EXPERIMENT NO.:08

Date of Performance:

Date of Submission:

Aim: Write test cases for white box testing

Software Used: **: Selenium/GitHub/Jira**

Theory:-

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

White box testing which also known as glass box is **testing, structural testing, clear box testing, open box testing and transparent box testing**. It tests internal coding and infrastructure of a software focus on checking of predefined inputs against expected and desired outputs. It is based on inner workings of an application and revolves around internal structure testing. In this type of testing programming skills are required to design test cases. The primary goal of white box testing is to focus on the flow of inputs and outputs through the software and strengthening the security of the software.

The term 'white box' is used because of the internal perspective of the system. The clear box or white box or transparent box name denote the ability to see through the software's outer shell into its inner workings.

Developers do white box testing. In this, the developer will test every line of the code of the program. The developers perform the White-box testing and then send the application or the software to the testing team, where they will perform the black box testing and verify the application along with the requirements and identify the bugs and sends it to the developer.

The developer fixes the bugs and does one round of white box testing and sends it to the testing team. Here, fixing the bugs implies that the bug is deleted, and the particular feature is working fine on the application.

Here, the test engineers will not include in fixing the defects for the following reasons:

- Fixing the bug might interrupt the other features. Therefore, the test engineer should always find the bugs, and developers should still be doing the bug fixes.
- If the test engineers spend most of the time fixing the defects, then they may be unable to find the other bugs in the application.

Techniques Used in White Box Testing

Data Flow Testing	This technique emphasizes verifying the correct initialization and retention of variables used in calculations. For instance, when dealing with product pricing, it's crucial to check that the variable storing the product price is correctly initialized before any computations (like discounts or totals). This ensures that the application accurately captures and maintains the initial input, preventing errors in further calculations.
Control Flow Testing	Control flow testing examines the program's execution paths based on user interactions. For example, when a user searches for a product, this test verifies that the correct functions are triggered and that the application follows the intended search logic. This ensures that users receive the expected results based on their input, and all paths through the search functionality are validated.
Branch Testing	Similar to control flow testing, branch testing focuses specifically on decision points within the code. When a user searches for a product, branch testing ensures that all possible branches in the search logic are executed. For instance, if the search can yield results, return no results, or produce an error, this test validates that each branch is handled correctly, providing comprehensive coverage of the search functionality.
Statement Testing	This technique ensures that every executable statement in a function is executed at least once. In the context of the price calculation function, statement testing verifies that all lines of code involved in calculating product prices are executed during testing. This guarantees complete coverage and helps identify any unreachable code or logic errors.
Decision Testing	Decision testing evaluates the correctness of the application at decision points, especially when determining outcomes like discount eligibility. For example, the website may offer discounts based on user status. This test ensures that all logical paths are assessed, confirming that the application correctly evaluates the conditions for granting discounts and that users receive the appropriate pricing based on their status.

Let us consider the following code :

INPUT A & B

C = A + B

IF C>100

PRINT "ITS DONE"

Now in the first, line, we assign the value of A and B. Let us suppose A = 60 and B = 50. Moving on to the second line, now C is assigned a value of A+B, here A = 60 and B = 50, hence C = 110. Moving on to the third line, we will check if C > 100, here the condition istrue and hence we should get our result as ITS DONE

Conclusion : Thus we have written test cases for black box testing .

Sign and Remark:

R1	R2	R3	Total Marks	Signature
(5)	(5)	(5)	(15)	