Assignment-2

Q1 What is Exploratory Testing?

ANS: In Exploratory Testing we try to Explore the application in all possible ways, understanding the flow of the application, we prepare a test document and then testing the application, this approach is known as Exploratory Testing.

Q2 What is traceability matrix?

ANS: RTM stands for Requirements Traceability Matrix. RTM maps all the requirements with the test cases. By using this requirement with the test case. By using this document one can verify cases covering all functionality of the application as per the requirements of the customer.

Q3 What is Boundary value testing?

ANS: Boundary Value Analysis is methodology for designing test cases that concentrates software testing effort on cases near the limits of valid ranges. This method refines equivalence partitioning, in the analysis tester generates test cases that highlight error better than equivalence partitioning

Q4 What is Equivalence partitioning testing?

ANS: Software testing technique that divides the input data of a software unit into partitions of data from which test case can be derived. It is usually performed by the QA teams.

Q5 What is Integration testing?

ANS: The Phase in Software testing in which individual software modules are combined and testing as a group. It is usually conducted by testing teams.

Q6 What determines the level of risk?

ANS: Risk should be prioritized according to their level, which is obtained by assessing the likelihood of the event occurring and the impact of that event. Then residual level should be determined by considering the management response to the risk.

Q7 What is Alpha testing?

ANS: Alpha testing is always performed by the developer at the Software development site. Sometimes it is also performed by the Independent Testing Team.

Alpha Testing is not open to Market and public. It is conducted for software applications and projects. It is always performed in virtual Environment. It is always performed within the organization. It is the form of Acceptance testing.

Alpha testing is definitely performed and carried out at the developing organization's location with the involvement of developers.

Q8 What is beta testing?

ANS: Beta Testing is always performed by the customer at their own site. It is not performed by an independent testing team.

- Beta testing is always open to the market and public. It is usually conducted for software products. It is performed in a real time environment. It is always performed outside the organization. It is also a form of acceptance testing.
- Beta testing (Field Testing) is performed and carried out by the users, or you can say people at their own location and site using

Q9 What is component testing?

ANS: A minimal software item that can be tested in isolation. It means **“A unit is the smallest testable part of software”.**

- Component testing is the testing of individual software components. Component testing is also called unit testing.
- 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 is designed.

Q10 What is functional system testing?

ANS: Functional System Testing is 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

Functional System Testing Functionality As below:

Accuracy	Provision of right or agreed result or effect
Interoperability	Ability to interact with specified systems
Compliance	Adhere to applicable standards, conventions, regulation or laws
Auditability	Ability to provide adequate and accurate audit data.
Suitability	Presence and appropriateness of functions for specified tasks.

Q11 What is Non-Functional Testing?

ANS: Non-Functional Testing is testing the attributes of a component or system that do not relate to functionality, e.g. reliability, efficiency usability, interoperability, maintainability, and portability.

Q12 What is GUI Testing?

ANS: GUI stands for Graphical User Interface 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.

Q13 What is Adhoc testing?

ANS: Adhoc testing is an informal testing type with an aim to break the system. It does not follow any test design techniques to create test cases.

- In fact is does not create test cases altogether! This testing is primarily performed if the knowledge of testers in the system under test is very high.
- Testers randomly test the application without any test cases or any business requirement document. Adhoc testing does not follow any structured way of testing and it is randomly done on any part of application.

Q14 What is load testing?

ANS: Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range 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.

Q15 What is stress Testing?

ANS: System is stressed beyond its specification 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.

It even tests beyond the normal operating point and evaluates how the system works under those extreme conditions. Stress testing is done to make sure that the system would not crash under crunch situations. Stress testing is also known as endurance testing.

Q16 What is white box testing and list the types of white box testing?

ANS: Testing based on an analysis of the internal structure of the component of the component or system.

Structure-Based testing techniques is also known as “White Box” or “Glass Box” testing technique because here the tester requires knowledge of how the software is implemented.

Here are testing techniques as below: -

- Statement Coverage.
- Branch Coverage.
- Condition Coverage.
- Multiple Condition Coverage.
- Basis Path Testing.
- Loop Testing.

Q17 What is black box testing? What are the different black box testing techniques?

ANS: Testing, either Functional or non-functional, without reference to the internal structure of the component or system.

