**🎯 MODULE 2: MANUAL TESTING 🎯**

**Que.1) What is Error, bug, defect and failure?**

**Ans.** A mistake in coding is called **error**, error found by tester is called **defect**, defect accepted by development team then it is called **bug**, build does not meet the requirements then it is called **failure**.

**Que.2) What is Exploratory testing?**

**Ans.** Exploratory Testing is an unscripted, hands-on approach to software testing where testers dynamically design, execute, and adapt tests in real-time based on their intuition, experience, and observations.

➢Unlike scripted testing (where predefined test cases are followed), exploratory testing emphasizes learning, creativity, and flexibility to uncover unexpected defects.

**Que.3) What is traceability matrix?**

**Ans.** A Traceability Matrix is a table that connects different project items (like requirements, test cases, and bugs) to ensure nothing is missed.

➢It helps track if all requirements are tested and if all tests cover the right requirements.

**Que.4) What is boundary value testing?**

**Ans.** Boundary value testing is a software testing technique where test cases are designed to check the behaviour of a system at the edges (boundaries) of input ranges.

**Que.5) What is Equivalence partitioning testing?**

**Ans.** Equivalence Partitioning is a software testing technique where input data is divided into groups (partitions) that are expected to behave the same way. Instead of testing every possible input, you test one representative value from each group to save time while maintaining coverage.

**Que.6) What is Integration testing?**

**Ans.** Integration testing is a level of software testing process where individual units are combined and tested as a group.

**Que.7) What is determines the level of testing?**

**Ans.** Risk is a factor that could result in future negative consequences; usually expressed as impact and likelihood.

➢There are two types of Risk

1. Project Risk
2. Product Risk

**Que.8) What is Component(Unit) testing?**

**Ans.** A minimal software item that can tested in isolation.it means “A unit is the smallest testable part of software.”

➢Sometimes known as **Unit testing, Module testing or Program testin**g.

**Que.9) What is functional system testing?**

**Ans.** Testing based on an analysis of the specification of the functionality of a component or system.

➢‘**Specification**’ – E.g. Requirements specification, Use Cases, Functional specification or maybe undocumented. ‘**Function**’ – what the system does Functional test – based on the Functions and features – may be applied at all Test levels (e.g. Component Test, System Test etc.)

**🎯** Functional Testing Types:-

1. Black Box Testing
2. White Box Testing
3. Experience based Testing
4. Smoke Testing
5. Sanity Testing
6. End to End Testing

**Que.10) What is Non-functional testing?**

**Ans.** Testing the attributes of a component or a system that do not relate to functionality. For example. Reliability, efficiency, usability, interoperability, maintainability and portability.

**🎯** Non Functional Testing Types:-

1. Usability Testing
2. Compatibility Testing
3. GUI Testing
4. Security Testing
5. Performance Testing
6. Stress Testing
7. Load Testing

**Que.11) What is Adhoc testing(Error Guessing)?**

**Ans.** Adhoc testing is an informal testing type with **an aim to break the system**.

➢ Main aim of this testing is to find defects by random checking. Adhoc testing can be achieved with the testing technique called **Error Guessing**.

➢This testing is primarily performed **if the knowledge of testers in the system under test is very high.**

**Que.12) What is black box testing? What are the different black box testing techniques?**

**Ans.** Testing, either functional or non-functional, without reference to the internal structure of the component or system.

➢**Specification-based testing technique** is also known as ‘**black-box**’ or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.

➢The testers **have no knowledge of how the system or component is structured inside the box.** In black-box testing the tester is concentrating on what the software does, not how it does it.

**🎯** Black Box Testing techniques:

1. Equivalence Partitioning
2. Boundary Value Analysis (B.V.A.)
3. Decision Table Testing
4. State Transition Testing

**Que.13) Mention what are the categories of defects.**

**Ans.** Based on Severity

1. Critical Defect
2. Major Defect
3. Minor Defect
4. Trivial Defect

**🎯** Based on Priority

1. High Priority
2. Medium Priority
3. Low Priority

**Que.14) Mention what Big Bang testing is?**

**Ans.** Big Bang Testing is an integration testing approach where all or most modules of a system are combined at once and tested as a single unit, *without* incremental integration.

