Tops Assingment Module	
Que-1	**What is Exploratory Testing?**
Ans-1	Exploratory testing is an approach to software testing that is often described as simultaneous learning, test design, and execution. It focuses on discovery and relies on the guidance of the individual tester to uncover defects that are not easily covered in the scope of other tests. Exploratory testing is all about discovery, investigation and learning/curiosity. This emphasizes on personal freedom and responsibility of the individual tester. Exploratory Testing is a type of software testing in which the tester is free to select any possible methodology to test the software. Exploratory Testing is a manual software testing type where testers jump straight to testing and make spontaneous decisions about what to test on the fly. Exploratory Testing is an approach that allows product testers to find bugs and errors without using a script or test cases. It basically encourages testers to think creatively and simulate real-world usage based on their experience by mimicking the actions of end users.
Que-2	**What is traceability matrix?**
Ans-2	A traceability matrix is a document that details the technical requirements for a given test scenario and its current state. It helps the testing team understand the level of testing that is done for a given product. The traceability process itself is used to review the test cases that were defined for any requirement Requirements Traceability Matrix is a testing artifact that keeps track of all the user requirements and the details of the test cases mapped to each of those requirements. It serves as a documented proof that all the requirements have been accounted for and validated to achieve their end purpose Traceability matrix is a table type document that is used in the development of software application to trace requirements. A Software Testing Traceability Matrix (STM) is a document that links and maps test cases to their respective requirements, RTM stands for Requirement Traceability matrix. RTM maps all the requirements with the test cases A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship. Requirement Traceability Matrix (RTM) is a document that maps and traces user requirement with test cases
Que-3	**What is Boundary value testing?**
Ans-3	Any values between 18 and 65 are valid values, while anything below the minimum or above the maximum is invalid. The developer then engages in testing to ensure that the values 17 and 66 are invalid, while 18 and 65 are valid. Boundary value analysis is a powerful testing technique that helps identify defects associated with boundary conditions. By focusing on the edges of the defined ranges of input parameters, this technique ensures efficient and accurate test coverage. Boundary conditions can include the minimum, maximum, or exact values of the input or output range, or the transitions between different states or branches of the logic. Testing edge cases and boundary conditions can help you identify and prevent potential bugs, errors, or failures in your unit
Que-4	**What is Equivalence partitioning testing?**
Ans-4	Equivalence partitioning or equivalence class partitioning is a software testing technique that divides the input data of a software unit into partitions of equivalent data from which test cases can be derived. In principle, test cases are designed to cover each partition at least once. Aim is to treat groups of inputs as equivalent and to select one representative input to test them all -> EP can be used for all Levels of Testing -> Equivalence partitioning is the process of defining the optimum number of tests by: -> Reviewing documents such as the Functional Design Specification and Detailed Design Specification, and identifying each input condition withina function, -> Selecting input data that is representative of all other data that would likely invoke the same process for that particular condition. -> If we want to test the following IF statement: “If value is between 1 and 100 (inclusive) (e.g value >=1 and value <=100) Then...”
Que-5	**What is Integration testing?**

Ans-5	Integration testing -- also known as integration and testing (I&T) -- is a type of software testing in which the different units, modules or components of a software application are tested as a combined entity. However, these modules may be coded by different programmers. -> Integration Testing - Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems ->Integration Testing is a level of the software testing process where individual units are combined and tested as a group. -> The purpose of this level of testing is to expose faults in the interaction between integrated units. Test drivers and test stubs are used to assist in Integration Testing. -> Integration testing tests integration or interfaces between components,interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems. -> Integration testing is done by a specific integration tester or test team. -> Components may be code modules, operating systems, hardware and even complete systems There are 2 levels of Integration Testing -> Component Integration Testing -> System Integration Testing The four types of integration testing Methods are: -> Big-bang integration testing. ->Incremental Integration Testing a) Top-down integration testing. Top-down integration testing is a famous approach focusing first on the main module and then on its other sub-modules and subroutines. b) Bottom-up integration testing. c) Mixed/sandwich integration testing.
Que-6	**What determines the level of risk?**
Ans-6	A risk score basically follows the following formula: RISK= IMPACT x LIKELIHOOD. Risk – ‘A factor that could result in future negative consequences; usually expressed as impact and likelihood’ The risk criteria define the level at which the risk can be considered acceptable or tolerable. During the process of making decisions, the criteria are used to determine if risks are acceptable, unacceptable, or need to be reduced to a reasonably practicable level.
Que-7	**What is Alpha testing?**
Ans-7	Alpha testing is the first end-to-end testing of a product to ensure it meets the business requirements and functions correctly. It is typically performed by internal employees and conducted in a lab/stage environment. An alpha test ensures the product really works and does everything it's supposed to do. Alpha Testing is a type of software testing performed to identify bugs before releasing the product to real users or to the public. Alpha Testing is one of the user acceptance testing. This is referred to as alpha testing only because it is done early on, near the end of the development of the software. Alpha testing is commonly performed by homestead software engineers or quality assurance staff. It is the last testing stage before the software is released into the real world.
Que-8	**What is beta testing?**

Tops Assingment Module	
Ans-8	Beta testing is the process of testing a software product or service in a real-world environment before its official release. It is an essential step in the software development lifecycle as it helps identify bugs and errors that may have been missed during the development process.It is also considered as the User Acceptance Testing (UAT) which is done at customers or users area. -> Beta testing can be considered “pre-release” testing. -> Pilot Testing is testing to product on real world as well as collect data on the use of product in the classroom. It is always performed by the customers at their own site. -> It is not performed by Independent Testing Team. -> Beta Testing is always open to the market and public. -> It is usually conducted for software product. -> It is performed in Real Time Environment. -> It is always performed outside the organization. -> It is also the form of Acceptance Testing. -> Beta Testing (field testing) is performed and carried out by users or you can say people at their own locations and site using customer data. -> It is only a kind of Black Box Testing. Beta testing is an opportunity for real users to use a product in a production environment to uncover any bugs or issues before a general release. Beta testing is the final round of testing before releasing a product to a wide audience.
Que-9	**What is component testing?**
Ans-9	Component(Unit) – A minimal software item that can be tested in isolation. It means “A unit is the smallest testable part of software.” -> Component Testing – The testing of individual software components. -> Unit Testing is a level of the software testing process where individual units/components of a software/system are tested. The purpose is to validate that each unit of the software performs as designed. Unit testing is the first level of testing and is performed prior to Integration Testing. Sometimes known as Unit Testing, Module Testing or Program Testing -> Component can be tested in isolation – stubs/drivers may be employed -> Unit testing frameworks, drivers, stubs and mock or fake objects are used to assist in unit testing. -> Functional and Non-Functional testing -> Unit tests are typically written and run by software developers to ensure that code meets its design and behaves as intended with debugging tool. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. -> A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits. -> Unit tests find problems early in the development cycle. -> Unit testing is performed by using the White Box Testing method.
Que-10	**What is functional system testing?**

