Test Execution Document

For

Hybrid App Warehouse Management

Version 1.1 approved

Prepared by: Chaitanya Reddy

Chittoori Meha

Jallepalli Prerna

Madarapu Srikar

Krishna Priya

Organization: NIIT University

Table of Contents

1. **Introduction----------Page 1**
   - **1.1 Purpose----------Page 1**
   - **1.2 Testing Process----------Page 1**
2. **Unit Testing----------Page 2**
   - **2.1 Introduction----------Page 2**
   - **2.2 Unit 1 → JQuery Submit/Index----------Page 2**
   - **2.3 Unit 2 → JQuery DOM/ERP Products----------Page 5**
   - **2.4 Unit 3 → JQuery Onclick ERP Products----------Page 7**
   - **2.5 Unit 4 → JQuery DOM/Profile----------Page 9**
   - **2.6 Unit 5 → JQuery OnClick Product_Details----------Page 12**
   - **2.7 Unit 6 → JQuery DOM Product Details----------Page 14**
   - **2.8 Unit 7 → JQuery – Page.Events/Bar----------Page 17**
   - **2.9 Unit 8 → JQuery DOM/ Purchase Orders---------- Page 18**
   - **2.10 Unit 9 → JQuery DOM/ Tables---------- Page 21**
   - **2.11 Unit 10 → JQuery DOM/ Sales Orders---------- Page 24**
   - **2.12 Unit 11 → JQuery Page.Events/Pie---------- Page 27**
   - **2.13 Unit 12 → JQuery DOM/Pie---------- Page 29**
   - **2.14 Unit 13 → JQuery DOM/ Warehouse_mg----------Page 34**
   - **2.15 Unit 14 → JQuery DOM/ Bar----------Page 35**
   - **2.16 Unit 15 → JQuery – DOM/ Warehouse_details----------Page 38**
3. **Integration Testing---------- Page 42**
   - **3.1 Introduction----------Page 42**
   - **3.2 Integrate Part 1 → Pushing Data from Content to Scripts------Page 42**
   - **3.3 Integrate Part 2 → Posting data through AJAX to Scripts------Page 44**
   - **3.4 Integrate Part 3 → Database Connection via Scripts----------Page 46**
   - **3.5 Integrate Part 4 → Posting data through AJAX to Scripts------Page 48**
   - **3.6 Intgrate Part 5 → Appending the Data to Content----------Page 50**
4. **Bugs----------Page 52**
   - **4.1 #Bug 1----------Page 52**
   - **4.2 #Bug 2----------Page 52**

1. Introduction

1.1 Purpose

This **Test Execution Document** has been prepared for a Hybrid Application to Warehouse Management System. This has been developed by a team which has been mentioned above in NIIT University. This application/product that has been developed does not possess any hierarchy in terms of development. This document explicitly shows the results of the test that has been executed on the product. This document only includes the test cases in different forms. Testing of the product has been done mainly in two different ways.

- Unit Testing
- Integration Testing

1.2 Testing Process

As mentioned above the testing that has been executed can be widely divided in to two major parts which were mentioned above in the purpose (1.1).
Upon Unit Testing, complete product has been divided in to small units. All the units are tested separately. Each unit test cases can be broadly divided into black box test cases and white box test cases. Black box and white box is a type of deriving test cases for the unit. These entire tests are again described/ explained clearly with appropriate reasons.
Upon Integration Testing, complete product branches have been divided in to integrated parts. All these integrated parts are the addition of units only. So before undergoing integration testing it should have completed unit testing. Each integration test case can be broadly divided into black box test cases and white box test cases. Black box and white box is a type of deriving test cases. These entire tests are again described/ explained clearly with appropriate reasons.

2. Unit Testing

2.1 Introduction

Complete code of the product is divided in to small units which can be tested under blocks. All the test cases can be derived/ divided from black box and white box testing methods. Each unit will be having two section named black box and white box testing. Each section will be having number of test cases and will be described appropriately.

2.2 Unit 1 → Jquery – Submit/index

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	Submit	Valid Arguments(All fields)
Test Case 2	Submit	Invalid Arguments (Discrete fields)
Test Case 3	Submit	Invalid Arguments (All fields)
Test Case 4	Submit	Special/ Unpermitted Arguments. (All fields)

Test Case 1

Test Description: Submit functions take the values from the user and shows the appropriate page for appropriate user. Function behavior will be analyzed by giving different type of inputs which a user can supply for the function.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where all the arguments that are supplied are in valid to the function. Eg: (user1 – 1234, user2 – abed).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.

