You have been hired to lead the automation testing efforts for an e-commerce platform.

The platform includes a web application, a mobile application, and various APIs that

handle transactions, user management, and product information.

1. Automate any web login flow using Selenium

2. Automate any POST API

3. Automate any mobile app flow (iOS or Android, any one)

1. Automating Web Login Flow Using Selenium

Prerequisites:

- **Selenium WebDriver**
- **Java (or any other language)** with a testing framework (e.g., TestNG)
- **WebDriver executable** (e.g., ChromeDriver for Chrome)

Example Code (Java + Selenium):

```
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;


public class WebLoginTestFunctionality {


  public static void main(String[] args) {
    // Set up ChromeDriver path
    System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");


    // Initialize WebDriver
    WebDriver driver = new ChromeDriver();
    // Open the e-commerce website login page
    driver.get("https://www.test.com/login");

    // Locate and interact with elements
    driver.findElement(By.id("username")).sendKeys("testuser"); // Enter username
    driver.findElement(By.id("password")).sendKeys("password123"); // Enter password
    driver.findElement(By.id("loginButton")).click(); // Click login button

    // Check if login was successful
    boolean isLoggedIn = driver.findElement(By.id("userProfile")) != null;

    if (isLoggedIn) {
      System.out.println("Login successful!");
    } else {
      System.out.println("Login failed.");
    }

    // Close the browser
    driver.quit();
  }
}
```

2. Automating a POST API Request Using REST-assured

Prerequisites:

- **REST-assured** library
- **Maven or Gradle** for dependency management
- **API endpoint** and credentials

```java
import io.restassured.RestAssured;
import io.restassured.response.Response;
import static io.restassured.RestAssured.given;

public class PostAPITest {

    public static void main(String[] args) {
        // Set the base URL for the API
        RestAssured.baseURI = "https://api.example.com";

        // Define the request body (in JSON format)
        String requestBody = "{\n" +
                "  \"username\": \"testuser\",\n" +
                "  \"password\": \"password123\",\n" +
                "  \"email\": \"testuser@example.com\"\n" +
                "}";

        // Send the POST request
        Response response = given()
                .header("Content-Type", "application/json")
                .body(requestBody)
                .post("/users");

        // Validate the response status code
        int statusCode = response.getStatusCode();
        if (statusCode == 201) {
            System.out.println("User created successfully.");
        } else {
            System.out.println("Failed to create user. Status code: " + statusCode);
        }
    }
}
```

3. Automating Mobile App Flow Using Appium (Android)

Prerequisites:

- **Appium** (for mobile automation)
- **Android Studio** (for device/emulator setup)
- **Java + Appium dependencies**

```
import io.appium.java_client.MobileElement;
import io.appium.java_client.android.AndroidDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import java.net.MalformedURLException;
import java.net.URL;

public class MobileAppLoginTest {

    public static void main(String[] args) throws MalformedURLException {
        // Set up desired capabilities for Android
        DesiredCapabilities caps = new DesiredCapabilities();
        caps.setCapability("deviceName", "Android Emulator");
        caps.setCapability("platformName", "Android");
        caps.setCapability("app", "path/to/your/app.apk"); // Path to the APK file
        caps.setCapability("automationName", "UiAutomator2"); // Automation engine for Android
    }

        // Initialize Android driver with Appium server URL
        AndroidDriver<MobileElement> driver = new AndroidDriver<>(new
URL("http://localhost:4723/wd/hub"), caps);

        // Perform login steps (example with an e-commerce app login)
        MobileElement usernameField = driver.findElementById("com.example:id/username");
        MobileElement passwordField = driver.findElementById("com.example:id/password");
        MobileElement loginButton = driver.findElementById("com.example:id/loginButton");

        usernameField.sendKeys("testuser");
        passwordField.sendKeys("password123");
        loginButton.click();

        // Add assertion for login success (example: check if dashboard is loaded)
        MobileElement dashboard = driver.findElementById("com.example:id/dashboard");
        if (dashboard.isDisplayed()) {
            System.out.println("Login successful!");
        } else {
            System.out.println("Login failed.");
        }

        // Close the session
        driver.quit();
    }
```

```
}
```