# **Section A: Test Design & Scenario Writing (35 Marks)**

1. Microservices Scenario Design (10 marks)  
Scenario: You’re testing a feature that fetches user profile data from multiple microservices.  
Task: List at least 8 functional and integration test scenarios (consider network failures, fallback logic, partial data, etc.).

Ans -

1. **Verify if the profile data fetched is fetched**

-The profile page should show complete and correct info without any errors.

1. **Validate if the data fetched is valid/ correctly**

-Check if the profile data which is fetched is correct

1. **Verify if the request and response time(performance)**

Check if the response time acceptable

1. **Verify if any of the service is not throwing 400(Client side) errors**

Check if there is any issue from the client side

1. **Verify if the any of the service is not throwing 500(server side) errors**

Check if there is any issue from server side(checking if there is any response or no response from the server)

1. **Verify if any of the data from the profile is missing**

Check if there any missing data from the APIs like missing name, address phone number etc

1. **Verify if the login is integrated with the user profile**

Verify and validate the navigation/integration of login page>Dashboard>My Profile section

1. **Verify and validate the security of the user profile page after logging in**

**(like masking the user’s sensitive data)**

-Masking and unmasking of card details, and sensitive details etc

1. **Verify if the service fails then it throws a timeout error**

-In an exception where the system is not able to fetch the data in expected response time, it should throw timeout error

**10.Network issue( low network, No Network)**

-If the system fails for a network issue, then a proper validation message must be thrown

**2. Data-Driven Testing Strategy (5 marks)**  
Task: You need to test a complex form with 15 fields. How would you approach data-driven testing using a tool like Excel/TestNG/Postman?

Ans-

**1. Export the form details in excel sheet with all the fields**

**2. Create test data, positive and negative scenarios**

**3 .Use APACHE POI to integrate it in the automation process**

**3. UI Compatibility Scenarios (5 marks)**  
Task: List 6 advanced test scenarios for checking UI behavior across browsers, screen resolutions, and accessibility settings.

Ans-

**1. Cross Browser Testing -**

-Testing the system in multiple browsers

**2. Cross-Device Testing**

-Testing the system in multiple devices and platforms

**3. Inspect element(Dev Tools)**

-Use dev tools to check for any issues

**4. Testing in different resolutions, widely used resolutions**

-Widely used 920x1080 (Full HD), 1366x768, and 2560x1440 (Quad HD)

**5. Performance testing**

-Checking in multiple stress test scenarios, load time, scroll time, render time etc

**6. ADA (Americans with Disabilities Act) testing**

This will consist multiple sub-testing, like Zoom Testing, Tab Focus, High/low contrast, text to speech etc

**4. Test Case Review & Optimization (5 marks)**  
Task: Review 3 poorly written test cases (you may assume their structure) and suggest concrete improvements for clarity, redundancy, reusability.

Ans-

1**. Test data provided in the “Test Data” section**

-No proper test data like login, SSN or info provided in Test Cases

**2. Duplicate or recurring Test cases**

-Same test cases used multiple times

**3. Invalid Test Case design format**

**-**Invalid Test Case design format

**4. Module not differentiated**

-No differentiation in modules and submodules

**5. Test suite not well documented**

**-**Test suite does not have proper series, rows columns, components

**6. Typo mistakes and not reviewed**

-Typo mistakes and not reviewed

**7.No clarity on the Test steps**

-No clarity in writing the test steps

**5. Test Metrics Planning (5 marks)**  
Task: List 5 key QA metrics you would track during a regression cycle. Explain how each helps the team or stakeholders.

Ans-

**1. Regression test cases completion rate**

-It will show how many regression test cases are completed, passed, failed

**2. Recurring Defects**

-This will explain how many defects are recurring in Regression testing phase

**3. Priority and severity of the defects found**

-This will effectively give priority to the bugs which are more obvious and need to be fixed on a priority basis.

**4. Number of new defects found**

-List of the new defects found in the Regression cycle and will help to cross verify the old ones.

**5. Testing coverage**

-This will evaluate how much testing (%) is covered

**6. Old defects vs new defects**

- Helps in cross-verifying old defects and new defects after the regression test,

**6. Exploratory Testing Scope (5 marks)**  
Task: Given a new chat module, list 5 exploratory testing ideas and what kind of bugs each may uncover.