No BUGS and No Faulty output have been received.

Test Case 2

Description: Submit functions take the values from the user and shows the appropriate page for appropriate user. Function behavior will be analyzed by giving different type of inputs which a user can supply for the function.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where all the arguments that are supplied are in Invalid/ Valid to the function. Some of the values may be valid and some may be the other side. Eg: (user1 – notno, user3 – abed).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Description: Submit functions take the values from the user and shows the appropriate page for appropriate user. Function behavior will be analyzed by giving different type of inputs which a user can supply for the function.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where all the arguments that are supplied are in Invalid to the function. Eg: (nonuser – qwerty, nouser2 – check).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 4

Description: Submit functions take the values from the user and shows the appropriate page for appropriate user. Function behavior will be analyzed by giving different type of inputs which a user can supply for the function.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where the arguments that are not permitted to the user to use them. Eg: (user@1– 1234, user-2 – abed).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) there is no need of extra test cases to complete/ achieve the above mentioned coverage's. Now, we can conclude that the white box test cases have been executed as above.

2.3 Unit 2 → Jquery – DOM/ erp_products

Section 1 – Black Box Test Cases

Test Case	Function	Type

Test Case 1	DOM	Local Storages returns data
Test Case 2	DOM	Local Storages Manipulates Data
Test Case 3	DOM	Data flush

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages remains same without any change. Eg: (user_key = user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been manipulated. The data that have been stored have been replaced with some other values. Eg: (user_key = user2 where the user logged in is user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been flushed by the systems environment. That means the values does not exit which were needed to the function. Eg: (user_key =’empty’).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) most of the coverage’s have been done. White box test cases which were below listed are adding up to/ continuation the all mention above test cases (limit to unit2).

Test Case	Function	Type
Test Case 1	DOM	Error returning to user key.

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: The arguments that are needed to the function will be taking it from the local storages. Even though the value that is needed is returned to the function, there may be details of the key corrupted in the system. This test case deals in scenario where function returns an error in returning details.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

2.4 Unit 3 → Jquery – onClick/erp_products

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	onClick	Data extracted
Test Case 2	onClick	Not Able to extract data
Test Case 3	onClick	Data Manipulated in extraction
Test Case 4	onClick	None Done

Test Case 1

Test Description: onClick functions extract the value from the parent, stores to the local storages and redirects to the appropriate page. Behavior of the function is analyzed on different scenarios of the data flow into it and out of it.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All

the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that data have been successfully extracted from the parent and given to the function to continue further steps.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: onClick functions extract the value from the parent, stores to the local storages and redirects to the appropriate page. Behavior of the function is analyzed on different scenarios of the data flow into it and out of it.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that data is not being extracted from the parent. So this function needs to make necessary steps to continue.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: onClick functions extract the value from the parent, stores to the local storages and redirects to the appropriate page. Behavior of the function is analyzed on different scenarios of the data flow into it and out of it.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server

which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that data is extracted from the parent but the data extracted is not the appropriate to continue for the further steps. So this function needs to make necessary steps to continue.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 4

Test Description: onClick functions extract the value from the parent, stores to the local storages and redirects to the appropriate page. Behavior of the function is analyzed on different scenarios of the data flow into it and out of it.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that although this onClick function is triggered it is not able to continue the steps in that block. So this function needs to make necessary steps to continue.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

2.5 Unit 4 → Jquery – DOM/ profile

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	DOM	Local Storages returns data

Test Case 2	DOM	Local Storages Manipulates Data
Test Case 3	DOM	Data flush

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages remains same without any change. Eg: (user_key = user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been manipulated. The data that have been stored have been replaced with some other values. Eg: (user_key = user2 where the user logged in is user1).

Test Results:

Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been flushed by the systems environment. That means the values does not exit which were needed to the function. Eg: (user_key =’empty’).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:

White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) most of the coverage’s have been done. White box test cases which were below listed are adding up to/ continuation the all mention above test cases (limit to unit2).

Test Case	Function	Type
Test Case 1	DOM	Error returning to user key.

