❖ *Assignment*

*1. What is Integration testing. ?

Integration Testing is a level of software testing where individual units or components of a software application are combined and tested as a group. The main purpose is to check data flow, interaction, and communication between integrated modules.

- Integration testing in test to one to more functionality test.
- Integration testing test after unit testing.

Advantages

- Verifies interactions between modules.
- Detects errors early in the interface.
- Improves reliability of the system.

*2. What is Alpha testing. ?

Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or the public. It is conducted at the developer's site, typically by internal staff or a dedicated testing team.

- **Alpha Testing Process:**

1. Developers complete most features.
2. QA team performs rigorous testing.
3. Testers report bugs to developers.
4. Developers fix issues.
5. Testing is repeated until the software is stable.

*3. What is beta testing. ?

Beta Testing is a type of software testing performed after alpha testing and before the final release. It is conducted by real users in a real environment outside the organization that developed the software.

*4. What is functional system testing. ?

Functional System Testing is a type of software testing where the complete system is tested to ensure that it behaves according to the specified functional requirements.

- It focuses on verifying the functionality of the system as a whole.
- It is performed after integration testing and before user acceptance testing (UAT).
- It is usually black-box testing, meaning the tester doesn't need to know the internal code structure.

- Login/logout functionality.
- User registration.
- File upload/download.
- Search and filtering.
- Payment processing.

*5. What is Non-Functional Testing. ?

Non-Functional Testing is a type of software testing that evaluates the non-functional aspects of a software application. It checks how well the system performs rather than what the system does.

- Not test to functionality

- Focuses on quality attributes
- Does not test specific behaviors or functions
- Often conducted after functional testing

- **Type**

  Performance Testing

  Load Testing

  Stress Testing

  Usability Testing

  Security Testing

*6. What is GUI Testing. ?

GUI Testing Graphical User Interface Testing is a type of software testing that focuses on verifying the graphical elements of an application. It ensures that the user interface behaves as expected and is visually consistent, functional, and user-friendly.

- GUI = Graphical User Interface, like buttons, menus, icons, input fields, etc.
- Ensures correct visual appearance (fonts, colors, layout).
- Validates functionality of UI elements (e.g., clicking a button performs the correct action).

What is Tested in GUI Testing?

(Visual Design, Functional Elements, Navigation,)

*7. What is Adhoc testing. ?

Adhoc Testing is a type of software testing performed without any formal test planning, documentation, or strategy. It is an informal and unstructured approach where testers aim to find defects randomly by exploring the application, relying mainly on their intuition, experience, and understanding of the application.

Software testing performed without proper planning and documentation.

Monkey Testing – Random inputs to check system crash.

*8. What is load testing. ?

Load testing is a type of **performance testing** in software engineering used to evaluate how a system behaves under expected user loads. The primary goal is to check the **system's performance, stability, and response time** when subjected to normal and peak conditions.

Load testing in tester check the capacity per mini second in website user use the website and increase capacity.

- **Objective:**
  To ensure the software application can handle the expected number of users or transactions without performance degradation.
- **Focus Areas:**

- Response time
- Throughput (requests per second)

- Typically during or after the development phase, before the software goes live.

*9. What is stress Testing. ?

Stress Testing is a type of software testing used to evaluate how a system behaves under extreme or unfavourable conditions**.** It helps determine the stability and reliability of software when it's pushed beyond normal operational limits.

To check how the system handles high load, data volume, traffic, or resource usage.

- Not just performance, but failure behaviour (e.g., does it crash, slow down, or recover gracefully?)

Types of Issues Found:

- Memory leaks
- Server crashes
- Database connection errors

- Performance bottlenecks
- Improper error messages or data loss

*10. What is white box testing and list the types of white box testing. ?

White Box Testing (also known as Clear Box Testing**,** Glass Box Testing**,** or Structural Testing) is a software testing technique in which the internal structure, design, and implementation of the software being tested is known to the tester.

White box testing in tester knowledge and full access

- **Types of White Box Testing**

  1.statement coverage

  2.branch coverage

  3.data flow testing

  4.path coverage

*11. What is black box testing? What are the different black box testing techniques. ?

