and it follows all the API specs provided |  |
| 2 | Verify the User Interface it should follows all the user requirements |  |

**2.1 Test Cases**

| Sr No | Description | Component | Status |
| --- | --- | --- | --- |
| 1 | Send POST request with valid request body and headers  Expected behaviour: A successful response is returned  Actual behaviour: Successful response is returned | API | Executed |
| 2 | Send POST request with empty request body and headers  Expected behaviour: Request should fail  Actual behaviour: Successful response is returned | API | Executed |
| 3 | Send POST request with name as null in body  Expected behaviour: A successful response is returned  Actual behaviour: Successful response is returned | API | Executed |
| 4 | Send POST request with name as empty string in body  Expected behaviour: A successful response is returned  Actual behaviour: Successful response is returned | API | Executed |
| 5 | Send POST request with name as empty string in body  Expected behaviour: A successful response is returned  Actual behaviour: Successful response is returned | API | Executed |
| 6 | Verify successful Response  Expected behaviour: The response body should be correct  Actual behaviour:  In the response body `United Brands` does not contain contact info information.  The company size of `United Brands` is not according to the specs. The company size should be medium but it is displayed as Small in the response |  | Executed |
| 7 | Access `<http://localhost:3000/>` and verify we see the home screen with Welcome message, input screen and Submit button  Expected behaviour: Verify we are routed to correct screen  Actual behaviour: Verify we are routed to correct screen | UI | Executed |
| 8 | Access `<http://localhost:3000/>` and click on submit button with input field blank  Expected behaviour: Popup should display with text `Please provide your name`  Actual behaviour: Popup opens with text `Please provide your name` | UI | Executed |
| 9 | Access `<http://localhost:3000/>` and click on submit button with name. Verify we are routed to correct screen  With name timestamp and customer table  Expected behaviour: Verify we are routed to correct screen  Actual behaviour: Verify we are routed to correct screen | UI | Executed |
| 10 | Verify the customer tables contains all the details and columns  Expected behaviour: Verify the table includes Name, no of employees and Size of company  Actual behaviour: All columns exists | UI | Executed |
| 11 | Click on company name it should route to customer details screen and verify all information is present on the screen and the return button is present  Expected behaviour: All data is correct and the return button is present  Actual behaviour: All data is correct and the return button is present | UI | Executed |
| 12 | Click on back to the list to return to previous screen  Expected behaviour: The button should take the user to previous screen  Actual behaviour: The button takes the user to previous screen | UI | Executed |
| 13 | Click on company `United Brands` it does not include company info. It should take you to the screen with text `No contact info available`  Expected behaviour: It should display the above text  Actual behaviour: The UI is broken and the correct page is not displayed. It shows a page `TypeError: Cannot read property 'name' of undefined` | UI | Executed |

**2.2 Test Levels**

| Sr No | Test Level | Status |
| --- | --- | --- |
| 1 | Manual Testing | Done |
| 2 | API Automation | Done |
| 3 | UI Automation | Done |

**2.3 Bug Triage**

| Sr No | Description | Module (API/FrontEnd) | Status |
| --- | --- | --- | --- |
| 1 | Contact Info is missing for `United Brands` | API | OPEN |
| 2 | Size of the company is not according to the requirement for `United Brands`  According to requirements the size of the company where size > 10 should be medium but the size is returned as `Small` | API/Frontend | OPEN |
| 3 | When a customer doesn't have contact info, the message No contact info should be available but when the user click on `United Brands` the `United Brands` is not available | FrontEnd | OPEN |
| 4 | Sending POST request with empty body returns successful response | API | OPEN |
| 5 | Sending POST request with name as empty string is body returns successful response | API | OPEN |
| 6 | Sending POST request with name as null is body returns successful response | API | OPEN |

**2.4 Exit Criteria**

* Specifies the criteria that denote a **successful** completion of a test phase
* All test cases are successfully executed
* All bugs are resolved

## **3. Test Deliverables**

**3.1 Before testing phase**

* Test plans document.
* Test cases documents
* Test Design specifications.

**3.2 After the testing cycles is over**

* Test Results/reports
* Defect Report

**4. Testing Tools**

| Sr No | Type of Testing | Tools |
| --- | --- | --- |
| 1 | Manual Testing | Google Document, Postman, Chrome |
| 2 | API Automation | Axios, NodeJS, mocha and Chai |
| 3 | UI Automation | TestCafe |