

Task 2: Automation Code

Setting Up Your Automation Framework

1. Create a Maven Project

- Use an IDE like IntelliJ IDEA or Eclipse.
- Create a new Maven project and add dependencies for Selenium and TestNG in your pom.xml.

```
<dependencies>
  <dependency>
    <groupId>org.seleniumhq.selenium</groupId>
    <artifactId>selenium-java</artifactId>
    <version>4.x.x</version>
  </dependency>
  <dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>7.x.x</version>
    <scope>test</scope>
  </dependency>
</dependencies>
```

2. Create Base Test Class:

- This class will handle browser setup and teardown.

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.testng.annotations.AfterClass;
import org.testng.annotations.BeforeClass;

public class BaseTest {
    protected WebDriver driver;

    @BeforeClass
```

```

public void setUp() {
    System.setProperty("webdriver.chrome.driver", "path/to/chromedriver");
    driver = new ChromeDriver();
    driver.get("https://dev-dash.janitri.in");
}

@AfterClass
public void tearDown() {
    driver.quit();
}
}

```

3. Create Login Page Class:

- Implement the Page Object Model.

```

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;

public class LoginPage {
    private WebDriver driver;

    // Locators
    private By userIdInput = By.id("userId");
    private By passwordInput = By.id("password");
    private By loginButton = By.id("loginButton");
    private By passwordToggle = By.id("passwordToggle");

    public LoginPage(WebDriver driver) {
        this.driver = driver;
    }

    public void enterUserId(String userId) {
        driver.findElement(userIdInput).sendKeys(userId);
    }

    public void enterPassword(String password) {
        driver.findElement(passwordInput).sendKeys(password);
    }
}

```

```
public void clickLogin() {  
    driver.findElement(loginButton).click();  
}  
public void togglePasswordVisibility() {  
    driver.findElement(passwordToggle).click();  
}  
}
```

4. Create Test Class:

- Implement the test cases using TestNG.

```
import org.testng.annotations.Test;  
public class LoginPageTest extends BaseTest {  
    private LoginPage loginPage;  
    @BeforeClass  
    public void setUp() {  
        super.setUp();  
        loginPage = new LoginPage(driver);  
    }  
    @Test  
    public void testLoginButtonDisabledWhenFieldsAreEmpty() {  
        // Implement test logic  
    }  
    @Test  
    public void testInvalidLoginShowErrorMsg() {  
        // Implement test logic  
    }  
    @Test  
    public void testPasswordMaskedButton() {  
        // Implement test logic  
    }  
}
```