Black Box Testing is a software testing method where the internal code, logic, or structure of the application is not known to the tester.

Instead, testers focus on testing the functionality of the software by providing inputs and examining the outputs, without any knowledge of how the system processes those inputs internally.

1. **Equivalence Partitioning**
2. Decision Table Testing
3. Use Case Testing

*12. When should "Regression Testing" be performed. ?

Regression Testing is performed **every time** a change is made to the codebase to ensure that **existing functionalities still work correctly** after the change. It's a critical part of maintaining software quality, especially in complex or frequently updated systems.

- **After Fixing a Bug**

- When a defect or issue is fixed, regression testing checks whether the fix has impacted other parts of the application.

- **After Code Enhancements or Feature Additions**

- When new features are added, even in different modules, they can accidentally affect existing functionality.

- **After Configuration Changes**

- System settings, environment changes, or configuration updates might affect software behavior.
- Example: Changing server configuration or database connection strings.

- Before a Release or Deployment

*13. What is 7 key principles? Explain in detail. ?

The 7 Key Principles often refers to foundational principles in software testing **.** These principles guide the planning, design, execution, and evaluation of testing efforts to ensure effective and efficient testing.

1. Testing Shows Presence of Defects

- Explanation: Testing can show that defects are present, but cannot prove that there are no defects. Even if no defects are found during testing, it doesn't guarantee the software is 100% bug-free.
- Example: A login system may pass all test cases but could still have a security loophole not covered in the tests.

2. Exhaustive Testing is Impossible

- Explanation: Testing all possible inputs and scenarios is not feasible, especially for large or complex applications. Therefore, risk-based and prioritized testing should be performed.
- Example: You can't test every value between 1 and 1,000,000. Instead, you test representative values (like 1, 100,000, and 1,000,000).

3. Early Testing

- Explanation: Testing activities should begin as early as possible in the software development life cycle (SDLC) to detect defects early, which helps reduce costs and effort.
- Example: Reviewing requirement documents for inconsistencies or ambiguities before coding starts.

4. Defect Clustering

- Explanation: A small number of modules usually contain most of the defects. This follows the Pareto principle (80/20 rule) — 80% of problems are often found in 20% of the code.
- Example: A payment gateway module may be more error-prone than a static homepage and thus needs more attention during testing.

5. Pesticide Paradox

- Explanation: If the same set of test cases is repeated, they will eventually stop finding new bugs. Test cases need to be regularly reviewed and updated.
- Example: Running the same test script for months might not catch new bugs introduced by recent changes.

6. Testing is Context Dependent

- Explanation: Testing approach and strategy should be adapted based on the context of the project — like the type of software (e.g., web app, embedded system), its criticality, and usage.

- Example: A safety-critical system (like for medical or aviation) needs rigorous testing, whereas a prototype app may be tested less formally.

7. Absence-of-Errors Fallacy

- Explanation: Just because the software has no known defects, it doesn’t mean it's useful or meets user needs. The software must also solve the intended problem correctly.
- Example: A calculator app that works perfectly but doesn't support basic functions users expect is technically defect-free but still useless.

*14. Difference between Smoke and Sanity. ?

Smoke testing	Sanity testing
1. To verify whether the basic and critical functionalities of a build are working.	1. To verify specific functionality or bug fixes after minor changes.
2.After a new build is received to check if it's stable enough for further testing.	2. After receiving a modified build, often after a bug fix or minor update.
3.Shallow and wide; covers major functionalities without going into details.	3. Narrow and deep; focuses on specific areas or functionalities.
4.Often automated; focuses on build verification.	4. Usually manual; confirms that recent changes work correctly.
5.Checking if the application launches, login works, and main menu opens.	5. Verifying if the "Reset Password" function works after being fixed.
6.Build Verification Testing	6. Subset of Regression Testing

*15. Difference between verification and Validation. ?