**🎯** For Example:

➢A team builds a basic calculator app and tests all buttons (add, subtract, etc.) together instead of checking each operation step-by-step.

**Que.15) What is the purpose of exit criteria?**

**Ans.** Exit Criteria are predefined conditions or standards that must be met before testing can be formally concluded. They ensure that the software is thoroughly validated and ready for release.

**Que.16) What is white box testing and list the types of white box testing?**

## **Ans.** Testing based on an analysis of the internal structure of the component or system.

➢ Structure-based testing technique is also known as ‘white-box’ or ‘glass-box’ testing technique because here the testers require knowledge of how the software is implemented, how it works.

➢White Box Testing (also called Clear Box or Structural Testing) is a software testing method where the tester examines the internal code, structure, and logic of an application to design test cases.

🎯 White Box Testing List:-

1. Unit Testing
2. Static Code Analysis
3. Statement Coverage
4. Branch Coverage
5. Path Coverage
6. Loop Testing
7. Mutation Testing
8. Integration Testing (White-Box Approach)

|  |  |  |
| --- | --- | --- |
| **QA** | **QC** | **Tester** |
| QA is stands for “Quality Assurance” | QC is stands for “Quality Control” | Tester. |
| QA is **process** oriented activities. | QC is **product** oriented activities. | QC is **product** oriented activities. |
| It is a Preventive activities. | It is a Corrective process. | It is a Preventive activities. |
| It is a subset of Software life cycle(STLC). | QC can be considered as the subset of Quality Assurance. | Testing is the subset of Quality Control. |
| Process audits, reviews, risk management, standards creation | Test case execution, defect identification and reporting | Writing and executing test cases, reporting defects |
| Creating testing strategies, defining processes | Functional testing, performance testing | Verifying features, running test cases, reporting bugs |

**Que.17) What is the difference between QA v/s QC v/s Tester**

**Que.18) What is the difference between Verification and Validation?**

|  |  |  |
| --- | --- | --- |
| **No.** | **Verification** | **Validation** |
| 1 | Verification is a process which is performed at development level | Validation is a process which is performed at testing level |
| 2 | Verification is a static testing | Validation is a dynamic testing |
| 3 | Bugs can be found during the process of development | Bugs can only be found after the process of development |
| 4 | It is used to prevent errors. | It is used to detect errors |
| 5 | Verification activities are Reviews and Inspections. | Validation activity is Testing. |
| 6 | Verification can be achieved by asking “Are you building a product right?” | Validation can be achieved by asking “Are you building a right product?” |

**Que.19) What is the difference between Smoke and Sanitary testing?**

|  |  |  |
| --- | --- | --- |
| **No.** | **Smoke Testing** | **Sanitary Testing** |
| 1 | Smoke testing is performed after receiving software build to ascertain that the critical functionalities of program is working fine. | Sanitary testing is after receiving software build with minor changes in code or functionality sanitary testing performed to ascertain that the bugs have been fixed. |
| 2 | Smoke testing is documented & scripted. | Sanity testing is not documented & Unscripted. |
| 3 | Smoke testing is performed by either developers or testers. | Sanity testing is normally performed by testers. |
| 4 | Smoke testing is done by measure the stability of the system. | Sanity testing is done by measure the rationality of the system. |
| 5 | For Example. Smoke testing is General Health Checkup. | For Example. Smoke testing is Specialized Health Checkup. |

**Que.20) What is the difference between STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?**

|  |  |  |
| --- | --- | --- |
| **No.** | **STLC** | **SDLC** |
| 1 | SDLC is mainly related to software development. | SDLC is mainly related to software testing. |
| 2 | STLC is a sequential different activities performed during software testing process. | SDLC is a structure imposed on development of software product it’s defines process of planning, analysis, design, Implementation and ongoing maintenance and support. |
| 3 | STLC involves total five phases or steps. | STLC involves total six phases or steps. |
| 4 | STLC phases are performed after SDLC phases. | SDLC phases are completed before the STLC phases. |
| 5 | It helps in developing good quality software. | It helps in making the software defects free. |

**Que.21) What is Alpha Testing?**

**Ans.** Alpha Testing is an early real-world testing phase where the software is tested internally (by developers or QA teams) before releasing it to external users. It happens in a controlled environment, often at the developer’s site.