**Ans-**

**Let's take an example of chatbot for Zomato App**

**1. Happy path Testing for the chat module**

**-** Positive Testing with regular inputs and verifying the output

**2. Verifying if the user gets the response**

-User may not receive any response when inputs are sent

**3. Validating if the user getting proper response**

-User may receive invalid response like Order number is wrong, order details are wrong

**4. Request and response time(Performance test)**

-Response time can take multiple seconds which is not ideal

**5. Testing the UI of the whole chat module**

-The UI of chatbox can have issues

**6. Chat module integration**

-Integration of Order details with chatbot, fetching data into chatbot can have issue

**7. ”Previous chat” functionality**

-No chat history can be an issue where the requirement is to display issue when user loads chatbot

**8. Checking the responses using Keywords**

-No response on previously provided Keywords, like “I Have packaging issues” “Order Not Received” etc

# **Section B: Bug Reporting & Analysis (34 Marks)**

1. Complex Bug Report (8 marks)  
Issue: Checkout fails only when promo codes and wallet balance are combined.  
Task: Write a detailed bug report with repro steps, logs to check, impact analysis, and priority justification.

Ans-

<https://docs.google.com/document/d/1OE6wXLmoYPSw_bhR96zRxN_OaFGDJYm6dPBajWRrud8/edit?usp=sharing>

**2. Scenario Based (6 Marks)**  
Issue: Your QA team has validated a web application on the developer site, and all tests pass. However, issues are being reported by internal stakeholders on the demo site, and clients are reporting bugs on the demo/production environment. What would be your approach to investigate and handle this situation?

Ans-

**1. Document all the issues reported by the internal stakeholders**

**2. Verify and validate if there is all the new fixes and updates are deployed by the dev team**

**3. Report the bugs on priority basis if found in a different environment**

**4. Cross-verify if the bugs are already reported previously**

**6.Compare the test data used by clients and stakeholders**

**7. Cross check the environment and notify the development team about the recurring bugs in different environments**

**8. Regression and smoke test once the issues are fixed in demo/production environment**

**9. Communicate with the Development team with the help of manager**

**3. Root Cause Analysis (5 marks)**   
Task: You discovered 3 intermittent failures in test runs. How would you investigate and isolate the issue?

Ans-

**1. Report the issues**

**2. Run the tests again and check if the issues are reproduced**

**3. Check error logs**

**4. Document the issues**

**5. If the issue is related to clicks, waits can be added so that loading of the page can be handled**

**4. Defect Life Cycle Ownership (5 marks)**  
Task: Describe how you ensure a critical defect gets resolved and not ignored. Include communication and follow-up steps.

Ans-

**1. Use efficient tools provided by the company**

For communication:

**MS Team, Slack, ZoHo Cliq, Gmail etc can be used**

For Bugs tracking :

**Jira, AzureDevOps, ZoHoProjects, , Trello Tickets can be used**

* **Discussing the issues/ constraints in daily standup meeting can be beneficial**
* **Logging the issues very clearly by assigning right priority and severity**
* **Once one round of testing is completed, inform the team with the email mentioning all the passed failed TCs, severity and priority with all the data of Test Execution.**
* **Following up with Daily/ Weekly Testing Status Report with QA Team, Dev Team and managers can be an added advantage**
* **After the bug fixes check the status of the bugs and application followed by retesting**
* **Keep the track of all the issues, new, old, recurring, deferred etc**

**5. Defect Metrics Analysis (5 marks)**  
Task: List the defect metrics you’d report post-sprint (e.g., reopen rate, defect aging), how you’d present them, and actions derived from them.

**1. Total Defects/ Bugs**

-Total number of defects logged in a sprint

**2. Reopen Rate**

-How many issues were closed but reopened in later phase

**3. Defect Aging**

-How long the defects are in open state

-How much time it takes to close the defect

**4. Defect Leakage**

-How many issues are found after the production deployment

**5. Severity Count**

-How many are reported as minor, major, critical issue

**6. Flaky Tests and Reporting (5 marks)**  
Task: How do you handle automation failures that aren’t consistently reproducible?