Verification	validation
1. The process of checking if the product is being built correctly.	1. The process of checking if the right product is being built.
2. To ensure the software meets design specifications.	2. To ensure the software meets user requirements.
3. Focuses on internal quality and correctness.	3. Focuses on external behaviour and usefulness.
4. Reviews, inspections, walkthroughs, static analysis	4.Testing, user acceptance testing, beta testing
5. Static process (no code execution involved)	5. Dynamic process (code is executed)
6. During the development phase	6. After development, during or after testing phase
7. Done by developers, QA team	7. Done by testers, end users

*16. Explain types of Performance testing. ?

Performance testing is a type of non-functional testing that determines how an application behaves under various conditions, such as load, stress, and scalability. The goal is to ensure the software performs well in terms of speed, responsiveness, stability, and scalability.

1. Load Testing

- Purpose: To check how the application performs under expected user load.
- Identify performance bottlenecks before the system goes live.

- Example: If a website is expected to handle 1,000 users simultaneously, load testing checks how it behaves with 1,000 users.

2. Stress Testing

- Purpose: To determine the application's robustness and behavior under extreme or beyond-expected loads.
- Identify the breaking point of the system and how it recovers.
- Example: Gradually increasing the load beyond normal levels (e.g., 10,000 users) to see when the system crashes.

3. Spike Testing

- Purpose: To evaluate the system’s response to sudden large increases or decreases in load.
- Check how well the system can handle sudden spikes in traffic.
- Example: Simulating a flash sale on an e-commerce site where user traffic jumps rapidly.

4. Endurance Testing

- Purpose: To determine if the system can handle a moderate workload over an extended period.
- Identify memory leaks, performance degradation, and long-term stability.
- Example: Running a system with 500 users for 24 hours to ensure it doesn't slow down or crash over time.

5. Scalability Testing

- Purpose: To check if the system can scale up or down efficiently when workload increases or decreases.
- Evaluate performance with increasing user loads and infrastructure changes.
- Example: Adding more users and servers to see if the response time remains consistent.

6. Volume Testing

- Purpose: To test the system's ability to handle large volumes of data.
- Check database performance and data processing under high data volumes.
- Example: Uploading millions of records to the database to evaluate its respon

*17. What is Error, Defect, Bug and failure. ?

1. Error

- **Definition**: A mistake made by a developer or human during software development.
- **Also known as**: Human mistake or slip.
- **Example**: A programmer writes `a = b + c` instead of `a = b - c`.

2. Defect

- **Definition**: A flaw or imperfection in the software code or design that is introduced during development.
- **Detected in**: Development or testing phase.
- **Example**: Due to the above error, the application calculates a wrong value.
- **Note**: When a defect is found by a tester, it's often referred to as a bug.

3. Bug

- **Definition**: A defect found during testing or after release.
- **Also known as**: Fault or issue.
- **Example**: A tester finds that a "Login" button doesn't work even when correct credentials are entered.

4. Failure

- **Definition**: When the software does not behave as expected in the actual environment (during execution).
- **Caused by**: One or more defects/bugs going undetected.
- **Example**: A banking app crashes during a money transfer, causing a loss in transaction.

*18. Difference between Priority and Severity. ?

Priority	severity
1. Indicates how soon the defect should be fixed.	1. Indicates the impact of the defect on the system.
2. Business or customer needs.	2. Functionality and system behaviour.
3. Project Manager / Business Analyst / Product Owner	3. Testers / QA Team
4. Urgency of the fix.	4. Seriousness of the defect.
5. Development schedule.	5. Testing and defect triage process.
6. A typo in the company name on the homepage.	6. Application crashes on login.
7. High, Medium, Low	7. Critical, Major, Minor, Trivial

*19. What is Bug Life Cycle. ?