- Specification-based testing technique is known as ‘Black Box’ or input/output driven testing techniques because they view the software as a black-box with inputs and outputs.
- The testers have no Knowledge of how the system or component is structured inside the box. In Black Box testing the tester is concentrating on what the software does, not how it does it
- The techniques of testing without having any knowledge of the interior working of the application is black box testing. The tester is oblivious to the system architecture and does not have access to the source code.
- A tester will interact with the system’s user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.
- **Techniques of Black Box Testing: -**
    - Equivalence Partitioning
    - Boundary Value Analysis
    - Decision Tables
    - State Transition Testing
    - Use-case Testing
    - Other Black Box Testing
        - Syntax or Pattern Testing.

Q18 Mention what are the categories of defects?

ANS: Defect can be categorized into different types basing on the core issues they address.

- **Data Quality/Database Defect:** Deals with improper handling of data in the database.

- **Critical Functionality Defects:** The Occurrence of these bugs hampers the crucial functionality of the application.
- **Functionality Defects:** These defects affect the functionality of the application.
- **Security Defects:** Application security Defects generally involve improper handling of data sent from the user to the application. These defects are the most severe and given highest priority for a fix.
- **User Interface Defects:** As the name suggests, the bugs deal with problems related to UI are usually considered less severe.

Q19 Mention what big bang testing is?

ANS: Big Bang integration testing all components or modules is integrated simultaneously, after which everything is tested as a whole.

Big Bang Testing has the advantages that everything is finished before integration testing starts.

Q20 What is the purpose of exit criteria?

ANS: Exit Criteria is used to determine when testing at any stage is complete the set of generic and specific conditions, agreed upon with the stakeholders, for permitting a process to be officially completed.

Purpose of exit criteria is to define when we stop testing either at the end of all testing – i.e. Product Go Live. And End of phase of testing.

Q21 When should "Regression Testing" be performed?

ANS: Regression Testing is performed: -

- When new functionality is added to the application.
- When there is a Change Requirement.
- When the defect fixed.
- When there is a performance issue fix.
- When there is an environment change.

Q22 What are the 7 Key Principles? Explain in detail?

ANS: The seven Key Principles of Software Testing.

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.
2. **Exhaustive Testing is Impossible: -** Testing everything including all combinations of inputs and preconditions is not possible. So instead of doing the exhaustive testing we can use risk and priorities to focus testing efforts. 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 preconditions). That is, we must Priorities our testing efforts 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 be focused objective – 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.
5. **Pesticide Paradox: -** If the same test is repeated eventually the same set of test case will no longer find any new defect. To overcome the ‘Pesticide paradox’, the test cases need to be regularly reviewed and revised and new and different test need to be written to exercise different parts of software or system to potentially find more defects
6. **Testing is Context Dependent: -** Testing is done differently in different contexts. Different kinds of sites are tested differently.
7. **Absence-of-Errors Fallacy: -** If the system built is unusable and does not fulfill the users' needs and expectations then finding and fixing defects does not help. If we built 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 fulfill the user.

Q23 Difference between QA v/s QC v/s Tester?

ANS:

S.N.	Quality Assurance	Quality Control	Testing
1	Activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.	Activities which ensure the verification of developed software with respect to documented requirements.	Activities which ensure the identification of bug/error/defect/in the software.
2	Focuses on processes and procedure rather than conducting actual testing on the system.	Focuses on actual testing by executing software with intent to identify bug/defect through implementation of procedures and process.	Focuses on actual testing.
3	Process-oriented activities.	Product-oriented activities.	Product-oriented activities.

4	Preventive activities.	It is a corrective process.	It is a preventive process.
5	It is a subset of Software Test Life Cycle (STLC).	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

Q24 Difference between Smoke and Sanity?

ANS:

S.N	Smoke	Sanity
1.	Smoke Testing is performed to ascertain that the critical functionalities of the program is working fine.	Sanity Testing is done to check the new functionality/Bug have been fixed.
2	The Objective of this testing is to verify “Stability” of the system in order to with more rigorous testing.	The Objective of the testing is to verify the “Rationality” of the system in order proceed to proceed with more rigorous testing.
3	This testing is performed by the developer or testers.	Sanity testing is usually performed by testers.
4	Smoke testing is usually documented or scripted.	Sanity testing is usually performed by testers.
5	Smoke testing is usually documented or scripted.	Sanity testing exercises the entire system from end to end.
6	Smoke testing exercises only the particular component of the entire system.	Sanity testing exercises the entire system from end to end.