Ans-10	A requirement that specifies a function that a system or system component must perform. -> A Requirement may exist as a text document and/or a model -> There is two types of Test Approach -> Requirement Based Functional Testing -> Process Based Testing Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations. Types of Functional testing are - White Box Testing (Structure Based) - Black Box Testing (Specification Based) - Unit Testing - Smoke Testing - Sanity Testing - Integration Testing - User Acceptance Testing - Regression Test - Experience Base Testing(Technique - gray box, adhoc (error guessing), exploratory) - End to End Testing
Que-11	**What is Non-Functional Testing?**
Ans-11	Non-Functional Testing: Testing the attributes of a componentor system that do not relate to functionality, e.g. reliability,efficiency, usability, interoperability, maintainability andportability -> May be performed at all Test levels (not just Non Functional SystemsTesting) -> Measuring the characteristics of the system/software that can be quantified on a varying scale- e.g. performance test scaling -> Non-functional testing includes, but is not limited to, performance testing, load testing, stress testing, usability testing, maintainability testing, reliability testing and portability testing. Types of Nonfunctional testing are : - Useability Tesing : (goal of it effectiveness of the system, efficiency, accuracy, user friendliness) - Compatibility Testing: (hardware,os,software,network,browser,devices,mobile,(version of the s/w types: backward/forward compatibility testing) - GUI Testing : (types: manual based, record & replay based, model based testing) - Security Testing - (types: vulnerseability scanning,security scanning,risk assessment, security auditing, ethical hacking,posture assessment) - (roles: hackers. crackers,ethical hacker, script kiddies or packet monkety) - security testing techniques : (tiger box,black box,gray box) - Performance Testing : (goal: speed, scalability, stability) - (type: Load Testing, Stress Testing, Endurance Tesing, Spike Tesing, Volume Testing, Scalability Testing) - Load Testing type/strategies :(Manual load testing, in house/ organization developed load testing tools, Open source load testing tools, Enterprise (record & play) testing tools) - Load Testing Tools : Loadrunner, Web Load, Review's Web Load, Studio, Rational Site load, Silk Performer) - Stress Testing type: (Application Stress Tesing, Transactional Stress, Systemic Stress Testing, Exploratory Stress Testing) - Stress Testing Tools : (Stress Tester, Neo Load, App Perfect) - Installation Testing - Penetration Testing - Migration Testing
Que-12	**What is GUI Testing?**

Tops Assingment Module	
Ans-12	it is the process of testing the system's GUI of the System under Test. GUI testing involves checking the screens with the controls like menus,buttons, icons, and all types of bars – tool bar, menu bar, dialog boxes and windows etc. GUI testing generally evaluates a design of elements such as layout, colors and also fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, links, and content. GUI testing processes may be either manual or automatic and are often performed by third-party companies, rather than developers or end users. Types of GUI Testing: MANUAL BASED TESTING RECORD AND REPLAY MODEL BASED TESTING
Que-13	**What is Adhoc testing(Error Guessing)?**
Ans-13	Adhoc testing is a type of software testing that is performed informally and randomly after the formal testing is completed to find any loophole in the system. For this reason, it is also known as Random or Monkey testing. Adhoc testing is not performed in a structured way so it is not based on any methodological approach. That's why Adhoc testing is a type of Unstructured Software Testing. Adhoc testing has – No Documentation,No Test cases,No Test Design. Types Of Adhoc Testing: 1. Buddy Testing 2. Pair Testing 3. Monkey Testing
Que-14	**What is load testing?**
Ans-14	Its a performance testing to check system behavior under load.Testing an application under heavy loads, such as testing of a web site under arange of loads to determine at what point the system's response time degrades or fails. Load testing is a kind of performance testing which determines a system's performance under real-life load conditions. This testing helps determine how the application behaves when multiple users access it simultaneously. Load Testing is done in order to check when the application fails by increasing the number of users and keeping the system resourcesas constant Strategies of Load Testing : - Manual Load Testing - In house(Organization) developed load testing tools - Open source load testing tools - Enterprise(Record and Play) load testing tools Load Testing Tools : - Loadrunner - Web Load - Astra Load Test - Review's Web Load - Studio, Rational Site Load -Silk Performer
Que-15	**What is stress Testing?**

Tops Assingment Module	
Ans-15	Stress Testing: Stress Testing is done in order to check when the application fails by reducing the system resources such as RAM, HDD etc. and keeping the number of users as constant. System is stressed beyond its specifications to check how and when it fails. Performed under heavy load like putting large number beyond storage capacity, complex database queries, continuous input to system or database load. Stress Testing is done to make sure that the system would not crash under crunch situations. Stress testing is also known as endurance testing. Stress testing is used to test the stability & reliability of the system. This test mainly determines the system on its robustness and error handling under extremely heavy load conditions. Types of Stress Testing : - Application Stress Testing: - Transactional StressTesting: - Systemic Stress Testing: - Exploratory Stress Testing: Stress TestingTools : - StressTester - Neo Load - App Perfect
Que-16	**What is white box testing and list the types of white box testing?**

Ans-16	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 is also called glass testing or open box testing. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. White Box Testing Techniques : -Test/code coverage -Statement /Segment Coverage -Decision/Branch coverage -Condition coverage -Path Coverage -Multiple Condition Coverage -Finite State Machine Coverage -Control Flow Testing -Branch Condition testing -Branch Condition Combination testing -Modified Condition Decision testing -Dataflow testing -Linear Code Sequence And Jump (LCSAJ) testing White Box Testing types : -Path Testing -Loop Testing -statement coverage -Branch Testing -Conditional Testing :imp -Unit Testing -Mutation Testing -Integration Testing -Penetration Testing -dynamic & static analysis -Testing based on Memory Perspective -Test Performance of the Program
Que-17	**What is black box testing? What are the different black box testing techniques?**

Tops Assingment Module	
Ans-17	Techniques of Black Box Testing: - There are four specification-based or black-box technique: - Equivalence partitioning - Boundary value analysis - Decision tables - State transition testing - Use-case Testing - Other Black Box Testing ------ Syntax or Pattern Testing Type Of Black Box Testing: -Functional Testing -Non Fucntional Tesing -Regression Testing
Que-18	**Mention what are the categories of defects?**
Ans-18	type of defects :Data Quality/Database Defects, Critical Functionality Defects, Functionality Defects, Security Defects, User Interface Defects, Arithmetic Defects ,Logical Defects, Syntax Defects, Multithreading Defects, Interface Defects, Performance Defects, software error, Software Fault, Software Failure, Boundary and Range Defects, Data Validation Defects, Deployment Defects, Integration Defects, Documentation Defects, Data Defects categories of defects : Error of Commission, Errors of Omissions, Error of Clarity, Error of Speed or Capacity
Que-19	**Mention what bigbang testing is?**
Ans-19	Big bang integration testing is a testing approach where all components or modules are integrated and tested as a single unit. This is done after all modules have been completed and before any system-level testing is performed Big-bang integration testing is a type of integration testing that combines all the modules or components of a system into a single unit and tests them as a whole. It does not use any intermediate stages or stubs to simulate the behavior of missing or incomplete modules.
Que-20	**What is the purpose of exit criteria?**
Ans-20	Exit Criteria: - Successful Testing of Integrated Application. - Executed Test Cases are documented - All High prioritized bugs fixed and closed - Technical documents to be submitted followed by release Notes. Exit criteria provide a sense of closure and help evaluate the overall quality of the software. Exit criterion is used to determine whether a given test activity has been completed or NOT. Exit criteria can be defined for all of the test activities right from planning, specification and execution. Exit criterion should be part of test plan and decided in the planning stage.
Que-21	**When should "Regression Testing" be performed?**
Ans-21	After bug fixes: Once a bug has been identified and fixed, regression testing should be conducted to ensure that the fix hasn't introduced new bugs or affected the functionality of unrelated features. This is especially important for complex software systems where different components are tightly interconnected. Regression testing should be carried out: - when the system is stable and the system or the environment changes - when testing bug-fix releases as part of the maintenance phase - It should be applied at all Test Levels - It should be considered complete when agreed completion criteria for regression testing have been met - Regression test suites evolve over time and given that they are run frequently are ideal candidates for automation
Que-22	**What is 7 key principles? Explain in detail?**