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: The arguments that are needed to the function will be taking it from the local storages. Even though the value that is needed is returned to the function, there may be details of the key corrupted in the system. This test case deals in scenario where function returns an error in returning details.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

2.6 Unit 5 → Jquery – onClick/product_details (3)

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	onClick	Data extracted
Test Case 2	onClick	Not Able to extract data
Test Case 3	onClick	Data Manipulated in extraction
Test Case 4	onClick	None Done

Test Case 1

Test Description: onClick functions extract the value from the parent, stores to the local storages and redirects to the appropriate page. Behavior of the function is analyzed on different scenarios of the data flow into it and out of it.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that data have been successfully extracted from the parent and given to the function to continue further steps.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: onClick functions extract the value from the parent, stores to the local storages and redirects to the appropriate page. Behavior of the function is analyzed on different scenarios of the data flow into it and out of it.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that data is not being extracted from the parent. So this function needs to make necessary steps to continue.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: onClick functions extract the value from the parent, stores to the local storages and redirects to the appropriate page. Behavior of the function is analyzed on different scenarios of the data flow into it and out of it.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that data is extracted from the parent but the data extracted is not the appropriate to continue for the further steps. So this function needs to make necessary steps to continue.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 4

Test Description: onClick functions extract the value from the parent, stores to the local storages and redirects to the appropriate page. Behavior of the function is analyzed on different scenarios of the data flow into it and out of it.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that although this onClick function is triggered it is not able to continue the steps in that block. So this function needs to make necessary steps to continue.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

2.7 Unit 6 → Jquery – DOM/ product_details

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	DOM	Local Storages returns data
Test Case 2	DOM	Local Storages Manipulates Data
Test Case 3	DOM	Data flush

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages remains same without any change. Eg: (user_key = user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices

should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been manipulated. The data that have been stored have been replaced with some other values. Eg: (user_key = user2 where the user logged in is user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been flushed by the systems environment. That means the values does not exit which were needed to the function. Eg: (user_key =’empty’).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:

White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) most of the coverage’s have been done. White box test cases which were below listed are adding up to/ continuation the all mention above test cases (limit to unit2).

Test Case	Function	Type
Test Case 1	DOM	Error returning to user key.

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: The arguments that are needed to the function will be taking it from the local storages. Even though the value that is needed is returned to the function, there may be details of the key corrupted in the system. This test case deals in scenario where function returns an error in returning details.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

2.8 Unit 7 → Jquery – Page.Events/Bar

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	Page.Events	Returning Variables
Test Case 2	Page.Events	Variable flush

Test Case 1

Test Description: Page.Events functions returns the necessary variables for the appropriate tags to the display content in the body through the function it has been referred. Returning variables are analyzed upon referring in different functions.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: This function is referred from different location of the body and content page and needs to check if it is returning or not. In this case we assume that it is going to return the appropriate values.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: Page.Events functions returns the necessary variables for the appropriate tags to the display content in the body through the function it has been referred. Returning variables are analyzed upon referring in different functions.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: This function is referred from different location of the body and content page and needs to check if it is returning or not. In this case we assume that it is going to return the appropriate values.

Test Results:
Instance 1
Result: FAILED
Output: Behavior is not appropriate to situation.
Bug #1 enrolled (Check Section 4 for further details)

Instance 2
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) there is no need of extra test cases to complete/ achieve the above mentioned coverage's. Now, we can conclude that the white box test cases have been executed as above.

2.9 Unit 8 → Jquery – DOM/ purchase_orders

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	DOM	Local Storages data updating
Test Case 2	DOM	Manipulation on data
Test Case 3	DOM	Stored Data flush

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages remains same without any change. Eg: (user_key = user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been manipulated. The data that have been stored have been replaced with some other values. Eg: (user_key = user2 where the user logged in is user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been flushed by the systems environment. That means the values does not exit which were needed to the function. Eg: (user_key =’empty’).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) most of the coverage's have been done. White box test cases which were below listed are adding up to/ continuation the all mention above test cases (limit to unit2).