**1. Document the flaky tests separately**

**2. Run the automated flaky automated tests separately**

**3. Report as separate bugs/ issues or add separate tags #FlakyTest**

**4. Try to find the root cause of the issues**

**5. Realtime example, some issues are with the waits, so adding custom waits for clicking a button can be a solution**

# **Section C: QA Concepts (6 Marks)**

1. Which of the following best defines regression testing?  
**Ans- B. Re-testing existing features**

2. What is confirmation testing?  
**Ans- B. Verifying that a fix resolved the issue**

3. What is the first state of a defect in the bug life cycle?  
**Ans-A. New**

4. Which testing technique uses input values at the edges?  
**Ans-B. Boundary value analysis**

5. Which testing approach is best under tight deadlines?  
**Ans-B. Risk-based testing**

6. Which metric best reflects QA effectiveness in catching defects before release?  
**Ans-B. Defect Leakage**

# 

# **Section D: Automation & Debugging Thinking (10 Marks)**

1. Selenium Code Review (5 marks)  
Task: Given a sample Selenium script (assume content), identify and fix 3 potential bugs or inefficiencies.

public class Test1 {

@Description("Open page")

@Test

@

public void test\_openVWOLoginPage(){

ChromeDriver chromeDriver = new ChromeDriver();

chromeDriver.get("https://www.google.com/");

chromeDriver.manage().window().maximize();

WebElement searchBox = chromeDriver.findElement(By.*xpath*("//\*[@id=\"APjFqb\"]"));

searchBox.sendKeys("Teknorix");

WebDriverWait wait = new WebDriverWait(chromeDriver, Duration.*ofSeconds*(5));

wait.until(ExpectedConditions.*visibilityOfAllElementsLocatedBy*(By.*xpath*("//\*[@id=\"Alh6id\"]/div[1]/div/ul/li[1]/div/div[1]")));

WebElement button = chromeDriver.findElement(By.*xpath*("/html/body/div[1]/div[3]/form/div[1]/div[1]/div[3]/center/input[1]"));

button.click();

**1. Use of semicolon/ closing of a tag**

**2. Importing the external pages, the Keywords are Case sensitive**

**3.chromeDriver.quit(); not used**

**4.Use of “ID”, multiple html elements can have same IDs**

2. Debugging Headless Failures (5 marks)  
Task: Your automation suite fails only in headless browser mode. Describe how you’d isolate and fix the issue.

Ans-

**1. Validate if the test runs in Headed mode and passes or throws same errors**

**2. Code review to check if there are any obvious issues**

**3. Adding waits for button clicks or any wait scenario can be useful**

**4. Changing the locators**

**5. Verifying and validating the logs**

# **Section E: Tools – Selenium, Postman, JIRA (15 Marks)**

1. Postman Chaining (5 marks)  
Task: Describe how you would chain API requests in Postman using environment variables and use results from one call in another.

Ans:

**> Make the first API call (ex. Login)**

**> Take the values from response (ex.Token)**

**> And use the Token for the next API call request**

2. Custom Wait in Selenium (5 marks)  
Task: Write a Java code snippet using WebDriverWait to wait for an element to be clickable.

package Teknorix;

import io.qameta.allure.Description;

import org.openqa.selenium.By;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.ExpectedConditions;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.annotations.Test;

import java.time.Duration;

public class TestWait {

@Description("Open page")

@Test

public void test\_openVWOLoginPage(){

ChromeDriver chromeDriver = new ChromeDriver();

chromeDriver.get("https://www.google.com/");

chromeDriver.manage().window().maximize();

WebElement searchBox = chromeDriver.findElement(By.*xpath*("//\*[@id=\"APjFqb\"]"));

searchBox.sendKeys("Teknorix");

WebDriverWait wait = new WebDriverWait(chromeDriver, Duration.*ofSeconds*(5));

wait.until(ExpectedConditions.*visibilityOfAllElementsLocatedBy*(By.*xpath*("/html/body/div[1]/div[2]/div/svg")));

WebElement button = chromeDriver.findElement(By.*xpath*("/html/body/div[1]/div[3]/form/div[1]/div[1]/div[3]/center/input[1]"));

button.click();

chromeDriver.quit();

}

}

**3. JIRA Defect Tracking (5 marks)**Task: Explain how you track reopened bugs, test cycles, and link test cases to issues in JIRA.  
Ans-

**1. Once the bug recurs, it is marked as reopened with the name of appropriate person assigned to**

**2. Add the comment as the bug is recurring stating more info and steps to reproduce**

**3. Using filters to sort bugs whichever is in "Open", "In Progress", "Fixed", "Reopened", "Under Review", "Approved", and "Deployed" state**

**4. Different tag can be added**

**5. You can Differentiate from specific Projects**

**6. Linking Test Cases to bugs by using Zephyr/Xray can be used**

**7. Linking proper valid evidences(Attachments) photos, screenshots videos etc to the bugs**

**8. When the bug life cycle is completed and it is fixed, add necessary passed attachments for reference and the bug is then marked closed.**