Ans-22	1) Testing shows presence of Defects :- Testing can show that defects are present, but cannot prove that there are no defects. - Testing reduces the probability of undiscovered defects remaining in the software but, even if no defects are found, it is not a proof of correctness. - We test to find Faults - As we find more defects, the probability of undiscovered defects remaining in a system reduces. - However Testing cannot prove that there are no defects present 2) Exhaustive Testing is Impossible :- - Testing everything including all combinations of inputsand preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts. - For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (515) tests. - This is very unlikely that the project timescales would allow for this number of tests. - So, accessing and managing risk is one of the most important activities and reason for testing in any project. - We have learned that we cannot test everything (i.e. all combinations of inputs and pre-conditions). - That is we must Prioritise our testing effort using a Risk Based Approach. 3) Early Testing :- - Testing activities should start as early as possible in the software or system development life cycle, and should be focused on defined objectives. - Testing activities should start as early as possible in the development life cycle - These activities should be focused on defined objectives – outlined in the Test Strategy - Remember from our Definition of Testing, that Testing doesn't start once the code has been written! 4) Defect Clustering :- - A small number of modules contain most of the defects discovered during pre-release testing, or are responsible for the most operational failures. - Defects are not evenly spread in a system - They are 'clustered' - In other words, most defects found during testing are usually confined to a small number of modules - Similarly, most operational failures of a system are usually confined to a small number of modules - An important consideration in test prioritisation! 5) Pesticide Paradox :- - If the same tests are repeated over and over again, eventually the same set of test cases will no longer find any new defects. - To overcome this "pesticide paradox", the test cases need to be regularly reviewed and revised, and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects. - Testing identifies bugs, and programmers respond to fix them - As bugs are eliminated by the programmers, the software improves - As software improves the effectiveness of previous tests erodes - Therefore we must learn, create and use new tests based on new techniques to catch new bugs - N.B It's called the "pesticide paradox" after the agricultural phenomenon, where bugs such as the boll weevil build up tolerance to pesticides, leaving you with the choice of ever-more powerful pesticides followed by ever-more powerful bugs or an altogether different approach.' – Beizer 1995 6) Testing is Context Dependent :- Testing is basically context dependent. - Testing is done differently in different contexts - Different kinds of sites are tested differently. - For example - Safety – critical software is tested differently from an e-commerce site. - Whilst, Testing can be 50% of development costs, in NASA's Apollo program it was 80% testing - 3 to 10 failures per thousand lines of code (KLOC) typical forcommercial software - 1 to 3 failures per KLOC typical for industrial software - 0.01 failures per KLOC for NASA Shuttle code! - Also different industries impose different testing standards 7) Absence of Errors Fallacy :- - If the system built is unusable and does not fulfill the user's needs and expectations then finding and fixing defects does not help. - If we build a system and, in doing so, find and fix defects .... - It doesn't make it a good system - Even after defects have been resolved it may still be unusable and/or does not fulfil the users' needs and expectations

Tops Assingment Module

Que-23 Difference between QA v/s QC v/s Tester

	QA	QC	Tester
Ans-23	Subset of SDLC	Subset of QA	Subset of QC
	Procees - Oriented	Product-Oriented	Product-Oriented
	Preventive Process	Corrective Process	Focus On Actual Testing, preventive process
	Proactive Process	Reactive Process	Reactive Process
	Verifies The Quality	Validates The Quality	Validates The Quality
	Whole Project Team Involved	Testing Team Involed	Testing Team Involved
	Make sure the right things are being done	make sure the things are doing right	evaluates the result of done things
Purpose	Preventive issues through establishing quality standards	Verify that the product meets requirements	Detect and fix quality issue
Focus	Development Processes	Finished code as a Whole	Different aspects of the product: functionality, integrations, performance etc
Who	External Stakholders, business analysis, QA engineers software developers	QA engineers, software developers	Qa Engineers, software developers
When	Throughout the entire product development life cycle	Before the release	At the testing stage or along the development process
Doing What	Introducing Standars creating guidelines, improving development processes	Validating the product against requirements	Reviewing code, running tests, addressing defects
	focus on process to achieve required quality	focus on product to check for the required quality	Focus On Actual Testing of the product
	Ensure that processes and procedures are in place to achive quality	Activities to ensure the product quality	Validate the product against specification

Que-24 Difference between Smoke and Sanity?

	Smoke	Sanity
Ans-24	Smoke Testing is performed to ascertain (to test) that the critical functionalities/ all parts of the of the program /application is working fine or not.	Sanity Testing is done to check the new functionality / bugs have been fixed
	Smoke testing is also called subset of acceptance testing.	Sanity testing is also called subset of regression testing.
	Smoke testing is documented.	Sanity testing isn't documented.
	Smoke testing is performed by either developers or testers.	Sanity testing is normally performed by testers.
	Smoke testing may be stable or unstable.	Sanity testing is stable.
	Smoke testing is scripted.	Sanity testing is usually not scripted.
	Smoke testing is done to measure the stability of the system/product by performing rigorous testing.	Sanity testing is done to measure the rationality of the system/product by performing rigorous testing.
	Smoke testing is used to test all over function of the system/product.	Sanity testing is used in the case of only modified or defect functions of system/products.
	Smoke testing can be performed either manually or by using automation tools.	Sanity testing is commonly executed manually, not by using any automation approach. Without using test cases or scripts sanity testing can be carried out.
	Smoke testing is performed when new product is built.	Sanity testing is conducted after the completion of regression testing.
	It includes all the system's essential basic functionality.	It includes only those modules where change in code is made.
	Smoke Testing firstly performs on the initial build. Smoke testing is done first.	Sanity Testing is done on stable builds or for the introduced new features in the software.
	It is used to test End to End function of the application.	It is used to test only modified or defect fixed functions.
	In the smoke testing process, the software build could be stable or unstable.	During sanity testing, the software build is comparatively stable.
	For every new build release smoke testing is carried out.	Sanity testing is carried out when in-depth testing is not possible because of short time.
	Smoke testing is like General Health Check Up	Sanity Testing is like specialized health check up
	Testing is done without getting into deep but whenever needed tester has to go into deep.	Sanity testing does not need to go into deep of the application.

Que-25 Difference between verification and Validation

Tops Assingment Module		
Ans-25	**Verification**	**Validation**
	Verification refers to the set of activities that ensure software correctly implements the specific function	Validation refers to the set of activities that ensure that the software that has been built is traceable to customer requirements.
	It comes before validation.	It comes after verification.
	Verification is the static testing.	Validation is dynamic testing.
	After a valid and complete specification the verification starts.	Validation begins as soon as project starts.
	It consists of checking of documents/files ,designs, codes, and programs. and is performed by human.	It consists of execution of program and It includes testing and validating the actual product. is performed by computer.
	Verification is for prevention of errors.	Validation is for detection of errors.
	Verification is also termed as white box testing or static testing as work product goes through reviews.	Validation can be termed as black box testing or dynamic testing as work product is executed.
	It does not include the execution of the code.	It includes the execution of the code.
	Quality assurance team does verification.	Validation is executed on software code with the help of testing team/QC.
	It checks whether the software conforms to specifications or not.	It checks whether the software meets the requirements and expectations of a customer or not.
	It can find the bugs in the early stage of the development phase of the product	It can only find the bugs that could not be found by the verification process.
	The goal of verification is application and software architecture and specification.	The goal of validation is an actual product.
	Methods used in verification are reviews, walkthroughs, inspections and desk-checking.	Methods used in validation are Black Box Testing, White Box Testing and non-functional testing functional testing, system testing, integration, and User acceptance testing.
	Verification is based on the opinion of reviewer and may change from person to person.	Validation is based on the fact and is often stable.
	We check whether we are developing/building the right product or not.	We check whether the developed product is right.
	It is a process of checking the work-products (not the final product) of a development cycle to decide whether the product meets the specified requirements.	It is a process of checking the software during or at the end of the development cycle to decide whether the software follow the specified business requirements.
	In this type of testing, we can verify that the inputs follow the outputs or not.	In this type of testing, we can validate that the user accepts the product or not.
Que-26	**Difference between Priority and Severity?**	
Ans-26	**Priority**	**Severity**
	Priority is a term that defines how fast we need to fix a defect. priority is all about scheduling or urgency.	Severity is a term that denotes how severely a defect can affect the functionality of the software.what degree the system is impacted
	Priority is basically a parameter that decides the order in which we should fix the defects.	Severity is basically a parameter that denotes the total impact of a given defect on any software.
	Priority relates to the scheduling of defects to resolve them in software.	Severity relates to the standards or functionalities
	The value of priority is subjective.	The value of severity is objective.
	The value of Priority changes from time to time.	The value of Severity is objective and less likely to change
	The product manager/owner/client basically decides a defect's priority level.	The testing engineer basically decides a defect's severity level.
	There are 3 types of Priorities: High, Medium, and Low.	There are 5 types of Severities: Cosmetic, Minor, Moderate, Major, and Critical.
	Priority is driven by business value	Severity is driven by functionality
	Priority indicates how soon the bug should be fixed	Severity indicates the seriousness of the defect on the product functionality
	High priority and low severity status indicates, defect have to be fixed on immediate bases but does not affect the application	High severity and low priority status indicates defect have to be fixed but not on immediate bases
	Priority status is based on customer requirements	Severity status is based on the technical aspect of the product
	During UAT the development team fix defects based on priority	During SIT, the development team will fix defects based on the severity and then priority
	Priority affects business.	The severity affects the technical working of the system
Que-27	**Explain the difference between Functional testing and NonFunctional testing**	
	Functional Testing	**Non-funcitonal Testing**