Stage	Description
1. New	A bug is identified and logged for the first time.
2. Assigned	The bug is assigned to a developer or team for

Stage	Description
	further analysis.
3. Open	The developer begins analysis and work on the bug.
4. In Progress / Fixed	The developer fixes the bug and marks it as resolved.
5. Ready for Retest	The build with the fix is given to testers to verify the fix.
6. Retesting	Testers retest the application to check if the bug is really fixed.
7. Reopened	If the bug still exists, it's moved back to "Open" or "Assigned".
8. Verified	If the bug is fixed successfully, it is marked as "Verified".
9. Closed	After verification, the bug is marked as "Closed".
10. Rejected / Not a Bug / Deferred / Duplicate	Optional states if the bug is invalid, duplicate, or postponed.

***20. Explain the difference between Functional testing and NonFunctional testing. ?**

Functional testing	NonFunctional testing
1. Verifies what the system does (its functionality).	1. Verifies how well the system performs under certain conditions.
2.Business requirements and features.	2.Performance, usability, reliability, etc.
3.To ensure the system behaves as expected.	3.To ensure the system meets quality standards.
4.Login, user registration, search functionality.	4.Load testing, stress testing, usability testing.
5.Validates actions and operations of the software.	5.Validates performance and behavior under constraints.

6.Based on functional specifications or user requirements.	6.Based on performance benchmarks or system specifications.
7.Selenium, QTP, Test Complete, etc.	7.JMeter, LoadRunner, BlazeMeter, etc.

*21. To create HLR & TestCase of 1)(Instagram , Facebook) first page and chat functionality. ?

HLR= high level requirements

First Page (Login/Sign-Up) – Instagram & Facebook

Hlr >

1. The system shall allow users to log in with a valid username/email and password.
2. The system shall allow new users to sign up with required personal details.
3. The system shall validate input fields (e.g., empty, invalid email format).
4. The system shall provide a “Forgot Password” feature for recovery.
5. The system shall display appropriate error messages for invalid actions.

- **Chat Functionality – Instagram & Facebook**

Hlr >

1. The system shall allow users to open a chat with friends/connections.
2. The system shall allow users to send and receive real-time text messages.
3. The system shall support media sharing (images, videos, etc.) in chat.
4. The system shall show read/delivered message status.
5. The system shall allow message deletion for sender/receiver.

- **Test case**

First Page (Login/Sign-Up)

	Description	**steps**	**Expected Result**
1.	Verify login with valid credentials	1. Enter valid email 2. Enter valid password 3. Click login	User is redirected to home/feed page
2.	Verify login with invalid credentials	Enter invalid email/password	Error message shown
3.	Verify "Forgot Password" flow	Click on "Forgot Password" and enter email	Reset link is sent to the email
4.	Verify mandatory fields in sign-up	Leave fields blank and submit	Show validation messages
5.	Verify successful sign-up	Fill all valid details and submit	Account is created, user redirected to feed

- **Chat Functionality**

	Description	**steps**	**Expected Result**
1.	Open a chat with a friend	Click on chat icon, select user	Chat window opens
2.	Send a text message	Type message and press send	Message is sent and appears in chat
3.	Receive message in real-time	Wait for another user to message	Message appears instantly
4.	Send image in chat	Click attach > select image > send	Image is delivered and viewable

5.	Delete message	Long press message > select delete	Message is removed as per selected option
6.	View read/delivered status	Send message and wait for recipient	Status changes (Sent > Delivered > Seen)

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

SDLC	STLC
1.Process used to design, develop, and deliver software	1.Process used to plan, design, execute, and evaluate software tests
2.Entire software development process (requirements to deployment)	2.Software testing process (test planning to test closure)
3.To build a complete software product	3.To ensure the software meets quality standards and is bug-free
4.Business analysts, developers, testers, project managers, etc.	4.Testers and quality assurance (QA) teams
5.Starts with requirement gathering	5.Starts after the requirement phase of SDLC
6.Requirement analysis, design, coding, testing, deployment, maintenance	6.Requirement analysis, test planning, test design, test execution, defect reporting, test closure
7.A working software product	7.A verified and validated software product

*23. What is the difference between test scenarios, test cases, and test script. ?