**Que.22) What is Beta Testing?**

**Ans**. Beta Testing is a real-world testing phase where a near-final version of the software is released to a limited group of external users (outside the company) to gather feedback before the official launch.

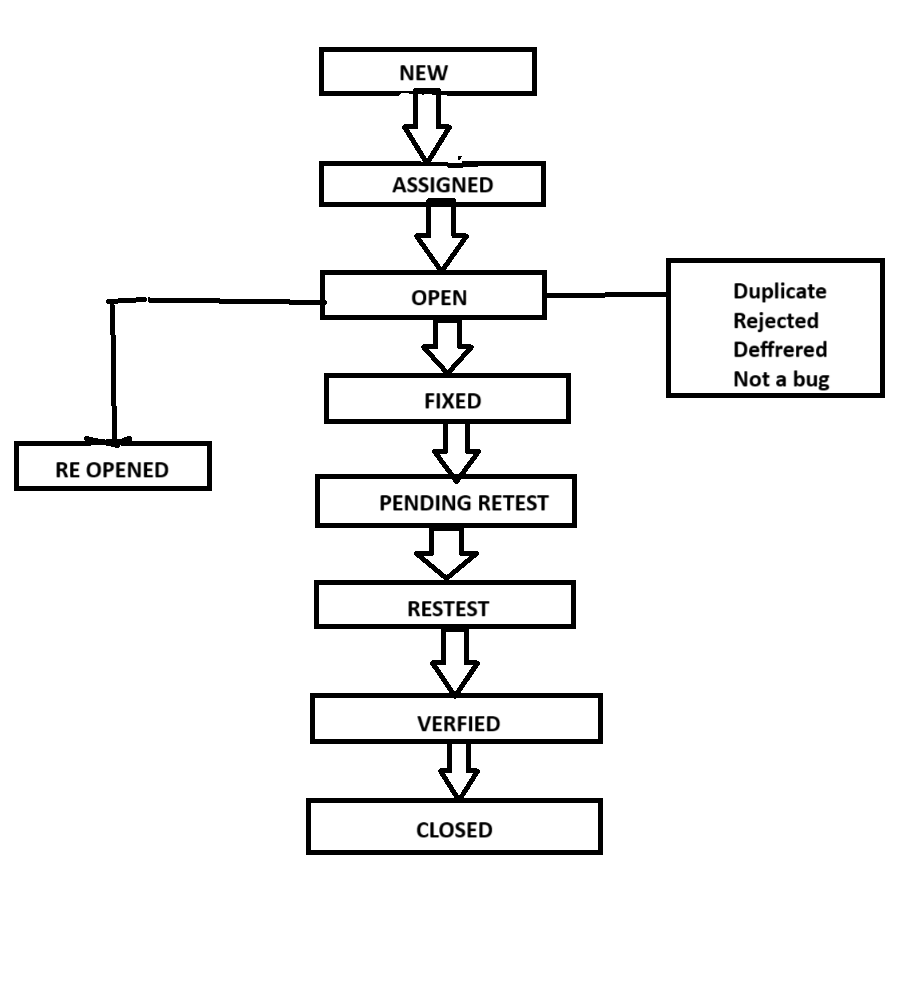
**Que.23) What is the difference between Severity and Priority?**

**Ans**.

|  |  |  |
| --- | --- | --- |
| **No.** | **Saverity** | **Priority** |
| 1 | How serious the impact of a defect is on functionality or performance. | How urgent is to fix the defect. |
| 2 | Technical impact of the bug. | Business urgency or importance. |
| 3 | Usually set by testers or QA team. | Usually set by product managers or clients. |
| 4 | Stability, functionality, or performance. | Development or release timeline. |
| 5 | May not be fixed immediately if priority is low. | Fixes may happen quickly even if severity is low. |
| 6 | App crashes on login → High Severity | Typo on homepage of a major site → High Priority |

**Que.24) What is bug life cycle?**

**Ans**. The bug life cycle describes the stages a defect goes through from discovery to resolution. It ensures proper tracking and closure of bugs in a systematic way.