Ans-27	It verifies the operations and actions of an application.	It verifies the behavior of an application.
	It is based on requirements of customer.	It is based on expectations of customer.
	It helps to enhance the behavior of the application.	It helps to improve the performance of the application.
	Functional testing is easy to execute manually It can be performed manually or using automated testing tools. Black box testing method.	It is hard to execute non-functional testing manually, Often requires specialized testing tools and frameworks to measure and evaluate the non-functional requirements. It is more feasible to test using automated tools. Example: Loadrunner.
	It tests what the product does.	It describes how the product does.
	Functional testing is based on the business requirement.	Non-functional testing is based on the performance requirement.
	Type: Unit Testing Smoke Testing Sanity Testing Integration Testing White box testing Black Box testing User Acceptance testing Regression Testing System Testing Localization Globalization Interoperability	Type: Scalability Testing Performance Testing Load Testing Volume Testing Stress Testing Security Testing Installation Testing Penetration Testing Compatibility Testing Migration Testing Usability Testing Portability Testing Disaster Recover Testing
	Functional testing is executed first, Done after unit testing and integration testing and before system testing.	Non functional testing should be performed after functional testing It can be done at various stages of the development lifecycle, from design to deployment and maintenance.
	Manual testing or automation tools can be used for functional testing	Using tools will be effective for this testing
	Business requirements are the inputs to functional testing	Nonfunctional testing describes how good the product works
	It is captured in use case.	It is captured as a quality attribute.
	It focuses on testing the functionality of the software or system.	It focuses on testing the non-functional aspects of the software or system.
	It involves testing the features and functionalities of the software, such as input/output, error handling, and user interface.	It involves testing the quality attributes of the software, such as response time, scalability, availability, and maintainability.
	Tests are typically conducted using test cases or scenarios that validate the functional requirements.	Tests are conducted using various techniques such as load testing, stress testing, security testing, and usability testing.
	Example 1) Authentication of user whenever he/she logs into the system. Check login functionality. A Login page must show textboxes to Enter the username and password. 2) System shutdown in case of a cyber attack. 3) A Verification email is sent to user whenever he/she registers for the first time on some software system.	Example 1)The dashboard should load in 2 seconds. Emails should be sent with a latency of no greater than 12 hours from such an activity. 2) The processing of each request should be done within 10 seconds 3) The site should load in 3 seconds when the number of simultaneous users are > 10000
	It is mandatory.	It is not mandatory.
Que-28	**Explain types of Performance testing.**	

Ans-28	Type : 1) Load testing :- The load testing is used to check the performance of an application by applying some load which is either less than or equal to the desired load is known as load testing. For example: In the below image, 1000 users are the desired load, which is given by the customer, and 3/second is the goal which we want to achieve while performing a load testing. 2) Stress testing :- The stress testing is testing, which checks the behavior of an application by applying load greater than the desired load. For example: If we took the above example and increased the desired load 1000 to 1100 users, and the goal is 4/second. While performing the stress testing in this scenario, it will pass because the load is greater (100 up) than the actual desired load. 3) Scalability testing :- Checking the performance of an application by increasing or decreasing the load in particular scales (no of a user) is known as scalability testing. Upward scalability and downward scalability testing are called scalability testing. Scalability testing is divided into two parts which are as follows: Upward scalability testing Downward scalability testing Upward scalability testing :- It is testing where we increase the number of users on a particular scale until we get a crash point. We will use upward scalability testing to find the maximum capacity of an application. Downward scalability testing :- The downward scalability testing is used when the load testing is not passed, then start decreasing the no. of users in a particular interval until the goal is achieved. So that it is easy to identify the bottleneck (bug). 4) Stability Testing :- Checking the performance of an application by applying the load for a particular duration of time is known as Stability Testing. 5) Volume testing :- Volume testing is testing, which helps us to check the behavior of an application by inserting a massive volume of the load in terms of data is known as volume testing, and here, we will concentrate on the number of data rates than the number of users. Volume is a capacity while Load is a quantity, i.e., load testing means no. of users, and volume testing means amount of data. 6) Soak testing (Endurance testing) :- In this type of testing, we will check the behavior of an application on the environment, which is unsupportive for a long duration of time is known as soak testing. itis a sort of performance testing used to assess how a system performs under prolonged stress. This entails giving your system a lot of work and watching to see if it slows down or crashes Generally, soak testing is a negative type of testing since we already know that the server or environment is not supportive. 7) Spike testing :- Spike testing is a type of performance testing in which an application receives a sudden and extreme increase or decrease in load. The goal of spike testing is to determine the behavior of a software application when it receives extreme variations in traffic.
Que-29	**What is Error, Defect, Bug and failure?**
Ans-29	We can say that a mistake made by a programmer during coding is called an error, an error found during the unit testing in the development phase is called a defect, an error found during the testing phase is called a bug and when an error is found at an end user's end is called as the failure.
Que-30	**What is Bug Life Cycle?**
Ans-30	“A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevents it from working correctly or produces an incorrect result. Bugs arise from mistakes and errors, made by people, in either a program’s source code or its design.” - The duration or time span between the first time defects is found and the time that it is closed successfully, rejected, postponed or deferred is called as ‘Defect Life Cycle’. A defect/bug life cycle is the sequence of steps a bug or defect goes through from its identification to its resolution in software development. This life cycle standardizes the bug management process, ensuring teams can manage and resolve them more effectively. New, Assigned, Opened, Duplicate, Differed, Not a Bug, Rejected, Reopened, Fixed, Pending Retest, Retest, Verified, and Closed.
Que-31	**To create HLR & TestCase of 1)(Instagram , Facebook) only first page?**

Tops Assingment Module

Ans-31	Ans in another sheet of this worksheet as ans 31 Test Case for instagram first page login in another sheet and ans 31 Test Case for facebook first page login in another sheet and both HLR in another sheet as Ans 31- HLR of bank/instagram and facebook

Que-32 What is the difference between the SDLC (Software Development Life Cycle) and STLC (Software Testing Life Cycle) ?