Test Scenario	Test case	Test Script
1.A high-level description of what to test	1.A detailed document with steps to test a specific functionality	1.A set of instructions (manual or automated) to execute the test case
2.Identifies what to test	2.Specifies how to test	2.Automates or guides the execution of a test
3.Broad and conceptual	3.Detailed and specific	3.Very detailed and executable
4.Verify login functionality	4.Enter valid username and password Click Login Verify dashboard	4.An automated script in Selenium that performs the login steps
5.Simple sentence or bullet point	5.Structured format with preconditions, steps, expected result	5.Code/script or step-by-step manual instruction
6.Test Analyst / QA	6.Test Engineer / QA	6.Automation Tester or Manual Tester
7.Not tool-specific	7.Managed in test management tools (postman, JIRA)	7.Written in tools like Selenium, QTP, or in test management tools

*24. Explain what Test Plan is? What is the information that should be covered. ?

A Test Plan is a detailed document that outlines the strategy, objectives, schedule, estimation, deliverables, and resources required to perform testing for a software project. It serves as a blueprint to ensure that testing is carried out in a systematic, organized, and traceable manner.

- **Covered in a Test Plan**

- A unique identifier for the test plan (e.g., TP_001).
- Overview of the project and purpose of the test plan.
- List of software components/modules to be tested.
- Specific functionalities to be verified (e.g., login, payment).
- Out-of-scope features for this test cycle.
- High-level testing approach (manual, automation, levels of testing).
- Conditions that must be met to start/end testing.
- Documents and outputs to be delivered (test cases, reports).
- Detailed list of activities involved in the testing process.
- Hardware/software requirements for testing (os, tools).
- Allocation of tasks among team members.
- Timeline of testing activities.
- Potential issues that may affect testing and how to handle them.
- Stakeholders who approve the test plan.

*25. <u>What is priority. ?</u>

- In software testing, priority refers to the order in which a defect or test task should be fixed or executed based on its business or project importance.
- Priority indicates how soon a defect or test case should be addressed.

  Exa: A spelling mistake on the homepage title may have high priority (because it's visible to all users) but low severity (it doesn't affect functionality).

*26. What is severity. ?

Severity indicates how serious the defect is in terms of affecting the system's functionality, regardless of how often it happens or how visible it is to users.

System crash, data loss, or complete failure of a core feature.

Application crashes on login.

*27. Bug categories are…?

- Functional Bugs**,** Performance, Bugs Security Bugs
- Usability Bugs, Compatibility Bugs, Interface Bugs
- Logic Bugs, Database Bugs, Installation Bugs
- Boundary Bugs, Localization Bugs, Regression Bugs
- Configuration Bugs.

*28. Advantage of Bugzila .

Bugzilla is a popular open-source bug tracking system used by many software development teams. Here are the key advantages of Bugzilla:

Advantage:

1. Free & Open Source:
2. Web based interface:
3. Powerful Search and Reporting:
4. Email Notifications:
5. Security and Access Control:

6. Custom Fields and Workflow:
7. Integration Capabilities:

*29. Difference between priority and severity. ?

severity	priority
It defines the impact of the defect on the application.	It defines the urgency of fixing the defect.
Related to the technical impact of the bug.	Related to business needs and timelines.
Usually set by Testers.	Usually set by Project Managers or Clients.
A crash in the payment module – High severity.	A spelling mistake in the company name on the homepage – High priority.
May not always be fixed immediately.	Low, Medium, High,

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

Authentication	Authorization
Verifies who the user is	Determines what the user is allowed to do
To confirm identity (e.g., username/password)	To check access permissions (e.g., user roles)

Before authorization	After authentication
Login with username and password	Accessing admin panel after login
Credentials and identity	Permissions and access control
User can't log in	User sees "Access Denied" or limited functionality

- **common problems faced in Web testing**

1. Cross-Browser Compatibility Issues
2. Broken Links
3. Performance Bottlenecks
4. Security Vulnerabilities
5. Localization and Globalization Issues
6. Validation Error

*31. To create HLR & TestCase of Web Based WhatsApp web. ?