Test Case	Function	Type
Test Case 1	DOM	Unpermitted return key

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: The arguments that are needed to the function will be taking it from the local storages. Even though the value that is needed is returned to the function, there may be details of the key corrupted in the system. This test case deals in scenario where function returns an error in returning details.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

2.10 Unit 9 → Jquery – DOM/ table

Section 1 – Black Box Test Cases

Test Case	Function	Type

Test Case 1	DOM	Valids from storages
Test Case 2	DOM	Manipulated valid from storages to the output
Test Case 3	DOM	Valids table fixation
Test Case 4	DOM	Valids table corrupt id on fixation
Test Case 5	DOM	Table dependencies

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a table from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid in respect to the function. Eg: (user_key = user1, product_key = product_value).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a table from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server

which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are manipulated with respect to the function. Eg: (user_key =interrupt, product_key = interrupt).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a table from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and checking it the data is fixed in form table to the appropriate section for the valid id.

Test Results:
Result: PASSED
Output: Function is displaying the required result in the appropriate section.
No BUGS and No Faulty output have been received.

Test Case 4

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a table from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and checking it the data is fixed in form table to the appropriate section for the corrupt id.

Test Results:
Result: PASSED
Output: Function is displaying the required result in the appropriate section.
No BUGS and No Faulty output have been received.

Test Case 5

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a table from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and data is fixed in form table to the appropriate section for the corrupt id. Checking whether dependencies are preloaded for the appropriate table.

Test Results:
Result: PASSED
Output: Function is displaying the required result in the appropriate section.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) there is no need of extra test cases to complete/ achieve the above mentioned coverage's. Now, we can conclude that the white box test cases have been executed as above.

2.11 Unit 10 → Jquery – DOM/ sales orders

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	DOM	Local Storages data updating
Test Case 2	DOM	Manipulation on data
Test Case 3	DOM	Stored Data flush

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages remains same without any change. Eg: (user_key = user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been manipulated. The data that have been stored have been replaced with some other values. Eg: (user_key = user2 where the user logged in is user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been flushed by the systems environment. That means the values does not exit which were needed to the function. Eg: (user_key =’empty’).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) most of the coverage's have been done. White box test cases which were below listed are adding up to/ continuation the all mention above test cases (limit to unit2).

Test Case	Function	Type
Test Case 1	DOM	Unpermitted return key

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: The arguments that are needed to the function will be taking it from the local storages. Even though the value that is needed is returned to the function, there may be details of the key corrupted in the system. This test case deals in scenario where function returns an error in returning details.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received

2.12 Unit 11 → Jquery – Page.Events/Pie

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	Page.Events	Returning Variables

Test Case 2	Page.Events	Variable flush

Test Case 1

Test Description: Page.Events functions returns the necessary variables for the appropriate tags to the display content in the body through the function it has been referred. Returning variables are analyzed upon referring in different functions.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: This function is referred from different location of the body and content page and needs to check if it is returning or not. In this case we assume that it is going to return the appropriate values.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: Page.Events functions returns the necessary variables for the appropriate tags to the display content in the body through the function it has been referred. Returning variables are analyzed upon referring in different functions.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: This function is referred from different location of the body and content page and needs to check if it is returning or not. In this case we assume that it is going to return the appropriate values.

Test Results:
Instance 1
Result: FAILED
Output: Behavior is not appropriate to situation.

Bug #1 enrolled (Check Section 4 for further details)

Instance 2
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) there is no need of extra test cases to complete/ achieve the above mentioned coverage's. Now, we can conclude that the white box test cases have been executed as above.

2.13 Unit 12 → Jquery – DOM/pie

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	DOM	Valids from storages
Test Case 2	DOM	Manipulated valid from storages to the output
Test Case 3	DOM	Valids pie fixation
Test Case 4	DOM	Valids pie corrupt id on fixation
Test Case 5	DOM	Pie dependencies

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a pie from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices

should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid in respect to the function. Eg: (user_key = user1, product_key = product_value).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a pie from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are manipulated with respect to the function. Eg: (user_key =interrupt, product_key = interrupt).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a pie from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All

the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and checking it the data is fixed in form pie to the appropriate section for the valid id.

Test Results:
Instance 1
Result: FAILED
Output: Charts are not able preloaded for the given data
Bug #2 enrolled (Check Appendix for further details)

Instance 2
Result: PASSED
Output: Charts Compatibility has been achieved for the application's environment.
No BUGS and No Faulty output have been received.

Test Case 4

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a pie from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and checking it the data is fixed in form pie to the appropriate section for the corrupt id.

Test Results:
Result: PASSED
Output: Function is displaying the required result in the appropriate section.
No BUGS and No Faulty output have been received.