Ans-32

SDLC	STLC
SDLC is mainly related to software development.	STLC is mainly related to software testing.
Besides development other phases like testing is also included.	It focuses only on testing the software.
SDLC involves total six phases or steps. :- Requirements Collection Feasibility Study Design Programming or Coding Testing Installation Maintenance	STLC involves only five phases or steps. :- Requirement collection or System study Test Plan Write test case Traceability Matrix Defect Tracking Test Execution Report Retrospect meeting
In SDLC, more number of members (developers) are required for the whole process.	In STLC, less number of members (testers) are needed.
n SDLC, development team makes the plans and designs based on the requirements. In SDLC the business analyst gathers the requirements and create Development Plan	In STLC, testing team(Test Lead or Test Architect) makes the plans and designs. In STLC, the QA team analyze requirement documents like functional and non-functional documents and create System Test Plan
Goal of SDLC is to complete successful development of software.	Goal of STLC is to complete successful testing of software.
It helps in developing good quality software.	It helps in making the software defects free.
SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
Post deployment support , enhancement , and update are to be included if necessary. SDLC phase also includes post-deployment supports and updates.	Regression tests are run by QA team to check deployed maintenance code and maintains test cases and automated scripts. Testers, execute regression suits, usually automation scripts to check maintenance code deployed.
Creation of reusable software systems is the end result of SDLC.	A tested software system is the end result of STLC.
In SDLC, the development team creates the high and low-level design plans	In STLC, the test analyst creates the Integration Test Plan
The real code is developed, and actual work takes place as per the design documents.	The testing team prepares the test environment and executes them

Que-33 What is the difference between test scenarios, test cases, and test script?

Ans-33

Test Scenario	Test Case	Test Script
Is any functionality that can be tested.	Is a set of actions executed to verify particular features or functionality.	Is a set of instructions to test an app automatically
Is derived from test artifacts like Business Requirement Specification (BRS) and Software Requirement Specification(SRS)	is mostly derived from test scenario	is mostrly derived from test cases.
helps test the end-to-end functionality in an agile way	helps in exhaustive testing of an app.	helps to test specific things repeatedly
is more focused on what to test.	is focused on what to test and how to test	is focused on the expected result
takes less time and fewer resources to create	requires more resources and time	require less time for testing but more resources for scripts creating and updating
includes an end-to end functionality to be tested	includes test steps,data,expected results for testing	includes different commands to develop a script
the main task is to check the full functionality of a software application	the man task is to verify compliance with the applicable standars, guidelines, and customer requirements	the main task is to verify that nothing is skipped, and the result are true as the desired tesing plan
allows quickly assessing the testing scope,	allows detecting errors and defects	allows carrying out an automatic execution of test cases

Que-34 Explain what Test Plan is? What is the information that should be covered.

	Test Planning in STLC is a phase in which a Senior QA manager determines the test plan strategy along with efforts and cost estimates for the project. This includes defining test objectives, test approach, test tools, test environment, test schedules and team responsibilities and composition. However, before the right test approach and other planning details can be defined, a larger view of the organizational and project objectives must be defined first - Moreover, the resources, test environment, test limitations and the testing schedule are also determined. - The Test Plan gets prepared and finalized in the same phase. - Activities in Requirement Phase Testing - Preparation of test plan/strategy document for various types of testing - Test tool selection - Test effort estimation - Resource planning and determining roles and responsibilities. - Training requirement - Deliverables of Requirement Phase Testing - Test plan /strategy document. - Effort estimation document.
Ans-34	A test plan is a document that consists of all future testing-related activities. It is prepared at the project level and in general, it defines work products to be tested, how they will be tested, and test type distribution among the testers. Before starting testing there will be a test manager who will be preparing a test plan. In any company whenever a new project is taken up before the tester is involved in the testing the test manager of the team would prepare a test Plan. The test plan serves as the blueprint that changes according to the progressions in the project and stays current at all times.It serves as a base for conducting testing activities and coordinating activities among a QA team. It is shared with Business Analysts, Project Managers, and anyone associated with the project. You already know that making a Test Plan is the most important task of Test Management Process. Follow the seven steps below to create a test plan as per IEEE 829 Analyze the product Design the Test Strategy Define the Test Objectives Define Test Criteria Resource Planning Plan Test Environment Schedule & Estimation Determine Test Deliverables
Que-35	**What is priority?**
Ans-35	Priority in software testing refers to the order in which defects should be addressed, fixed, and tested based on their importance and impact on the project's goals. Priority levels are typically categorized into several levels, such asImmediate, High, Medium, and Low. Priority is defined as the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved. Priority is a term that defines how fast we need to fix a defect. priority is all about scheduling or urgency.
Que-36	**What is severity?**
Ans-36	Severity is absolute and Customer-Focused. It is the extent to which the defect can affect the software. In other words it defines the impact that a given defect has on the system. Bug Severity or Defect Severity in testing is a degree of impact a bug or a Defect has on the software application under test. A higher effect of bug/defect on system functionality will lead to a higher severity level. A Quality Assurance engineer usually determines the severity level of a bug/defect. "Severity is a term that denotes how severely a defect can affect the functionality of the software. what degree the system is impacted" One can define Severity as the extent to which any given defect can affect/ impact a particular software. Severity is basically a parameter that denotes the impact of any defect and its implication on a software's functionality. In other words, Severity defines the overall impact that any defect can have on a system. A defect that has completely blocked the functionality of an application where the user or the tester cannot proceed or test anything. If the whole application's functionality is inaccessible or down because of a defect, such a defect is categorized as a critical defect
Que-37	**Bug categories are?**

Ans-37	Data quality/database defects Functionality/ critical functionality defect Security Defects (Authorise - Authontication) non functionality(UI Defects), usabiltiy,performance, compatibility error of commission, error of omission, error of clarity, error of speed or caoacity. logical,syntax,arithmetic, interface, multithreding, structural defects, process defects, human defects, system integration, unit level,
Que-38	**Advantage of Bugzila ?**
Ans-38	Bugzilla is an open-source issue/bug tracking system that allows developers effectively to keep track of outstanding problems with their product. It is written in Perl and uses MYSQL database. - Bugzilla is a defect tracking tool, however it can be used as a test management tool as such it can be easily linked with other test case management tools like Quality Center, Testlink etc. - This open bug-tracker enables users to stay connected with their clients or employees, to communicate about problems effectively throughout the datamanagement chain. - Key features of Bugzilla includes: - it is easy in usage and its user interface is understandable for people without technical knowledge; - Advanced search capabilities - Effective bug tracking, -I ntegrated Collaboration, - it is an open-source widely used bug tracker; - it easily integrates with test management instruments; - E-mail Notifications ,it integrates with an e-mailing system; - Modify/file Bugs by e-mail - Time tracking - Strong security - Customization and flexibility - Localization - it automates documentation & workflow management - Comprehensive Reporting and analysis
Que-39	**What are the different Methodologies in Agile Development Model?**
Ans-39	Kanban, Scrum, Feature-driven development (FDD), Behavior-driven development (BDD), Lean development, Adaptive software development (ASD), Crystal, Extreme programming (XP), Dynamic systems development method (DSDM), Scaled Agile Framework(SAFe)
Que-40	**Explain the difference between Authentication and Authorization in Web testing.What are the common problems faced in Web testing?**

Ans-40	**Authentication**	**Authorization**
	In the authentication process, the identity of users are checked for providing the access to the system. Authentication is the process of identifying a user to provide access to a system.	While in authorization process, a the person's or user's authorities are checked for accessing the resources. Authorization is the process of giving permission to access the resources.
	In the authentication process, users or persons/client and server are verified.	While in this process, users or persons are validated. In this, it is verified that if the user is allowed through the defined policies and rules.
	It is done before the authorization process.	While this process is done after the authentication process. It is usually done once the user is successfully authenticated.
	It needs usually the user's login details. It requires the login details of the user, such as user name & password, etc.	While it needs the user's privilege or security levels.
	Authentication determines whether the person is user or not.	While it determines What permission does the user have?
	Generally, transmit information through an ID Token. Data is provided through the Token Ids.	Generally, transmit information through an Access Token. Data is provided through the access tokens.
	The OpenID Connect (OIDC) protocol is an authentication protocol that is generally in charge of user authentication process.	The OAuth 2.0 protocol governs the overall system of user authorization process.
	The authentication credentials can be changed in part as and when required by the user.	The authorization permissions cannot be changed by user as these are granted by the owner/manager of the system and only he/she has the access to change it.
	The user authentication is visible at user end.	The user authorization is not visible at the user end.