- **Hlr**
  1. The system shall allow users to scan a QR code to log in via a linked mobile device.
  2. The system shall display a list of recent chats and contacts after login.
  3. The system shall allow sending and receiving text messages in real time.
  4. The system shall allow sending and receiving media files (images, videos, documents, etc.).
  5. The system shall show message status indicators (sent, delivered, read).
  6. The system shall support group chat functionality.
  7. The system shall allow users to search for chats and contacts.
  8. The system shall allow users to view and update their profile.

- **Test Cases for WhatsApp Web**

- **Login Functionality**

Test Case ID	Description	Test Steps	Expected Result
TC-01	Verify QR Code Login	Navigate to WhatsApp Web, scan QR code using mobile.	User is logged in successfully.
TC-02	Verify incorrect QR code	Scan expired or invalid QR code.	System shows an error or asks to rescan.

- **Message Functionality**

TC-03	Send a text message	Select contact → Type message → Press enter/send.	Message appears in chat window and is sent.
TC-04	Receive a message	Another user sends message to logged-in user.	Message appears in chat window.
TC-05	Verify message status indicators	Send message to another user.	Shows single tick, double tick, and blue ticks.
Media Sharing TC-06	Send image	Click attachment → Select image → Send.	Image is sent successfully.
TC-07	Download received media	Click on received image/video → Click download icon.	Media is downloaded to local system.
Group Chat TC-08	Send message in group chat	Open group chat → Send message.	All members see the message.
TC-09	Add participant to group	Open group info → Add participant.	Participant is added to the group.
Search and Logout TC-10	Search chat/contact	Type keyword in search bar.	Relevant chats/contacts are

TC-11	Logout	Click menu → Select Log out.	shown. User is logged out and redirected to QR screen.

*32. Write a scenario of only Whatsapp chat messages. ?

Rahul: Hey! Are we still on for the movie tonight?
Priya: Hey! Yes, definitely. What time are we meeting?
Rahul: The show is at 7:30 PM. Let's meet by 7 at the mall entrance?
Priya: Sounds good
Rahul: Should I book the tickets now?
Priya: Yes please, and send me the QR code.
Rahul: Done! Just sent it. Check your messages.
Priya: Got it. Thanks!
Rahul: Want to grab something to eat after the movie?
Priya: Sure! I'm craving pizza
Rahul: Perfect, there's a Domino's nearby.
Priya: Great, see you at 7!
Rahul: See you! Don't be late
Priya: Haha, I'll try my best

*33. Write a Scenario of Pen. ?

1. The pen must have sufficient ink to write.
2. The ballpoint/gel/ink flow mechanism must be functional.
3. The user must have the pen physically available.
4. The user must have a writable surface (e.g., notebook, paper).
5. The pen should be intact (not broken or damaged).

6. The pen should be uncapped or retracted (for click pens).
7. The user knows how to use the pen.
8. The lighting and a stable surface to write on should be available.

*34. Write a Scenario of Pen Stand. ?

1. The user begins their workday and sits at their desk.

2.They notice scattered pens, pencils, and other stationery items across the desk.

3.The user picks up the pen stand and places it in a convenient location on the desk.

4.They sort through the scattered stationery and place each item into a separate compartment of the pen stand:

5.The desk now appears neat and organized.

6.Throughout the day, the user easily picks up and returns stationery from the pen stand.

7.The pen stand helps save time and reduces desk clutter.

*35. Write a Scenario of Door. ?

1. The user approaches the door.
2. The user checks if the door is locked.
3. If the door is locked, the user unlocks it using a key.
4. The user turns the handle and pushes/pulls the door.
5. The door opens successfully.
6. The user passes through the doorway.
7. The user closes the door behind them.
8. Optionally, the user locks the door again.

*36. Write a Scenario of ATM.?

- Customer inserts ATM card into the ATM machine.
- ATM prompts the customer to enter the PIN.

- Customer enters the correct PIN.
- ATM displays transaction options (Withdraw, Balance Inquiry, Deposit, etc.).
- Customer selects "Withdraw Cash".
- ATM asks for the amount to withdraw.
- Customer enters the desired amount.
- ATM verifies the amount and account balance.
- ATM dispenses the cash.
- ATM prints a transaction receipt (optional).

