**Lab Exercise - Scripted Testing using JUnit**

**Objective:**

To perform **Scripted Testing** using **JUnit** by testing a **User Login System** where test cases are predefined and systematically executed.

**What is Scripted Testing?**

Scripted testing follows a **predefined set of test cases** with specific **input, expected output, and steps**. Unlike **exploratory testing**, it does not allow testers to deviate from the script.

**Scenario: User Login System**

We will test a **Login Module** that checks:

1. **Valid username and password** → "Login Successful"
2. **Invalid username or password** → "Invalid Credentials"
3. **Empty fields** → "Username and password required"
4. **Account locked after 3 failed attempts** → "Account Locked"

**Steps to Perform**

1. Implement a **Login System** in Java.
2. Write **JUnit test cases** based on a predefined script.
3. Run the tests and **validate expected vs. actual results**.

**Step 1: Implement the Login System**

Create a class **LoginSystem.java**:

import java.util.HashMap;

public class LoginSystem {

private static HashMap<String, String> users = new HashMap<>();

private static HashMap<String, Integer> failedAttempts = new HashMap<>();

static {

users.put("user1", "password123");

users.put("admin", "adminPass");

}

public static String login(String username, String password) {

if (username.isEmpty() || password.isEmpty()) {

return "Username and password required";

}

if (!users.containsKey(username)) {

return "Invalid Credentials";

}

if (failedAttempts.getOrDefault(username, 0) >= 3) {

return "Account Locked";

}

if (!users.get(username).equals(password)) {

failedAttempts.put(username, failedAttempts.getOrDefault(username, 0) + 1);

return "Invalid Credentials";

}

failedAttempts.put(username, 0); // Reset failed attempts on success

return "Login Successful";

}

}

**Step 2: Write JUnit Test Cases for Scripted Testing**

Create a test class **LoginSystemTest.java**:

import static org.junit.Assert.assertEquals;

import org.junit.Test;

public class LoginSystemTest {

@Test

public void testValidLogin() {

assertEquals("Login Successful", LoginSystem.login("user1", "password123"));

}

@Test

public void testInvalidPassword() {

assertEquals("Invalid Credentials", LoginSystem.login("user1", "wrongPass"));

}

@Test

public void testInvalidUsername() {

assertEquals("Invalid Credentials", LoginSystem.login("unknownUser", "password123"));

}

@Test

public void testEmptyUsername() {

assertEquals("Username and password required", LoginSystem.login("", "password123"));

}

@Test

public void testEmptyPassword() {

assertEquals("Username and password required", LoginSystem.login("user1", ""));

}

@Test

public void testAccountLockAfterThreeFailedAttempts() {

LoginSystem.login("user1", "wrongPass");

LoginSystem.login("user1", "wrongPass");

LoginSystem.login("user1", "wrongPass");

assertEquals("Account Locked", LoginSystem.login("user1", "password123"));

}

@Test

public void testSuccessfulLoginAfterAccountLock() {

LoginSystem.login("user1", "wrongPass");

LoginSystem.login("user1", "wrongPass");

LoginSystem.login("user1", "wrongPass");

assertEquals("Account Locked", LoginSystem.login("user1", "password123"));

assertEquals("Account Locked", LoginSystem.login("user1", "password123"));

}

}

**Step 3: Execute the Tests**

Run **JUnit 4 or JUnit 5** in your IDE (**Eclipse, IntelliJ, VS Code**).

**Expected Output (JUnit Results)**

✔ **testValidLogin()** → **Passed**  
✔ **testInvalidPassword()** → **Passed**  
✔ **testInvalidUsername()** → **Passed**  
✔ **testEmptyUsername()** → **Passed**  
✔ **testEmptyPassword()** → **Passed**  
✔ **testAccountLockAfterThreeFailedAttempts()** → **Passed**  
✔ **testSuccessfulLoginAfterAccountLock()** → **Passed**

**Key Takeaways from Scripted Testing**

✅ **Predefined test cases ensure systematic coverage.**  
✅ **Validates both normal and edge cases.**  
✅ **Repeatable and consistent testing approach.**  
✅ **Identifies issues in a structured manner.**