	The user authentication is identified with username, password, face recognition, retina scan, fingerprints, etc.	The user authorization is carried out through the access rights to resources by using roles that have been pre-defined.
	Example: Employees in a company are required to authenticate through the network before accessing their company email. Entering Login details is necessary for the employees to authenticate themselves to access the organizational emails or software.	Example: After an employee successfully authenticates, the system determines what information the employees are allowed to access. After employees successfully authenticate themselves, they can access and work on certain functions only as per their roles and profiles.
	authentication factor: singel factor authentication, two factor authentication, Multi- Factor authentication Popular Authentication Techniques- Password-Based Authentication Passwordless Authentication 2FA/MFA (Two-Factor Authentication / Multi-Factor Authentication) Single sign-on (SSO) Social authentication	Popular Authorization Techniques- Role-Based Access Controls (RBAC) JSON web token (JWT) Authorization SAML Authorization OpenID Authorization OAuth 2.0 Authorization
	The common problems faced in Web testing : Cross Browser Compatibility Responsiveness Cross-Device Compatibility Integration & Automation interoperability/Compatibility testing expectaion Vs. Reality Security /Privacy Performance Testing/Scalability Application Getting Slow / Challenges Arising From Insufficient Bandwidth Usability Testing Entry and Exit Points Checking the Standards and Compliance Firewalls Accessibility Testing Project Deadline User Experience/ overcoming user issues/ User Interface skill level Requirements Web Service Requests User Input Validation/ UI testing challenges Ensuring Continuous Testing Ever-Changing Environment Short Development Cycles Team Collaboration Debugging and reporting	
Que-41	**To create HLR & TestCase of WebBased (WhatsApp web)**	

Tops Assingment Module	
Ans-41	HLR IN THE WORKSHEET AS ANS -31 & 41 Verify that the QR code for WhatsApp Web is present. Verify that users can successfully scan the QR code on WhatsApp Web. Confirm that users can link multiple devices to the same WhatsApp account. Test the synchronization of messages and settings across devices. Test the security logout feature by remotely logging out of WhatsApp Web. Test WhatsApp Web on different browsers (Chrome, Firefox, Safari, Edge). Test the stability of the connection between the mobile app and WhatsApp Web. Verify that messages are synchronized in real-time. Test sending and receiving messages through WhatsApp Web. Confirm that media files can be sent and received. Test how notifications are handled when using WhatsApp Web. Verify the behavior when the mobile device is disconnected from the internet. Confirm that emoticons and stickers are displayed correctly on WhatsApp Web. Test accessing and modifying account settings on WhatsApp Web. Confirm that changes made on WhatsApp Web are reflected on the mobile app.
Que-42	**To create HLR & TestCase of ArtofTesting?**
Ans-42	HLR IN THE WORKSHEET AS ANS -31 & 42 TestCase of ArtOfTesting: When the user clicks on the name field, the text cursor should be visible in the name field. Check whether the user can click on the name field or not. Verify that the field allow copy-paste or not Verify that the user can remove text from the name field or not Verify that the field shows error message for alphanumeric value Verify that the field shows error message for numbers value. Verify that the field shows error message for special characters value. Verify that the field show error message for numeric special characters value. Verify that the field shows error message for symbol Verify that the field shows error message for when field left blank Verify that the maximum character length of the field. Verify that the entered values should be properly visible or not. Verify that the email id is only accept as per the standar specification Verify that the email id is lenght is validate or not Verify that the first character has not space verify that Email ID must not be blank verify that email id is not valid verify that first character can not have space verify that subject accept only alphabet/charater value or not verify that subject filed length is validate or not verify that subject filed can accepet 1st character space Verify that message filed can accept limited character only as per requirement specified. Verify that message field can accept alphabet, special charater, email-id, number etc. verify that messafe field can not accept symbol
Que-43	**Write a scenario of only Whatsapp chat messages**

Ans-43	Verify that the user can send messages to any individual selected from his contact list. Verify that 'Chats' window contains all the chat list with DP and name and last message preview of the other person with whom chat was initiated. Verify that clicking a chat in the chat list opens a new window containing all the chats received and sent with the other person. Verify that users can check the message delivered and read the time for a message in the 'Message Info' section. Verify that users can send and receive the message in group chats. Verify that users can send and receive voice message in individual chat. Verify that users can send and receive video message in individual chat. Verify that users can send and receive contact and location in individual chat. Verify that message display accurate timestamps or not. Verify that users can react to individual chat messages. Verify that user can delete chat messages individualy or not Verify that user can edit chat messages which sent within 15 minutes. Confirm that read receipts (blue ticks) are displayed accurately unless stated otherwise in the privacy setting of the receiver. Verify that users can send and receive images, audio, video, and emoticons in the chat with individuals. Verify that users can send and receive chats in the secondary languages available. Verify that users can delete text, images, audio, and video messages within a chat. Verify that users can clear their complete chat history in an individual or group chat. Verify that users can archive chats in an individual or group chat. Verify that users can mark chats as favorites and access all chats marked as favorites from the 'Favorites' section. Verify that users can send emojis, avtar, and gif in the chat Verify that users can set different notification set to each chat and individual account Verify that users can check all media, links, doc direct from chat Verify that users can search media chat message from direc to search option on individual Verify that users can mute & unmute chat notification Verify that users can set Disappearing messages to individual Verify that users can set a background /wallpaper to individual chat Verify that users can set Starred messages to keep that message safe from unknowingly delete. Verify that users can encryption that group and individual chat. Verify that users can set chat lock to individual chat and group chat
Que-44	**Write a Scenario of Pen?**
Ans-44	Verify the type of pen, whether it is a ballpoint pen, ink pen, or gel pen. Verify that the user is able to write clearly over different types of papers. Check the weight of the pen. It should be as per the specifications. In case not mentioned in the specifications, the weight should not be too heavy to impact its smooth operation. Verify if the pen is with a cap or without a cap. Verify the color of the ink on the pen. Check the odor of the pen's ink on writing over a surface. Verify the surfaces over which the pen is able to write smoothly apart from paper e.g. cardboard, rubber surface, etc. Verify that the text written by the pen should have consistent ink flow without leaving any blob. Check that the pen's ink should not leak in case it is tilted upside down. Verify if the pen's ink should not leak at higher altitudes. Verify if the text written by the pen is erasable or not. Check the functioning of the pen by applying normal pressure during writing. Verify the strength of the pen's outer body. It should not be easily breakable. Verify that text written by pen should not get faded before a certain time as mentioned in the specification. Check if the text written by the pen is waterproof or not. Verify that the user is able to write normally by tilting the pen at a certain angle instead of keeping it straight while writing. Check the grip of the pen, and whether it provides adequate friction for the user to comfortably grip the pen. Verify if the pen can support multiple refills or not. In the case of an ink pen, verify that the user is able to refill the pen with all the supported ink types. For ink pens, verify that the mechanism to refill the pen is easy to operate. In the case of a ballpoint pen, verify the size of the tip. In the case of a ball and gel pen, verify that the user can change the refill of the pen easily.
Que-45	**Write a Scenario of Pen Stand?**