7	Smoke testing is like a General health Check Up.	Sanity Testing is like a specialized health check-up.

Q25 Difference between verification and Validation?

ANS:

Criteria	Verification	Validation
Definition	The process of evaluating work-product (not the actual final product) of a development phase to determine whether they meet the specified requirements for the phases.	The Process of evaluating software during or at the development process to determine whether it satisfies specified business requirements.
Objective	To ensure that the product is being built according to the requirement and design specifications. In other words, to ensure that the work product meets their specified requirements.	To ensure that the product actually meets the user's needs, and that the specifications were correct in the first place. In other words, to demonstrate the product fulfills its intended use when placed in it intended environments.
Question	Are we building the product, right?	Are we building the product, right?
Evaluation Items	Plans, Requirements, Specs, Design	The actual Product/Software.

	Specs, Code, Test Cases.	
Activates	• Review • Walkthroughs • Inspection	• Testing

Q26 Explain types of Performance testing.

ANS: There are two types of Performance testing Load testing and Stress Testing.

- Load Testing:
  - Its a performance testing to check system behavior under load. Testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades or fails
- Stress Testing:
  - System is stressed beyond its specification 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.
  -

Q27 What are Error, Defect, Bug and failure?

ANS: Error: A discrepancy between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition. This can be a misunderstanding of the internal state of the software, an oversight in terms of memory

management, confusion about the proper way to calculate a value, etc.

Defect: Commonly refers to several troubles with the software products, with its external behavior or with its internal features.

Bug: A fault in program which causes the program to perform in an unintended or unanticipated manner. See anomaly, defect, error, exception and fault. Bug is the terminology of tester.

Failure: The inability of the system or component to perform its required function within specified performance requirements. See: bug, crash, exception and fault.

Q28 Difference between Priority and Severity

ANS:

Features	Severity	Priority
Definition	Severity is a parameter to denote the impact of a particular defect on the software.	Priority is a parameter to decide the order in which defect should be fixed.
Purpose	Severity means how severe the defect is affecting the functionlity.	Priority means how fast the defect has to be fixed.
Relation	Severity is related to the quality standard.	Priority is related to scheduling to resolve the problem.
Categories	Severity is divided into 4 categories: • Critical	Priority is divided into 3 Categories: • Low

	• Major • Medium • Low	• Medium • High
Who decides defects?	The Testing engineer decides the severity level of the defect.	The Product manager decided the priorities of defects.
Value	Its Values is objective.	Its Value is subjective.
Value change	Its Values doesn't change from time to time.	Its Value changes from time to time.
Association	It is associated with functionality or standard.	It is associated with scheduling.
Indication	It indicates the seriousness of the bug in the product functionality.	It indicated how soon the bug should be fixed.
Driving factor	It is driven by functionality.	It is driven by business value.
Based On	It is based on the technical aspect of the product.	It is based on the customer's requirements.

Q29 What is Bug Life Cycle?

ANS: A computer bug is an error, flaw, mistake, failure, or fault in a computer program that prevent 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 defect is found and the time that it is closed successfully, rejected, postponed or deferred is called as "Defect Life Cycle".

Q30 Explain the difference between Functional testing and Nonfunctional testing

ANS:

Functional Testing	Non-Functional Testing
Functional Testing is performed using the Functional specification provided by the client and verifies the system against the functional requirements.	Non-Functional Testing checks the performance, reliability, scalability and other non-functional aspects of the software system.
Functional testing is executed first.	Non-Functional testing should be performed after functional testing.
Manual testing or automation tool can be used for functional testing.	Using tools will be effective for this testing.
Business requirement are the inputs to functional testing.	Performance parameters like speed, scalability are inputs to non-functional testing.
Functional testing describes what the product does.	Non-functional testing describe how good the product works
Easy to do manual testing.	Tough to do manual testing.
Types of Functional testing are: • Unit Testing • Smoke Testing • Sanity Testing • Integration Testing • White Box Testing • Black Box Testing • User Acceptance Testing • Regression Testing	Types of Non-Functional Testing are: • Performance Testing • Load Testing • Volume Testing • Stress Testing • Security Testing • Installation Testing • Penetration Testing

	• Compatibility Testing • Migration Testing

Q31 To Create HLR & TestCase of

1. **(Instagram, Facebook) only first page**
2. **Facebook Login Page:**

ANS: HLR of Instagram and Facebook: - (GitHub Link).