**Que.25) What is priority?**

**Ans.** Priority refers to how soon a bug or test case should be fixed or executed, based on business or project urgency.

**Que.26) What is Saverity?**

**Ans.** Severity describes the extent to which a defect affects the system. It is a technical measure of how badly the system is broken because of the bug.

**Que.27) Explain types of performance testing?**

**Ans.** There are various types of performance testing in following below:-

1. Load testing
2. Stress testing
3. Endurance testing
4. Spike testing
5. Volume testing
6. Scalability testing

**In brief:**

1. **Load testing :-** To check how the system behaves under an expected load.

➢ Example: Testing a website with 1,000 users logging at the same time.

1. **Stress testing** **:-** To determine the system’s breaking point by pushing it beyond normal load limits.

➢ Example: Doubling the expected user load to see if the system crashes.

1. **Endurance testing :-** To test the system’s performance over an extended period under a normal load.

➢ Example: Running a test with 500 users for 24 hours.

1. **Spike testing :-** To test how the system reacts sudden, large spikes in user load.

➢ Example: Evaluate the system's ability to handle sharp changes in traffic.

1. **Volume Testing (Flood Testing)** :- To test the system’s ability to handle a large volume of data.

➢ Example: Inputting a huge database file into the application.

1. **Scalability testing :-** To determine how well the system scales when resources(like users, data or hardware) are increased.

**Que.28) When should “Regression Testing” be perfomed?**

**Ans.** Regression Testing ensures that new code changes (like bug fixes, updates, or features) do not break existing functionality.

It should be performed in these scenarios:

1. After Code Changes
2. During Integration
3. Environment/Config Updates
4. Release Cycles
5. Periodic/Scheduled Runs

**Que.29) What is the difference between Functional and Non-Functional testing?**

**Ans.**

|  |  |  |
| --- | --- | --- |
| **No.** | **Functional Testing** | **Non-Functional Testing** |
| 1 | Functional testing is testing based on analysis of specification of the functionality of a component. | Non-functional testing is testing the attributes of the component that that do not relate to the functionality. |
| 2 | **Easy** to do manual testing. | **Tough** to do manual testing. |
| 3 | Functional testing is **executed first**. | Non-functional testing should be **performed after** functional testing. |
| 4 | It tests what **the product does**. | It tests how the **good product work**. |
| 5 | It is **based on requirements** of customer. | It is **based on expectations** of customers. |
| 6 | Types of functional testing:  - **Black box** testing **- White box** testing **- Sanity** testing - **Smoke** testing | Types of Non-Functional testing: - **Performance** testing - **Usability** testing - **GUI** testing - **Load** testing |

**Que.30) To create HLR & Test case of**   
**1) (Instagram , Facebook) only first page and chat functionality**

**2) Facebook Login Page :** **https://www.facebook.com/**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see.

**Que.31) What is the difference between test scenarios, test cases, and test script?**

**Ans.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Aspect** | **Test Scenario** | **Test Case** | **Test Script** |
| **Level** | High-level | Medium to detailed | Low-level (automation code) |
| **Format** | Plain language | Structured steps | Programming/scripting code |
| **Purpose** | Identify areas to test | Describe how to test those areas | Automate the execution of test cases |
| **Who writes it** | Testers, Bas | Testers | Automation engineers |

**Que.32) Explain what Test Plan is? What is the information that should be covered.**

**Ans.** A Test Plan is a detailed document that outlines the strategy, objectives, scope, resources, schedule, and activities needed to perform testing for a software project. It acts as a blueprint for how testing will be conducted and managed.