Test Case 5

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a pie from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and data is fixed in form pie to the appropriate section for the corrupt id. Checking whether dependencies are preloaded for the appropriate table.

Test Results:
Result: PASSED
Output: Function is displaying the required result in the appropriate section.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) there is no need of extra test cases to complete/ achieve the above mentioned coverage's. Now, we can conclude that the white box test cases have been executed as above.

2.14 Unit 13 → Jquery – DOM/ warehouse_mg

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	DOM	Local Storages returns data
Test Case 2	DOM	Local Storages Manipulates Data
Test Case 3	DOM	Data flush

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages remains same without any change. Eg: (user_key = user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been manipulated. The data that have been stored have been replaced with some other values. Eg: (user_key = user2 where the user logged in is user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been flushed by the systems environment. That means the values does not exit which were needed to the function. Eg: (user_key =’empty’).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) most of the coverage’s have been done. White box test cases which were below listed are adding up to/ continuation the all mention above test cases (limit to unit2).

Test Case	Function	Type
Test Case 1	DOM	Error returning to user key.

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: The arguments that are needed to the function will be taking it from the local storages. Even though the value that is needed is returned to the function, there may be details of the key corrupted in the system. This test case deals in scenario where function returns an error in returning details.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

2.15 Unit 14 → Jquery – DOM/bar

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	DOM	Valids from storages
Test Case 2	DOM	Manipulated valid from storages to the output
Test Case 3	DOM	Valids bar fixation
Test Case 4	DOM	Valids bar corrupt id on fixation
Test Case 5	DOM	Bar dependencies

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a bar from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices

should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid in respect to the function. Eg: (user_key = user1, product_key = product_value).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a bar from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are manipulated with respect to the function. Eg: (user_key =interrupt, product_key = interrupt).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a bar from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All

the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and checking it the data is fixed in form bar to the appropriate section for the valid id.

Test Results:
Instance 1
Result: FAILED
Output: Charts are not able preloaded for the given data
Bug #2 enrolled (Check Appendix for further details)

Instance 2
Result: PASSED
Output: Charts Compatibility has been achieved for the application's environment.
No BUGS and No Faulty output have been received.

Test Case 4

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a bar from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and checking it the data is fixed in form bar to the appropriate section for the corrupt id.

Test Results:
Result: PASSED
Output: Function is displaying the required result in the appropriate section.
No BUGS and No Faulty output have been received.

Test Case 5

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. After getting the needed values then it generates a bar from the dynamic data which has been obtained from the Ajax call. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that the values to the function being supplied are valid and the function get the data from the Ajax call and data is fixed in form bar to the appropriate section for the corrupt id. Checking whether dependencies are preloaded for the appropriate table.

Test Results:
Result: PASSED
Output: Function is displaying the required result in the appropriate section.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) there is no need of extra test cases to complete/ achieve the above mentioned coverage's. Now, we can conclude that the white box test cases have been executed as above.

2.16 Unit 15 → Jquery – DOM/ warehouse_details

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	DOM	Valid returns from the local storages
Test Case 2	DOM	Data Manipulation over stored data
Test Case 3	DOM	Variables flush in storage

Test Case 4	DOM	Ajax fixation

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages remains same without any change. Eg: (user_key = user1).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been manipulated. The data that have been stored have been replaced with some other values. Eg: (user_key = user2 where the user logged in is user1).

Test Results:
Result: PASSED

Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values that are stored in local storages have been flushed by the systems environment. That means the values does not exit which were needed to the function. Eg: (user_key =’empty’).

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Test Case 4

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: All the types of inputs are divided in equivalence blocks. This test case deals with the approximation that values the values that have been stored are valid and get the approximate data. Now we need look over such that the received is fixed in appropriate content or not.

Test Results:
Result: PASSED

Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) most of the coverage's have been done. White box test cases which were below listed are adding up to/ continuation the all mention above test cases (limit to unit2).

Test Case	Function	Type
Test Case 1	DOM	Key values initiating the error

Test Case 1

Test Description: DOM function takes the values from the local storages that have been stored after the user validates his details. Output behavior will be analyzed upon the different cases of values.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser.

Test Inputs: The arguments that are needed to the function will be taking it from the local storages. Even though the value that is needed is returned to the function, there may be details of the key corrupted in the system. This test case deals in scenario where function returns an error in returning details.