Assignment-of-Software-Testing/HLR OF INSTAGRAM AND FACEBOOK.xlsx at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)

Test Case of Instagram and Facebook: - (GitHub Link).

Assignment-of-Software-Testing/TEST-CASE_MANUAL_TESTING_ASSIGNMENT.xlsx at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)

Q32 What is the difference between the STLC (Software Testing Life Cycle) and SDLC (Software Development Life Cycle)?

ANS:

Aspect	SDLC	STLC
Domain	SDLC is mainly related to software development.	STLC is mainly related to software testing.
Focus	Besides development other phases like testing is also included.	It focuses only on testing the software.
Phases	SDLC involves total six phases or steps.	STLC involves only five phases or steps.

Number of Member	In SDLC, more number of members (developers) are required for the whole process.	In STLC, less number of members (testers) are needed.
Team Involved	In SDLC, development team makes the plans and designs based on the requirements.	In STLC, testing team (Test lead or Test Architect) makes the plans and designs.
Objective	Goals of SDLC is to complete successful development of software.	Goal of STLC is to complete successful testing of software.
End Result	It helps in developing good quality software.	It helps in making the software defects free.
Execution	SDLC phases are completed before the STLC phases.	STLC phases are performed after SDLC phases.
Maintenance	Post deployment support, enhancement, and update are to be included if necessary.	Regression tests are run by QA teams to check deployed maintenance code and maintains test case and automated scripts.
End Result	Creation of reusable software systems is the end result of SDLC.	A tested software system is the end result of STLC.

Q33 What is the Difference between test scenario, test cases and test script?

ANS:

Test Scenario	Test Case
A test Scenario contains high-level documentation to be tested.	Test cases contain definite test steps, data, expected result for testing all the features of an application.
It focuses on more “What to Test” than “How to Test”.	A complete emphasis on “What to Test” and “How to test”.
Test scenarios and a one-liner. So there is always the possibility of ambiguity during the testing.	Test cases have defined a step, pre-requisites, expected result, etc. Therefore, there is no ambiguity in this process.
Test scenarios are derived from test artifacts like BRS, SRS, etc.	Test Case is mostly derived from test scenarios, multiple test case can be derived from a single test scenario.
It helps in an agile way of testing the end to end functionality.	It helps in exhaustive testing of an application.
Test scenarios are high-level actions.	Test cases are low-level actions.
Comparatively less and resources are required for creating and testing using scenarios.	More resources are needed for documentation and execution of test cases.

Q34 Explain What Test Plan is? What is the information that should be covered?

ANS: A test plan is a document that outlines the objectives, scope, approach resources, and schedule for a testing project. It defines the strategy for how testing will be conducted, including what will be tested, who will perform the test environment and the criteria for success.

The purpose of a test plan is to ensure that testing is systematic, though, and aligned with the project's goals and requirements.

- **Introduction:** Provides an overview of the document and its purpose.
- **Objective:** Clearly states the goals and objective of the testing efforts.
- **Scope:** Defines the boundaries of the testing, including what will and will not be tested.
- **Approach:** Describes the overall strategy for testing, including methodologies, techniques, and tools to be used.
- **Test Items** lists the specific components or features to be tested.
- **Features to be tested:** Describe the features or functionalities to be tested, often with reference to requirements or specifications.
- **Test Environment:** specifies the hardware, software, and other resources needed for testing.

- **Test Deliverables:** lists the documents, reports, and other deliverables expected from the testing process.
- **Test Schedule:** outlines the timeline for testing activities, including milestones and deadlines.
- **Test Team:** identifies the roles and responsibilities of individuals involved in testing**.**
- **Risks and Assumptions:** identifies potential risks to the testing process and any assumptions made during planning.

Q35 What are the different Methodologies in Agile Development Model?

ANS: Individuals and interactions, over processes and tools:

Suppose the team finds any issue in software then they search for another process or tool to resolve the issue. But, in agile, it is preferable to interact with client, manager or team regarding issues and make sure that the issue gets resolved.

- **Working software, over comprehensive documentation:** Documentation is needed, but working software is much needed. Agile is not saying that documentation is not needed, but working software is much needed. EXAMPLE: you have 20-page documents, but you do not have a single prototype of the software. In such a case, the client will not be happy because in the end, the client needs a document.
- **Customer collaboration, over contract negotiation:** Contract negotiation is important as they make the budget of software, but customer collaboration is more important than over contract negotiation. EXAMPLE: if you suck with the requirements or process, then do not go for a contract which

we have negotiated. You need to interact with the customer, gather their requirements.