**🎯 Scope**: Define what will and will not be tested, including in-scope and out-of-scope items.

**🎯 Objectives**: Clearly state what the testing effort aims to achieve, such as validating functionality and ensuring requirements are met.

**🎯 Approach**: Outline the testing strategy, methodologies, and tools to be used.

**🎯 Resources**: Identify the necessary resources, including personnel, tools, and equipment, for testing.

**🎯 Schedule**: Establish a timeline for testing activities, including milestones and deadlines.

**🎯 Deliverables**: List the documents and facts to be produced during testing, such as test cases and reports.

**Que.33) What is Adhoc testing?**

**Ans.** Adhoc Testing is an informal testing type, with an **Aim to break the system.**

➢ Unplanned testing approach where testers randomly check the software without predefined test cases or documentation. It relies on the tester's experience, intuition, and creativity to find hidden bugs.

**Que.34) What is Load testing?**

**Ans.** Load Testing is a type of performance testing that checks how a system behaves under expected or high user traffic.

➢The goal is to identify bottlenecks (like slow responses or crashes) before real users experience them.

➢ Example: Doubling the expected user load to see if the system crashes.

**Que.35) What is Stress testing?**

**Ans.** Stress Testing is a type of performance testing that pushes a system beyond its normal limits (e.g., extreme user traffic, high data volumes, or limited resources) to see how it handles overload and recovers from failure.

➢ Example: Testing a website with 1,000 users logging at the same time.

**Que.36) What is GUI testing?**

**Ans.** GUI Testing (Graphical User Interface Testing) checks the visual elements of a software application to ensure they look, behave, and interact correctly with users. It focuses on buttons, menus, layouts, fonts, colours and responsiveness.

➢Example: Web Based Testing & Desktop Based Testing, Mobile based testing.

**Que.37) Adavantage of Bugzilla.**

**Ans. Deadlines:** To fix the bugs, deadlines can be established.

**🎯 Types:** It reports in a variety of formats and types.

**🎯 Request System:** You can use the 'request system' provided by Bugzilla to ask other users to evaluate codes, provide information and other things.

**🎯 Flexible:** Bugzilla is quite flexible, so you can modify it to fit your unique process and requirements.

**🎯 Bug tracking tool:** Bugzilla is extremely good at monitoring and handling bugs and issues.

**Que.38) Bugs Categories are…**

**Ans. Functional Bugs:** These occur when a software component fails to perform its intended function correctly. For example, a button might not submit a form as expected.

**🎯 Logical Bugs:** These result from flaws in the design or implementation of the software's logic, leading to incorrect behavior.

**🎯 Usability Bugs:** These affect the user experience, making it difficult or confusing for users to interact with the application.

**🎯 Security Bugs:** These pose risks to the security of the software and its data, potentially leading to unauthorized access or data breaches.

**🎯 Performance Bugs:** These impact the speed, efficiency, or responsiveness of the software.

**🎯 System-Level Integration Bugs:** These occur when different software components or modules fail to work together seamlessly.

**🎯 Unit-Level Bugs:** These occur within individual code units or modules of the software.

**🎯 Syntax Bugs:** These arise from errors in the syntax or structure of the software code.

**🎯 Workflow Bugs:** These relate to errors in the sequence of actions a user performs in the software.

**🎯 Compatibility Bugs:** These arise when the software does not function correctly with certain operating systems, browsers, or other software.

**Que.39) What are the different Methodologies in Agile Development Model?**

**Ans. Scrum:** A framework that uses short cycles (sprints) to manage software development, emphasizing teamwork and continuous improvement.

**🎯Kanban:** A visual workflow management system that focuses on limiting work in progress and optimizing flow to enable continuous delivery.

**🎯Extreme Programming (XP):** An agile methodology that emphasizes technical practices like pair programming, test-driven development, and continuous integration.

**🎯Feature-Driven Development (FDD):** An iterative approach where developers focus on delivering features in short cycles.

**🎯Lean Software Development:** An approach inspired by Lean Manufacturing that aims to reduce waste and deliver value quickly.

**🎯Dynamic Systems Development Method (DSDM**)**:** A structured approach to agile project management that emphasizes user involvement and frequent delivery.

