Selenium, Cucumber, and TestNG Assignment

Background: You are a **software development engineer in test** at a major IT consulting company. Your company's client is a business called Mercury Tours (http://demo.guru99.com/test/newtours/). Your manager has assigned you a project to create a new E2E testing framework from scratch which the rest of team will use as a foundation for automation testing of Mercury Tour's website in order to automatically and easily validate all major website functionalities. Your objective is to set up a Java project that will serve as the basic framework for the future tests that will be written. To serve as an example for other employees, you will also write tests for the Login and Logout features.

Framework requirements

- **TestNG** for running the test cases
- **Cucumber** for behavior driven development (BDD)
- **Selenium** to automate the browser

Setup

1. Ensure that your IDE has the required plugins. This will require you to go the Eclipse Marketplace in Spring Tool Suite and install these plugins:
    - Cucumber Eclipse plugin: Enables you to execute the feature files directly from the IDE itself
    - TestNG for Eclipse: Enables you to execute TestNG tests from Eclipse as well as generating the testng.xml configuration file
2. Create a Maven project and include the required dependencies for TestNG, Selenium, and Cucumber
3. Download the appropriate chromedriver for your version of chrome and put the chromedriver.exe file in a location that you remember

Features to be tested:

- Login
  - Scenario: Login unsuccessful
  - Scenario: Login successful
    - Use mercury for username
    - Use mercury for password
- Logout
  - Scenario: Logout successful

Instructions:

1. Create 4 packages in your project:
    - com.revature.features
    - com.revature.gluecode
    - com.revature.pages
    - com.revature.testrunners
2. Inside of the com.revature.pages package
    - Create a class called HomePage

```
public class HomePage {

	private WebDriver driver;

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

	public WebElement usernameInput() {
		return this.driver.findElement(By.name("userName"));
	}

	public WebElement passwordInput() {
		return this.driver.findElement(By.name("password"));
	}

	public WebElement submitButton() {
		return this.driver.findElement(By.name("submit"));
	}

	public WebElement unsuccessfulLoginElement() {
		return this.driver.findElement(By.xpath("//span[contains(text(), 'Enter your userName and password correct')]"));
	}

}
```

    - Create a class called LandingPage

```java
public class LandingPage {

	private WebDriver driver;

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

	public WebElement loginSuccessfulHeader() {
		return this.driver.findElement(By.xpath("//h3[text()='Login Successfully']"));
	}

	public WebElement signoffButton() {
		return this.driver.findElement(By.linkText("SIGN-OFF"));
	}

}
```

3. Inside of the com.revature.testrunners package,
    - Create a class called TestRunner
    - Insert the following code:

```java
@CucumberOptions(glue = "com.revature.gluecode",
features = { "src/test/java/com/revature/features/login.feature",
		"src/test/java/com/revature/features/logout.feature" })
public class TestRunner extends AbstractTestNGCucumberTests {

	public static WebDriver driver;
	public static HomePage homePage;
	public static LandingPage landingPage;

	@BeforeMethod
	public void setupDriver() {
		System.setProperty("webdriver.chrome.driver", "C:/webdrivers/chromedriver.exe");

		ChromeOptions options = new ChromeOptions();
		options.addArguments("--incognito");

		TestRunner.driver = new ChromeDriver(options);

		TestRunner.homePage = new HomePage(driver);
		TestRunner.landingPage = new LandingPage(driver);
	}

	@AfterMethod
	public void quitDriver() {
		TestRunner.driver.quit();
	}

}
```

4. Create the feature files inside com.revature.features
    - login.feature
    - logout.feature
5. Create the gluecode classes inside com.revature.gluecode
    - LoginDefinition
    - LogoutDefinition
6. Then wait for me (Bach) to facilitate discussion on how we will approach writing the scenarios for each feature and the steps.