- **Responding to change, over following a plan:**
  In the waterfall model, everything is planned, i.e., at what time, each phase will be completed. Sometimes you need to implement the new requirements in the middle of the software, so you need to be versatile to make changes in the software

Q36 Explain the difference between Authorization and Authentication in Web Testing. What are the Common Problems Faced in Web Testing?

ANS:

Authorization	Authentication
In the authentication process, the identity of users is checked for providing the access to the system.	While in authorization process, a person's or user's authorities are checked for accessing the resources.
In the authentication process, users or persons are verified.	While in this process, users or persons are validated.
It is done before the authorization process.	While this process is done after the authentication process.
It need usually the user's login details.	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.	Generally, transmit information through an Access Token.

Popular Authentication Techniques- • Password-Based Authentication. • Password less Authentication. • 2FA/MFA (Two Factor Authentication / Multi-Factor Authentication) • Social Authentication.	Popular Authorization Techniques- • Role-Based Access Control (RBAC) • JSON web token (JWT). • SMAL Authorization. • OpenID Authorization. • OAuth 2.0 Authorization.
The authentication credentials can be changed in parts as and when required by the user.	The authorization permissions cannot be changed be user as these are granted by the owner 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 right to resources by using roles that have been pre-defined.

Q37.What are the common problems faced in Web testing?

ANS: Security vulnerabilities: Web applications are often targeting for malicious attacks, so security testing is crucial to identify and address vulnerabilities.

- **Compatibility issues:** Web applications need to work across various browsers, devices, and operating systems. Compatibility testing ensures that the application functions correctly and displays properly across different environments.

- **Performance issues:** Poor performance, slow loading times, and high server response times can negatively impact user experience and lead to decreased engagement and customer dissatisfaction. Performance testing helps identify and address bottlenecks, scalability issues, and other performance-related problems.
- **Usability concerns:** Usability testing evaluates the user interface, navigation, and overall user experience of the web application to ensure it is intuitive, user-friendly, and meets the needs of its target audience.

Q38 To Create HLR & Test Case of Web Based (WhatsApp web, Instagram)

1. **WhatsApp Web:**
2. **Instagram:**

ANS: WhatsApp Web HLR (GitHub Link): -

Assignment-of-Software-Testing/Whatsapp HLR.xlsx at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)

WhatApp Web TestCase (GitHub Link): -

Assignment-of-Software-Testing/TEST-CASE_MANUAL_TESTING_ASSIGNMENT.xlsx at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)

Instgram TEST CASE AND HLR (GitHub Link): -

Assignment-of-Software-Testing/TEST-CASE_MANUAL_TESTING_ASSIGNMENT.xlsx at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)

Assignment-of-Software-Testing/HLR OF INSTAGRAM AND FACEBOOK.xlsx at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)

Q39 To Create HLR and TestCase on this

Link: Contact Us - ArtOfTesting

ANS: HLR :- Assignment-of-Software-Testing/HLR-ArtOfTesting.xlsx at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)

Test Case : - Assignment-of-Software-Testing/TEST-CASE_MANUAL_TESTING_ASSIGNMENT.xlsx at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)

Q40 When to used Usability Testing?

ANS: All fields on a page (For Example, text box, radio options, and drop-down lists) should be aligned properly.

- The user should not be able to type in drop-down select lists. • Tab and Shift +Tab order should work properly.
- All buttons on a page should be accessible by keyboard shortcuts and the user should be able to perform all operations using a keyboard.
- All buttons on a page should be accessible by keyboard shortcuts and the user should be able to perform all operations using a keyboard,
- All pages should have a title.
- Confirmation messages should be displayed before performing any update or delete operation.
- Hourglass should be displayed when the application is busy
- Page text should be left-justified.
- The user should be able to select only one radio option and any combination for checkboxes.
- Goal of Usability Testing are Effectiveness of the system, Efficiency, Accuracy, User Friendliness.

Q41 What is the procedure for GUI Testing?

ANS: 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.

Q41 Write a Scenario?

ANS: [All scenario is in one PDF in GitHub]

Link:- Assignment-of-Software-Testing/Document 14.pdf at main · mihirpatel03200/Assignment-of-Software-Testing (github.com)