Test Results:
Result: PASSED
Output: Function is returning appropriate result for the given arguments.
No BUGS and No Faulty output have been received.

3. Integration Testing

3.1 Introduction

After unit testing is completed, all the units are integrated (integration might have been done before, talking in terms of testing) to make the product full functioning. All these integrated parts are to be tested so that exchange of data between them, combine functioning of the functions. Test cases can be drawn in two different ways. Black box test cases, White box cases. Each section will be having number of test cases and will be described appropriately.

3.2 Integrate Part 1 → Pushing Data from Content to Scripts

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	Ajax	Appropriate Data Sent Over
Test Case 2	Ajax	Invalid Data Sent Over

Test Case 1

Test Description: Java script functions take the input values from the content (HMTL) and makes all the data variables as an object. This object is being sent to server script. We need to check how may the variables might reach jscript and analyze the behavior.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where all the data variables are sent in valid and received in valid to the JavaScript function.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are continued appropriately.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: Java script functions take the input values from the content (HMTL) and makes all the data variables as an object. This object is being sent to server script. We need to check how may the variables might reach jscript and analyze the behavior.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where all the data variables are sent and received in the respective functions might be Invalid.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are continued appropriately.
No BUGS and No Faulty output have been received.

Section 1 – White Box Test Cases

Test Case	Function	Type
Test Case 1	Ajax	Data interchanged/ manipulated while sent over

Test Case 1

Test Description: Java script functions take the input values from the content (HMTL) and makes all the data variables as an object. This object is being sent to server script. We need to check how may the variables might reach jscript and analyze the behavior.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: Even the data is sent is it cannot verify that the values for its functionality. This test case deals with data objects where any type of data should be adjusted by the appropriate function.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are continued appropriately.
No BUGS and No Faulty output have been received.

3.3 Integrate Part 2 → POSTing data through AJAX to scripts

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	Ajax	Valid Data object Sent
Test Case 2	Ajax	Data object being manipulated over sending
Test Case 3	Ajax	Data Lost Over Server or in between

Test Case 1

Test Description: Ajax need to POST the data to the PHP scripts that are residing in the server to continue the further steps or to get/ set the data in the database. Behavior of the combined parts is being analyzed over different type of objects sent.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where the data object that is being sent or sent over POST is valid to the appropriate functions.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are connected appropriately.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: Ajax need to POST the data to the PHP scripts that are residing in the server to continue the further steps or to get/ set the data in the database. Behavior of the combined parts is being analyzed over different type of objects sent.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where the data object that is being sent or sent over POST is manipulated before reaching the appropriate function that is to be executed.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are connected appropriately.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: Ajax need to POST the data to the PHP scripts that are residing in the server to continue the further steps or to get/ set the data in the database. Behavior of the combined parts is being analyzed over different type of objects sent.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where the data object that is being sent or sent over POST is completely lost or there is a connection failure over sending the data.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are connected appropriately.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) there is no need of extra test cases to complete/ achieve the above mentioned coverage's. Now, we can conclude that the white box test cases have been executed as above.

3.4 Integrate Part 3 → Database Connection via Script

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	PHP	Valid Credentials to server
Test Case 2	PHP	Invalid Credentials to server

Test Case 1

Test Description: Scripts needs to pass the credentials to sever to connect to the database, so as to get and set the values in the needed approximate tables. Connection depends on credentials that have been shared to the server. Behavior of the connection is analyzed on different inputs.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where valid credentials are sent over the server.

Test Results:
Result: PASSED
Output: Database reacted according the given input appropriately.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: Scripts needs to pass the credentials to sever to connect to the database, so as to get and set the values in the needed approximate tables. Connection depends on credentials that have been shared to the server. Behavior of the connection is analyzed on different inputs.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where Invalid credentials are sent over the server.

Test Results:
Result: PASSED
Output: Database reacted according the given input appropriately.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Test Case	Function	Type
Test Case 1	PHP	Connection Fail over network

Test Case 1

Test Description: Scripts needs to pass the credentials to sever to connect to the database, so as to get and set the values in the needed approximate tables. Connection depends on credentials that have been shared to the server. Behavior of the connection is analyzed on different inputs.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: What might be the credentials given to the server if the connection over network fails we need to make the behavior of the function is appropriate to the situation.

Test Results:
Result: PASSED

Output: Database reacted according the given input appropriately.
No BUGS and No Faulty output have been received.

3.5 Integrate Part 4 → POSTing data through AJAX to scripts

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	Ajax	Valid Data object Sent from Script
Test Case 2	Ajax	Ripped data object received.
Test Case 3	Ajax	Incomplete data received.

Test Case 1

Test Description: Ajax need to get the data from the PHP scripts that are residing in the server to continue the further steps in the content. Behavior of the combined parts is being analyzed over different type of objects received.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where the data object that is being received is valid and have all the data that is required to complete the function.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are connected appropriately.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: Ajax need to get the data from the PHP scripts that are residing in the server to continue the further steps in the content. Behavior of the combined parts is being analyzed over different type of objects received.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where the data object that is being received is ripped off and not compatible to the committed query.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are connected appropriately.
No BUGS and No Faulty output have been received.

Test Case 3

Test Description: Ajax need to get the data from the PHP scripts that are residing in the server to continue the further steps in the content. Behavior of the combined parts is being analyzed over different type of objects received.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where the data object that is being received is incomplete and the data is not sufficient to the function to complete the query.

Test Results:
Result: PASSED
Output: 'X' type of Data is being received and further steps are connected appropriately.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Note:
White Box test cases included, where for a given unit of code path coverage, branch coverage, statement coverage must have been achieved by the derived set of test cases. By the above test cases which were covered (Black box test cases) there is no need of extra test cases to complete/ achieve the above mentioned coverage's. Now, we can conclude that the white box test cases have been executed as above.

3.6 Integrate Part 5 → Appending the Data to Content

Section 1 – Black Box Test Cases

Test Case	Function	Type
Test Case 1	Jquery	Valid Data Supplied
Test Case 2	Jquery	Invalid Data Supplied

Test Case 1

Test Description: After getting the necessary data from the server this is to be appended to the content in the appropriate section so that it makes everything visible to the user what is undergoing. This data is to be analyzed such that it shows only the appropriate and needed data to the user.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where valid data and valid section id is received. This is to be appended to the content and verified.

Test Results:
Result: PASSED
Output: Needed Data received and necessary continuation has been undergone.
No BUGS and No Faulty output have been received.

Test Case 2

Test Description: After getting the necessary data from the server this is to be appended to the content in the appropriate section so that it makes everything visible to the user what is undergoing. This data is to be analyzed such that it shows only the appropriate and needed data to the user.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: All the types of inputs are divided in equivalence blocks. This case deals with the block where valid data and valid section id might not be received. Upon this the appending behavior is to be analyzed.

Test Results:
Result: PASSED
Output: Needed Data received and necessary continuation has been undergone.
No BUGS and No Faulty output have been received.

Section 2 – White Box Test Cases

Test Case	Function	Type
Test Case 1	Jquery	Error/ Insufficient Data

Test Case 1

Test Description: After getting the necessary data from the server this is to be appended to the content in the appropriate section so that it makes everything visible to the user what is undergoing. This data is to be analyzed such that it shows only the appropriate and needed data to the user.

Test Approach: Testing should be conducted on different android devices, IOS devices, web interfaces with different resolutions. The application is developed on Phonegap CLI Platform. All the devices where the testing should be undergone should be able to connect to the local server which was created by the Phonegap CLI. For connecting to the server Android devices, IOS devices should have phonegap developer App. For testing web interface appropriate devices should have a standard browser. As to send the data separately we need external clients like postman to make it possible.

Test Inputs: Data has been undergone so many phase before appending. There is a probability that after so many restrictions may be error data of insufficient data might be received to the content. This behavior is to be analyzed.

Test Results:
Result: PASSED
Output: Needed Data received and necessary continuation has been undergone.
No BUGS and No Faulty output have been received.

4. Bugs

4.1 Bug #1

Bug Description: Invalid variables are returned by the function pages.event if the variables are flushed. This has been resolved by adding the buffer to the current variable existing in the content and the variables are exchanged on new arrival.

Bug Status: RESOLVED

4.2 Bug #2

Bug Description: After data is retrieved form the ajax calls that is to be fixed in the content. Some of the devises are not compatible for this like IOS and other. This has been resolved by adding page roles which affected them load after loading the dependencies needed from the generation of them.

Bug Status: RESOLVED