- Customer takes the cash and receipt.

- ATM returns the ATM card.

- Customer leaves.

*37. <u>When to used Usability Testing. ?</u>

1. During Early Design Stages
2. Before Product Launch
3. When Adding New Features
4. When There Are User Complaints
5. When Redesigning the Interface
6. To Compare Multiple Designs
7. For Continuous Improvement

*38. What is the procedure for GUI Testing. ?

1. Understand the GUI Requirements

- Review UI/UX design documents, wireframes, mockups.
- Understand functionality, layout, colors, fonts, icons, alignments, and workflows.
- Confirm platform (web, desktop, mobile) and browsers/devices supported.

2. Prepare GUI Test Plan

- Define scope, test objectives, and approach.
- Identify tools (e.g., Selenium, Test Complete, Ranorex).
- Define responsibilities, schedule, risks, and deliverables.

3. Design GUI Test Cases

Include test cases for:

- Visual Elements: Buttons, input fields, labels, icons, fonts.
- Layout: Alignment, spacing, responsiveness.
- Functionality: Button clicks, dropdowns, scroll bars.
- Navigation: Menu behavior, tab flow, hyperlinks.
- Error Messages: Pop-ups, alerts, validations.
- Cross-browser & Cross-platform compatibility.

Example:

Test Case ID	Description	Expected Result
GUI_TC_001	Check "Login" button alignment	Aligned to center and clickable

4. Set Up the Test Environment

- Install necessary software/tools.
- Configure devices, browsers, or emulators.
- Deploy the application version to be tested.

5. Execute GUI Test Cases

- Perform manual or automated testing as per the plan.
- Verify each UI component's look and behavior.
- Record actual vs. expected results.

6. Log Defects

- Report GUI issues: misalignment, incorrect color, broken links, overlapping elements.
- Use bug tracking tools like JIRA.
- Include screenshots or videos for clarity.

*39. Write a scenario of Microwave Owen.?

- User places the leftover food into a microwave-safe bowl.
- User opens the microwave oven door.
- User places the bowl inside the microwave on the rotating plate.
- User closes the microwave door securely.
- User sets the timer to 2 minutes using the keypad.
- User selects the "Start" button.
- The microwave starts heating the food with a rotating motion.
- After 2 minutes, the microwave beeps and stops automatically.
- User opens the door carefully.
- User checks if the food is properly heated.
- If more heating is needed, user repeats the process for another 30 seconds.
- Once done, user removes the bowl and closes the microwave door.

*40. Write a scenario of Coffee vending Machine. ?

- User approaches the coffee vending machine.
- Machine displays a welcome message and list of available beverages (e.g.,

  Espresso, Cappuccino, Black Coffee, Milk Tea).

- User inserts coins/card/tap UPI to add balance (if required).
- Machine shows balance and prompts to select a beverage.
- User selects Cappuccino with sugar**.**
- Machine confirms selection and asks for additional options (e.g., "Sugar level:

  Low, Medium, High").

- User selects Medium sugar level**.**
- Machine starts dispensing**:**
- Machine displays Please wait while preparing.
- After a few seconds, machine dispenses the coffee**.**
- Machine displays Thank you! Please collect your drink**.**

*41. Write a scenario of chair. ?

- The employee enters the office and approaches their workstation.
- The employee pulls the chair out from under the desk.
- The employee adjusts the height of the chair for comfortable seating.
- The employee sits down on the chair.
- They roll slightly forward to align themselves with the desk.
- The backrest supports the employee as they lean back for comfort.
- Armrests are used while typing or using the mouse.
- During a break, the employee swivels the chair to turn toward a colleague for a quick discussion.
- The employee stands up, and the chair returns to its original position.

*42. Create Test Cases on Compose Mail Functionality. ?