**🎯Behaviour-Driven Development (BDD):** A collaborative approach that focuses on defining and testing software based on specific scenarios and user stories.

**🎯Adaptive Software Development (ASD):** A framework that emphasizes adaptability and flexibility in response to changing requirements.

**🎯Crystal:** A family of agile methodologies that adapts to different project types, team sizes, and criticality levels.

**Que.40) Explain the difference between Authorization and Authentication in Web testing. What are the common problems faced in Web testing?**

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Authentication** | **Authorization** |
| **Definition** | The process of verifying the identity of a user. | The process of verifying what actions or resources the user has permission to access. |
| **Question it answers** | *"*Who are you?" | *"*What are you allowed to do?" |
| **Occurs When** | During login (e.g., username and password check). | After authentication, when accessing specific resources or features. |
| **Example** | Entering correct credentials to log in. | Accessing the admin panel only if you're an admin. |
| **Failure Result** | Login denied or invalid credentials message. | Access denied or 403 Forbidden error. |
| **Testing Focus** | Login functionality, session management, password security. | Role-based access control, page/feature restrictions. |

**Ans.**

**Common problems:**

**1. Cross-Browser and Device Compatibility:**

➢Web applications need to function consistently across various browsers (Chrome, Firefox, Safari, Edge) and devices (desktops, tablets, mobile phones).

➢This requires careful testing to ensure layout, functionality, and responsiveness are maintained across different platforms.

**2. Dynamic Content:**

➢Web applications often utilize dynamic content, which can be difficult to test due to its changing nature and reliance on backend services.

➢Testers need to ensure that dynamic elements render correctly and that data is displayed accurately.

**3. Performance and Load Testing:**

➢Web applications must handle traffic and user load effectively without performance degradation.

➢Performance testing helps identify bottlenecks and optimize load times.

➢Load testing simulates real-world traffic to assess how the application behaves under stress.

**4. Security Vulnerabilities:**

➢Web applications are susceptible to various security threats, including cross-site scripting (XSS), SQL injection, and unauthorized access.

➢Security testing is crucial to identify and mitigate these vulnerabilities.

**5. UI/UX Issues:**

➢Ensuring a good user experience (UX) is essential for web applications.

➢This includes testing aspects like usability, accessibility, and visual design consistency.

**6. Integration Testing:**

➢Web applications often integrate with external services, databases, and other systems.

➢Integration testing verifies that these components interact correctly and exchange data seamlessly.

**7. Responsive Design:**

➢Web applications need to adapt to different screen sizes and devices.

➢Responsive design testing ensures that the layout and functionality are optimized for various devices.

**8. Data Privacy:**

➢Web applications handling user data must comply with privacy regulations.

➢Testing ensures that data is stored, processed, and transmitted securely, and that users' privacy is protected.

**Que.41) To create HLR & TestCase of WebBased(Whatsapp Web)**

1. **WhatsApp Web :** [**https://web.whatsapp.com/**](https://web.whatsapp.com/)

**Write a positive and negative scenario of Whatsapp front & back camera, chat, Audio video call, Status, Payment.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(File Name: Whatsapp HLR and Test Case).

**Que.42) Write a scenario of Pen.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.43) Write a scenario of Pen Stand.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.44) Write a scenario of Door.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.45) Write a scenario of ATM.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.46) Write a scenario of Microwave owen.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.47) Write a scenario of Coffee Vending Machine.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.48) Write a scenario of Chair.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.49) Create Test Case on compose Mail Functionality.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name:Compose Mail [Assignment]).

**Que.50) Online Shopping to buy product[Flipkart] Create HLR and Test.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name:Flipkart HLR and Test case [Assignment]).

**Que.51) Write a scenario of Wrist Watch.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.52) Write a scenario of Lift(Elevator).**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Positive and Negative All Test scenario).

**Que.53) Write a scenario of Whatsapp Payment.**

**Ans.** The answer is attached to the **Excel sheet** created with the assignment, which you can see(file name: Whats app web based test scenario).