❖ **Assignment No:** 4

❖ **Problem Statement:** Create Defect Report for Any application or web application

❖ **Objective:**

- To identify and document defects found during testing of the Gmail login page.
- To ensure that all functionality, usability, and security features work correctly.
- To provide actionable information for developers to fix issues efficiently.

❖ **Hardware Requirements:** Intel Core i3 or higher Processor, 4 GB RAM, 250 GB Hard Disk

❖ **Software Requirements:** Windows 10 / Linux / macOS Operating System, Google Chrome / Mozilla Firefox / Microsoft Edge Browser, Test Management Tool (e.g., Excel, Jira)

❖ **Theory:**

- **What is a Defect Report?**

  A Defect Report is a document that records issues or bugs found during software testing. It provides details about the defect, including how to reproduce it, its severity, and priority, so that developers can understand and resolve it efficiently.

- **Purpose:**
  1. To track and manage software defects systematically.
  2. To provide a clear understanding of the issue to the development team.
  3. To improve application quality, usability, and reliability.

- **Scope:**

  The Gmail login page defects include issues related to:
  1. Input validation for email/phone and password fields
  2. Login functionality with valid and invalid credentials
  3. Security features like CAPTCHA and 2-step verification

4. Button and link functionality (“Next,” “Forgot Password,” “Create Account”)
5. Error messages and usability

- **Key Components of a Defect Report:**

  A well-documented defect report generally includes the following fields:

  1. Defect ID: Unique identifier for tracking each defect.
  2. Module: The functional area where the defect was found.
  3. Defect Summary: Short description of the issue.
  4. Defect Description: Detailed explanation with screenshots or logs (if any).
  5. Severity: Impact level (Critical, High, Medium, Low).
  6. Priority: Urgency of fixing the issue.
  7. Steps to Reproduce: Exact sequence of steps that led to the defect.
  8. Expected Result: What should have happened.
  9. Actual Result: What actually happened.
  10. Status: Current state (Open, Fixed, Retest, Closed).
  11. Assigned To: Developer or team responsible for fixing.
  12. Environment: Hardware, OS, and browser details.

- **Importance of Defect Report**

1. Helps in clear communication between testers and developers.
2. Tracks and manages defects systematically.
3. Assists in prioritizing and fixing critical issues first.
4. Ensures accountability and traceability of bugs.
5. Improves overall software quality and reliability.
6. Supports root cause analysis and process improvement.

❖ **Conclusion:**

This exercise taught me how to identify, document, and prioritize defects effectively. It highlighted the importance of defect management in ensuring software quality and user satisfaction.

❖ **DEFECT REPORT**

ID: DR-GML-001

Project: Gmail Login Functionality Testing

Product: Gmail Web Application

Release Version: v1.0.0

Module: Login and Authentication

Detected Build Version: v1.0.0-beta (Build Date: 12-Oct-2025)

Summary:

When the user attempts to log in using a valid email but an incorrect password, the application sometimes allows access without proper validation. This occurs inconsistently on certain browsers during the initial login attempt after cache clearing.

Description:

During testing of the Gmail login page, it was observed that if a valid email address is entered with an incorrect password, the system occasionally bypasses authentication and redirects to the inbox page. The issue occurs intermittently on the Chrome browser and appears related to cached session data handling. This could lead to unauthorized access and security risks.

Steps to Replicate:

1. Open the Gmail login page (https://mail.google.com).
2. Enter a valid registered email ID and click Next.
3. Enter an incorrect password and click Next/Login.
4. Observe whether access is granted or an error message appears.
5. Repeat the same after clearing cache and cookies.

Actual Result:

The application sometimes redirects the user to the inbox even with an incorrect password, bypassing proper authentication.

Expected Result:
The system should strictly validate the credentials and show an error message — “Wrong password. Try again or click Forgot password to reset it.”

Attachments:

1. Screenshot – *login_invalid_password.png*
2. Browser console log showing skipped authentication validation.
3. Network trace showing missing authentication error response.

Remarks:
Possible cause: Cached authentication token from a previous session being reused due to improper cache clearing or session management in the browser. Backend session validation or frontend cache invalidation needs review.
Defect Severity: High
Defect Priority: P1 (Critical – must fix before production release)
Reported By: QA Engineer
Assigned To: Backend Developer
Status: Open / Under Investigation
Date Reported: 12-Oct-2025
Fixed Build Version: Planned for v1.0.1 (ETA: 18-Oct-2025)

Root Cause (Preliminary Analysis):
The defect originates in the loginAuth.js module’s session handling mechanism. The frontend reuses an old session token stored in local storage even after password validation failure. The backend doesn’t invalidate this token immediately, leading to unauthorized access.

Conclusion:
This is a critical security defect that compromises user authentication integrity. Immediate fixes are required in both frontend and backend modules to ensure session tokens are invalidated after failed login attempts. The issue must be resolved before the next stable release (v1.0.1).