Tops Assingment Module	
Ans-45	Verify the material of the pen stand wood, acrylic, paper, plastics, metal, ceramic,cardboard and glass. Verify the size & diemension of the pen stand. Verify tha shape of the pen stand oval,circle, spherical, hexagon, rectangular, square , pentagon, octagon. Verify the weight of the pen stand. Verify the color combination of the pen stand. Verify the look of the pen stand it's match with the surface where it's keep. verify the botton surface of the pen stand it's not slippery. Verify the company logo, name is available or not if available then its clearly visible or not Verify that how many pen it can hold. Verify that pen stand made by hand or machine.
Que-46	**Write a Scenario of Door?**
Ans-46	Verify if the door is single door or bi-folded door. Check if the door opens inwards or outwards. Verify that the dimension of the doors are as per the specifications. Verify that the material used in the door body and its parts is as per the specifications. Verify that color of the door is as specified. Verify if the door is sliding door or rotating door. Check the position, quality and strength of hinges. Check the type of locks in the door. Check the number of locks in the door interior side or exterior side. Verify if the door is having peek-hole or not. Verify if the door is having stopper or not. Verify if the door closes automatically or not – spring mechanism. Verify if the door makes noise when opened or closed. Check the door condition when used extensively with water. Check the door condition in different climatic conditions- temperature, humidity etc. Check the amount of force- pull or push required to open or close the door.
Que-47	**Write a Scenario of ATM?**

Tops Assingment Module	
Ans-47	Verify that all the labels and controls including text boxes, buttons, images, and links are present on the screen. Check the informative text written displayed on the screen is clearly visible and legible. Verify that the size, color, and UI of the different objects are as per the specifications. Verify that the application's UI is responsive i.e. it should adjust to different screen resolutions of ATM machines. Verify the type of ATM machine, if it has a touch screen, both keypad buttons only, or both. Verify that on properly inserting a valid card different banking options appear on the screen. Check that no option to continue and enter credentials is displayed to the user when the card is inserted incorrectly. Verify that the touch of the ATM screen is smooth and operational. Verify that the user is presented with the option to choose a language for further operations. Check that the user is asked to enter a pin number before displaying any card/bank account detail. Verify that there is a limited number of attempts up to which the user is allowed to enter the pin code. Verify that if the total number of incorrect pin attempts gets surpassed then the user is not allowed to continue further. And operations like temporary blocking of the card, etc get initiated. Check that the pin is displayed in masked form when entered. Verify that the user is presented with different account type options like- saving, current, etc. Verify that the user is allowed to get account details like available balance. Check that the correct amount of money gets withdrawn as entered by the user for cash withdrawal. Verify that the user is only allowed to enter the amount in multiple denominations as per the specifications. Verify that the user is prompted to enter the amount again in case the amount entered is less than the minimum amount configured. Check that the user cannot withdraw more amount than the total available balance and a proper message should be displayed. Verify that the user is provided the option to get the transaction details in printed form. Verify that the user's session timeout is maintained. Check that the user is not allowed to exceed one transaction limit amount. Verify that the user is not allowed to exceed the one-day transaction limit amount. Verify that the user is allowed to do only one transaction per pin request. Check that in case the ATM machine runs out of money, a proper message is displayed to the user. Verify that the applicable fee gets deducted along with the withdrawn amount in case the user exceeds the limit of the number of free transactions in a month. Verify that the applicable fee gets deducted along with the withdrawn amount in case the user uses a card of a bank other than that of an ATM. Check that the user is not allowed to proceed with the expired ATM card and that a proper error message gets displayed. Verify that in case of sudden electricity loss before withdrawing cash, the transaction is marked as null and the amount is not withdrawn from the user's account.
Que-48	**When to used Usablity Testing?**

Ans-48	he short answer to is early and often. Usability testing is an iterative process from the prototyping phase to the post-launch. At the conceptual stage, before making any design decisions, test your low-effort prototypes, as it'll reveal user feedback and pain points that you can work to resolve. However, it doesn't stop there. Test at every phase of the product development cycle to learn user behavior and understand what works well, what needs improvement, and how you can fix it. Continuously testing keeps your customer at the forefront and enables you to design products with vast human insight. Aesthetics and design are important. How well a product looks usually determines how well it works. - There are many software applications / websites, which miserably fail,once launched, due to following reasons – - Where do I click next? - Which page needs to be navigated? - Which Icon or Jargon represents what? - Error messages are not consistent or effectively displayed - Session time not sufficient. - Usability Testing identifies usability errors in the system early in development cycle and can save a product from failure. Goal of Usability Testing :- - Effectiveness of thesystem - Efficiency - Accuracy - User Friendliness Pros :- - It helps uncover usability issues before the product is marketed. - It helps improve end user satisfaction - It makes your system highly effective and efficient - It helps gather true feedback from your target audience who actually use your system during usability test. You do not need to rely on “opinions” from random people.
Que-49	**What is the procedure for GUI Testing?**
Ans-49	Graphical User Interface (GUI) testing is the process of testing the system’s GUI of the System under Test. GUI testing involves checking the screens with the controls like menus, buttons, icons, and all types of bars – tool bar,menu bar, dialog boxes and windows etc. - WHAT DO YOU CHECK IN GUI TESTING? - Check all the GUI elements for size, position, width, length and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields. - Check you can execute the intended functionality of the application using the GUI - Check Error Messages are displayed correctly - Check for Clear demarcation of different sections on screen - Check Font used in application is readable - Check the alignment of the text is proper - Check the Color of the font and warning messages is aesthetically pleasing - Check that the images have good clarity - Check that the images are properly aligned - Check the positioning of GUI elements for different screen resolution. Approach of GUI Testing :- - MANUAL BASED TESTING - RECORD AND REPLAY - MODEL BASED TESTING
Que-50	**Write a scenario of Microwave Owen?**

Tops Assingment Module	
Ans-50	Verify that the dimensions of the oven are as per the specification provided. Verify that the oven's material is optimal for its use as an oven and as per the specification. Verify that the oven heats the food at the desired temperature properly. Verify that the oven heats food at the desired temperature within a specified time duration. Verify the ovens functioning with the maximum attainable temperature. Verify the ovens functioning with minimum attainable temperature. Verify that the oven's plate rotation speed is optimal and not too high to spill the food kept over it. Verify that the oven's door gets closed properly. Verify that the oven's door opens smoothly. Verify the battery requirement of the microwave oven and check that it function's smoothly at that power. Verify that the text written over the oven's body is clearly readable. Verify that the digital display is clearly visible and functions correctly. Verify that the temperature regulator is smooth to operate. Verify that the temperature regulator works correctly. Check the maximum capacity of the oven and test its functioning with that volume of food. Check the oven's functionality with different kinds of food – solid, and liquid. Check the oven's functionality with different food at different temperatures. Verify the oven's functionality with different kinds of container material. Verify that the power cord of the oven is long enough. Verify that the usage instruction or user manuals have clear instructions.
Que-51	**Write a scenario of Coffee vending Machine?**
Ans-51	Verify that the dimension of the coffee machine is as per the specification. Verify that outer body, as well as inner part's material, is as per the specification. Verify that the machine's body color as well brand is correctly visible and as per specification. Verify the input mechanism for coffee ingredients-milk, water, coffee beans/powder, etc. Verify that the quantity of hot water, milk, coffee powder per serving is correct. Verify the power/voltage requirements of the machine. Verify the effect of suddenly switching off the machine or cutting the power. The machine should stop in that situation and in power resumption, the remaining coffee should not get come out of the nozzle. Verify that coffee should not leak when not in operation. Verify the amount of coffee served in single-serving is as per specification. Verify that the digital display displays correct information. Check if the machine can be switched on and off using the power buttons. Check for the indicator lights when the machine is switched on-off. Verify that the functioning of all the buttons work properly when pressed. Verify that each button has an image/text with it, indicating the task it performs. Verify that complete quantity of coffee should get poured in a single operation, no residual coffee should be present in the nozzle. Verify the mechanism to clean the system work correctly- foamer. Verify that the coffee served has the same and correct temperature each time it is served by the machine. Verify that system should display an error when it runs out of ingredients. Verify that pressing the coffee button multiple times leads to multiple serving of coffee. Verify that there is the passage for residual/extra coffee in the machine. Verify that machine should work correctly in different climatic, moistures and temperature conditions. Verify that machine should not make too much sound when in operation. Check the amount of time the machine takes to serve a single serving of coffee. Check the performance of the machine when used continuously until the ingredients run out of the requirements. Check the functioning of the coffee machine when two/multiple buttons are pressed simultaneously. Check the functioning of coffee machine with a lesser or higher voltage than required. Check the functioning of the coffee machine if the ingredient container's capacity is exceeded.
Que-52	**Write a scenario of chair?**