Test Case 1: Verify UI Elements on Compose Mail Window

- **Test Case ID:** TC_CM_001
- **Precondition:** User is logged in and Compose button is visible
- **Steps:**

1. Click on "Compose" button

- **Expected Result:**
  - New mail window opens
  - Fields visible: *To*, *CC*, *BCC*, *Subject*, *Body*, *Send button*, *Attachment option*

Test Case 2: Send Mail with Valid Inputs

- **Test Case ID**: TC_CM_002
- **Precondition**: Valid email addresses
- **Steps**:
  1. Enter valid email in "To" field
  2. Enter subject and body text
  3. Click "Send"
- **Expected Result**:
  - Email is sent successfully
  - Confirmation message displayed

Test Case 3: Send Mail Without Subject

- **Test Case ID**: TC_CM_003
- **Precondition**: Valid email and body, empty subject
- **Steps**:
  1. Leave subject blank
  2. Fill "To" and body
  3. Click "Send"
- **Expected Result**:
  - System shows a warning: “Send without subject?”
  - Allows user to continue or go back

Test Case 4: Send Mail Without Body

- **Test Case ID**: TC_CM_004
- **Steps**:
  1. Fill "To" and subject
  2. Leave body empty
  3. Click "Send"
- **Expected Result**:
  - Email is sent
  - Optional warning or silent send

Test Case 5: Invalid Email Format

- **Test Case ID**: TC_CM_005
- **Steps**:
    1. Enter invalid email like user@com or user@.com
    2. Click "Send"
- **Expected Result**:
    - Error message displayed
    - Prevents sending

Test Case 6: Add CC and BCC

- **Test Case ID**: TC_CM_006
- **Steps**:
    1. Add valid emails in *CC* and *BCC*
    2. Fill subject and body
    3. Click "Send"
- **Expected Result**:
    - All recipients receive the mail

Test Case 7: Attach a File

- **Test Case ID**: TC_CM_007
- **Steps**:
    1. Click attachment icon
    2. Choose a file
    3. Fill "To", subject, body and click "Send"
- **Expected Result**:
    - Mail with attachment sent successfully

Test Case 8: Send Button Disabled with No Recipient

- **Test Case ID**: TC_CM_008
- **Steps**:
    1. Leave "To", "CC", and "BCC" fields empty
    2. Click "Send"
- **Expected Result**:
    - Send button is disabled or shows error

Test Case 9: Draft Auto-Save Functionality

- **Test Case ID**: TC_CM_009
- **Steps**:
    1. Type subject and body without sending
    2. Close compose window
- **Expected Result**:
    - Draft is auto-saved and accessible in “Drafts” folder

*43. Write a Scenario of Wrist Watch. ?

1.User is walking in the street and wants to know the time.

2.User raises their wrist and looks at the watch dial.

3.User reads the time from the watch (e.g., 3:45 PM).

4.User checks if they are on time for their appointment.

5.User puts the wrist back down and continues walking.

*44. Write a Scenario of Lift(Elevator). ?

1. Passenger enters the building and walks toward the elevator.
2. Passenger presses the Up call button on the elevator panel on the ground floor.
3. The elevator receives the signal and moves to the ground floor.
4. Elevator doors open automatically when it arrives.
5. Passenger enters the elevator.
6. Elevator doors close automatically after a few seconds.
7. Passenger presses the 5 button on the floor selection panel inside the elevator.
8. The elevator processes the request and begins moving upward.
9. The elevator stops at any requested floors along the way.
10. The elevator reaches the 5th floor.
11. Elevator doors open automatically.
12. Passenger exits the elevator.

*45. Write a Scenario of Whatsapp payment. ?

User opens WhatsApp on their Smartphone.

User selects a contact (who has WhatsApp Payments enabled).

User taps on the ₹ (Rupee) icon or selects the Payment option from the attachment menu .

A payment screen opens.

- User enters the amount (₹500).
- Optionally, user adds a note (Dinner bill).

User taps Send Payment**.**

WhatsApp prompts the user to enter their UPI PIN**.**

After entering the correct UPI PIN, the payment is processed.

A confirmation message appears:

- ₹500 sent to [Recipient's Name] successfully.

In the chat window, a payment confirmation message is displayed to both users.

The recipient receives a notification: