**Name:** PATEL UMME AIMAN ABDUL WAHID..

**Roll No:** 22DC005 **Batch:** 1 **Subject:** SE

**Experiment No. 09**

**Aim:** To write test cases for white box testing.

**Theory:**

**White Box Testing: -**

White box testing techniques analyse the internal structures the used data structures, internal design, code structure and the working of the software rather than just the functionality as black box testing. It is also called glass box testing or clear box testing or structural testing.

Working process of white box testing:-

* Input: Requirements, Functional specification, design documents, source code.
* Processing: Performing risk analysis for guiding through the entire process.
* Proper test planning: Designing test cases to cover entire code. Execute rinse repeat until error-free software is reached. Also, the results are communicated.
* Output: Preparing final report of the entire testing process.

**Testing Techniques:**

**Statement Coverage:**

In this technique, the aim is to traverse all statement at least once. Hence, each line of code is tested. In case of a flowchart, every node must be traversed at least once.

Since all lines of code are covered, helps in pointing out faulty code.

**Branch Coverage:**

In this technique, test cases are designed so that each branch from all decision points are traversed at least once. In a flowchart, all edges must be traversed at least once. 4 test cases required such that all branches of all decision are covered, i.e. all edges of flowchart are covered.

**Condition Coverage:**

In this technique, all individual conditions must be covered as shown in the following example:

* READ X, Y
* IF (X=0 || Y==0)
* PRINT ‘0’

In this example, there are 2 conditions: X==0 and Y = 0. Now, test these conditions get TRUE and FALSE as their values. One possible example would be:

* #TC1 - X= 0, Y=55 #TC2 - X= 5, Y=0

**Multiple Condition Coverage:**

In this technique, all the possible combinations of the possible outcomes of conditions are tested at least once. Let's consider the following example:

* READ X. Y
* IF (X=0 || Y==0)
* PRINT ‘0’
* #TC1: X=0,Y=00 #TC2: X=0, Y=5 #TC3: X=55, Y=0
* #TC4: X= 55, Y= 5

Hence, four test cases required for two individual conditions. Similarly, if there are n conditions then 2^n test cases would be required.

**Basis Path Testing:**

In this technique, control flow graphs are made from code or flowchart and then Cyclomatic complexity is calculated which defines the number of independent paths so that the minimal number of test cases can be designed for each independent path.

**Loop Testing:**

Loops are widely used and these are fundamental to many algorithms hence, their testing is very important. Errors often occur at the beginnings and ends of loops.

**Test Cases For AIKTC Training & Placement Website:**

| **Project Title: FUNDRAISING WEBSITE FOR EDUCATION.** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case 1** | | | | | | | |
| **Test Case ID:** 01 | | | | **Test Designed By:** PATEL MUSKAN KASIM SHAH. | | | |
| **Test Priority:** High | | | | **Test Design Date:** 22/07/2023 | | | |
| **Module Name:** Database Interaction Module | | | | **Test Execution Date:** 24/07/2023 | | | |
| **Test Title:** Verify that data is stored in the database. | | | | **Test Executed By:** Aiman Patel | | | |
| **Test Description:** Verify that new users are stored accurately in the database. | | | |  | | | |
| **Pre-condition:** The database is properly configured and accessible. | | | | | | | |
|  | | | | | | | |
| **Steps** | **Test Steps** | **Test Data** | **Expected Result** | | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Create a new  user/ sign in. | username, password, gmail id. | Data is  stored  accurately in the  respective database tables. | | Data was  successfully  stored in the database tables. | Pass | N/A |
| **Post Conditions:** The database remains in a consistent state after the storage operations. | | | | | | | |

| **Project Title: FUNDRAISING WEBSITE FOR EDUCATION.** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case 2** | | | | | | | |
| **Test Case ID:** 02 | | | | **Test Designed By:** AIMAN PATEL | | | |
| **Test Priority:** High | | | | **Test Design Date:** 25/07/2023 | | | |
| **Module Name:** Authentication and Authorization Module | | | | **Test Execution Date:** 27/07/2023 | | | |
| **Test Title:** Authentication and Authorization Test. | | | | **Test Executed By:** MUSKAN PATEL | | | |
| **Test Description:** Test the authentication and authorization mechanisms to ensure they correctly restrict access to authorized users only. | | | |  | | | |
| **Pre-condition:** The authentication and authorization mechanisms are in place and properly configured. | | | | | | | |
|  | | | | | | | |
| **Steps** | **Test Steps** | **Test Data** | **Expected Result** | | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Attempt to access a protected resource without proper authentication. | Invalid credentials  (username, password) | Access is denied  without  proper  authenticatio  n. | | Access denied  without  proper  authenticatio  n. | Pass | Additional security testing  should be conducted |
|  |  |  |  | |  |
| 2 | Use invalid credentials to authenticate. | Valid credentials  (username, password) | Authenticati on and  access are both successful. | | Authenticati on and  access both successful as expected. |  | to ensure  robustness against  various attack scenarios. |
| 3 | Use valid credentials for an authorized user. |
|  | | | | | | | |
| **Post Conditions:** The database remains in a consistent state after the storage operations. | | | | | | | |

| **Project Title: FUNDRAISING WEBSITE FOR EDUCATION.** | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case 3** | | | | | | | |
| **Test Case ID:** 03 | | | | **Test Designed By:** MUSKAN PATEL | | | |
| **Test Priority:** High | | | | **Test Design Date:** 15/08/2023 | | | |
| **Module Name:** Error Handling Module | | | | **Test Execution Date:** 20/08/2023 | | | |
| **Test Title:** Error Handling Test. | | | | **Test Executed By:** AIMAN PATEL | | | |
| **Test Description:** Testing system's response to unexpected Inputs and Errors. | | | |  | | | |
| **Pre-condition:** The system remains in a consistent state. | | | | | | | |
|  | | | | | | | |
| **Steps** | **Test Steps** | **Test Data** | **Expected Result** | | **Actual Result** | **Status (Pass/Fail)** | **Notes** |
| 1 | Submit a form with incomplete/m issing mandatory fields. | Incomplete or missing data for mandatory fields. | Error message for  incomplete  /missing data is displayed. | | Error message  displayed for  incomplete/ missing data. | Pass | Further testing with  edge cases  is  recommend  ed to ensure  thorough error  handling validation. |
| 2 | Submit a request with unauthorized user credentials. | Unauthorized user credentials. | Access denied  error  message is displayed. | | Access denied error message displayed. |
| 3 | Submit a request with unauthorized user credentials. | Invalid URL or endpoint. | Error message  for invalid  URL/endp oint is displayed. | | Error message  displayed  for invalid  URL. |
|  | | | | | | | |
| **Post Conditions:** The system remains in a consistent state. | | | | | | | |

**Conclusion:**

In this experiment, we employed white-box testing techniques to thoroughly examine the internal logic, structure, and pathways of the **f**undraising website for education.The primary goal was to ensure the correctness of the code, uncover hidden defects, and verify that the system adheres to the specified requirements.