Tops Assingment Module	
Ans-52	Verify that the chair is stable enough to take an average human load. Check the material used in making the chair-wood, plastic etc. Check if the chair's leg are level to the floor. Check the usability of the chair as an office chair, normal household chair. Check if there is back support in the chair. Check if there is support for hands in the chair. Check if there is support for head/neck in the chair. Verify the paint's type and color. Verify if the chair's material is brittle or not. Check if cushion is provided with chair or not. Check the condition when washed with water or effect of water on chair. Verify that the dimension of chair is as per the specifications. Verify that the weight of the chair is as per the specifications. Check the height of the chair's seat from floor.
Que-53	**To Create Scenario (Positive & Negative) 1.Gmail (Receiving mail) 2. Online shopping to buy product (Flipkart)?**
Ans-53	Gmail Receiving Mail :- Verify that a newly received email is displayed as highlighted in the Inbox section. Verify that a newly received email has correctly displayed sender email Id or name, mail subject and mail body(trimmed to a single line). Verify that on clicking the newly received email, the user is navigated to email content. Verify that the email contents are correctly displayed with the desired source formatting. Verify that any attachments are attached to the email and are downloadable. Verify that the attachments are scanned for viruses before download. Verify that all the emails marked as read are not highlighted. Verify that all the emails read as well as unread have a mail read time appended at the end on the email list displayed in the inbox section. Verify that count of unread emails is displayed alongside 'Inbox' text in the left sidebar of Gmail. Verify that unread email count increases by one on receiving a new email. Verify that unread email count decreases by one on reading an email (marking an email as read). Verify that email recipients in cc are visible to all users. Verify that email recipients in bcc are not visible to the user. Verify that all received emails get piled up in the 'Inbox' section and get deleted in cyclic fashion based on the size availability. Verify that email can be received from non-Gmail email Ids like – yahoo, Hotmail etc. Flipkart buy :- Verify that on the product page, the user can select the desired attribute of the product e.g. size, color, etc. Verify that the user can add to the cart one or more products. Verify that users can add products to the wishlist. Verify that the user can see the previously added products on the cart page, after signing in to the application. Verify that the user can successfully buy more than one products that were added to his/her cart. Verify that the user cannot add more than the available inventory of the product. Verify that the limit to the number of products a user can buy is working correctly. Also, an error message gets displayed, preventing the user from buying more than the limit. Verify that the delivery can be declined during checkout for the places where shipping is not available. Verify that the Cash on Delivery option of payment is working fine. Verify that the different prepaid methods of payments are working fine. Verify that product return functionality works correctly.
Que-54	**Write a Scenario of Wrist Watch?**

Tops Assingment Module	
Ans-54	Verify the type of watch – analog or digital. In the case of an analog watch, check the correctness time displayed by the second, minute, and hour hand of the watch. In the case of a digital watch, check the digital display for hours, minutes, and seconds is correctly displayed. Verify the material of the watch and its strap. Check if the shape of the dial is as per specification. Verify the dimension of the watch is as per the specification. Verify the weight of the watch. Check if the watch is waterproof or not. Verify that the numbers in the dial are clearly visible or not. Check if the watch is having a date and day display or not. Verify the color of the text displayed in the watch – time, day, date, and other information. Verify that clock's time can be corrected using the key in case of an analog clock and buttons in case of a digital clock. Check if the second hand of the watch makes ticking sound or not. Verify if the brand of the watch and check if its visible in the dial. Check if the clock is having stopwatch, timers, and alarm functionality or not. In the case of a digital watch, verify the format of the watch 12 hours or 24 hours. Verify if the watch comes with any guarantee or warranty. Verify if the dial has glass covering or plastic, check if the material is breakable or not. Verify if the dial's glass/plastic is resistant to minor scratches or not. Check the battery requirement of the watch.
Que-55	**Write a Scenario of Lift(Elevator)?**
Ans-55	Verify the dimensions of the lift. Verify the type of door of the lift is as per the specification. Verify the type of metal used in the lift interior and exterior. Verify the capacity of the lift in terms of the total weight. Verify the buttons in the lift to close and open the door and numbers as per the number of floors. Verify that the lift moves to the particular floor as the button of the floor is clicked. Verify that the lift stops when the up/down buttons on a particular floor are pressed. Verify if there is an emergency button to contact officials in case of any mishap. Verify the performance of the floor – the time taken to go to a floor. Verify that in case of power failure, the lift doesn't free-fall and gets halted on the particular floor. Verify lifts working in case the button to open the door is pressed before reaching the destination floor. Verify that in case the door is about to close and an object is placed between the doors if the doors sense the object and again open or not. Verify the time duration for which the door remains open by default. Verify if the lift interior is having proper air ventilation. Verify lighting in the lift. Verify that at no point the lift door should open while in motion. Verify that in case of power loss, there should be a backup mechanism to safely get into a floor or a backup power supply. Verify that in case the multiple floor number button is clicked, the lift should stop on each floor. Verify that in case of capacity limit is reached users are prompted with a warning alert- audio/visual. Verify that inside lift users are prompted with the current floor and direction information the lift is moving towards- audio/visual prompt.
Que-56	**Write a Scenario of whatsapp Group (generate group)?**

Tops Assingment Module	
Ans-56	Verify that the user can create a group by adding multiple people from his contact list. check the ability to add new members to a group, ensuring they receive invitation verify that how much maximum member can add in the group verify that minimum member can add in the group. Verify that the admin role can be assigned to a specific group member. Test the functionality of admin-specific permissions, such as adding/removing members and changing group settings. Check if admins receive notifications for important group activities, like member additions or removals. Verify that only admins can create, add/remove, and make other users admin in the group. Verify that the user can send and receive the message in group chats. Verify that the user / admin can set make/edit admin in the group. vefify that user / admin must approve new members/anyone who want to join the group when approve new members setting is on. Verify that user /admin allow to edit the name /icon/ description/ disappering message timer, and the ability to pin, keep or unkeep message permission Verify that in group other persion can invite via link of group or not Verify that group chat lock and hide or not. verify that user able to keep group photo and edit it verify that user can able to change group name or not verify that user can be able to chat with voice or not. verify that user can post media file doc. file and emojis, gif, avtar or not verify that user can put poll on some topic or not. Verify that user can mute group notification or not Check if new members have appropriate permissions, such as sending messages and sharing media. Verify that existing members receive notifications for new member additions. Verify that a user can successfully create a new group. Confirm that the group is immediately visible in the user's chat list. Confirm that all added members receive an invitation and join successfully. Verify that any member of the group can change the group name and icon. Verify that the group admin can remove a member from the group. Confirm that the removed member no longer has access to the group. verify that group member voluntarily leaving the group. Confirm that the group updates for the remaining members. Verify that the group creator can generate an invitation link. Confirm that users joining via the link become members without further approval. Verify that users can search for specific messages within the group chat. Verify that the group admin can transfer admin privileges to another group member. Confirm that the new admin has the necessary permissions. Confirm that group information and messages are synchronized across multiple devices linked to the same account.
Que-57	**Write a Scenario of Whatsapp payment?**
Ans-57	Verify that user has latest whatsapp version which support whatsapp payment. check that which mobile number used in whatsapp that is available in the mobile device or not. Verify that user used the same number for the whatsapp & in register bank & upi. verify that user has the minimum balance to add and verify the bank on whatsapp upi platform by messages becouse standard sms charges may apply for verify whatsapp number . verify that user has proper available internet connection for doing payment. verify that whom user do payment that is receiver has the supported version of the whatsapp.