Selenium + Java + Cucumber BDD Framework

Project Structure

```
SeleniumCucumberBDDFramework/
■■■ src/main/java/
■   ■■■ pages/
■   ■   ■■■ LoginPage.java
■   ■■■ factory/
■   ■   ■■■ DriverFactory.java
■   ■■■ utilities/
■   ■   ■■■ ConfigReader.java
■
■■■ src/test/java/
■   ■■■ stepdefinitions/
■   ■   ■■■ LoginSteps.java
■   ■■■ runners/
■   ■   ■■■ TestRunner.java
■   ■■■ hooks/
■   ■   ■■■ ApplicationHooks.java
■
■■■ src/test/resources/
■   ■■■ features/
■   ■   ■■■ Login.feature
■   ■■■ config/
■   ■   ■■■ config.properties
■   ■■■ testdata/
■       ■■■ LoginData.xlsx  (optional for DDT)
■
■■■ pom.xml
■■■ README.md
```

Feature File (Login.feature)

```
Feature: Login functionality

  Scenario: Valid login with correct credentials
    Given user is on login page
    When user enters username "admin" and password "password123"
    And clicks on login button
    Then user should see the home page
```

Page Object Model (LoginPage.java)

```
public class LoginPage {
    private WebDriver driver;

    private By usernameField = By.id("username");
    private By passwordField = By.id("password");
    private By loginBtn = By.id("login");
    private By homePageLogo = By.id("logo");

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

    public void enterUsername(String username) { driver.findElement(usernameField).sendKeys(username
    public void enterPassword(String password) { driver.findElement(passwordField).sendKeys(password
    public void clickLogin() { driver.findElement(loginBtn).click(); }
    public boolean isHomePageDisplayed() { return driver.findElement(homePageLogo).isDisplayed(); }
}
```

Driver Factory (DriverFactory.java)

```
public class DriverFactory {
    private static WebDriver driver;

    public static WebDriver initDriver() {
        if (driver == null) {
            driver = new ChromeDriver();
            driver.manage().window().maximize();
        }
        return driver;
    }

    public static WebDriver getDriver() { return driver; }
    public static void quitDriver() { if (driver != null) { driver.quit(); driver = null; } }
}
```

Config Reader (ConfigReader.java)

```
public class ConfigReader {
    private Properties prop;

    public Properties initProp() {
        prop = new Properties();
        try {
            FileInputStream ip = new FileInputStream("src/test/resources/config/config.properties");
            prop.load(ip);
        } catch (IOException e) {
            e.printStackTrace();
        }
        return prop;
    }
}
```

Hooks (ApplicationHooks.java)

```
public class ApplicationHooks {
    private WebDriver driver;
    private ConfigReader configReader;
    Properties prop;

    @Before
    public void launchBrowser() {
        configReader = new ConfigReader();
        prop = configReader.initProp();
        driver = DriverFactory.initDriver();
        driver.get(prop.getProperty("url"));
    }

    @After
    public void quitBrowser() {
        DriverFactory.quitDriver();
    }
}
```

Step Definitions (LoginSteps.java)

```
public class LoginSteps {
    private LoginPage loginPage = new LoginPage(DriverFactory.getDriver());

    @Given("user is on login page")
    public void user_is_on_login_page() {}

    @When("user enters username {string} and password {string}")
    public void user_enters_username_and_password(String username, String password) {
        loginPage.enterUsername(username);
        loginPage.enterPassword(password);
    }

    @When("clicks on login button")
    public void clicks_on_login_button() { loginPage.clickLogin(); }

    @Then("user should see the home page")
    public void user_should_see_the_home_page() {
        Assert.assertTrue(loginPage.isHomePageDisplayed());
    }
}
```

Test Runner (TestRunner.java)

```
@RunWith(Cucumber.class)
@CucumberOptions(
        features = "src/test/resources/features",
        glue = {"stepdefinitions", "hooks"},
        plugin = {"pretty", "html:target/cucumber-reports.html"},
        monochrome = true
)
public class TestRunner {}
```

Config File (config.properties)

```
url=https://example.com/login
browser=chrome
username=admin
password=password123
```

Interview Explanation

My Selenium + Java + Cucumber BDD framework is structured with features written in Gherkin,
stepdefinitions implementing steps, and pages following Page Object Model.
Driver setup is handled in factory, reusable functions in utilities, and global setup/teardown in ho
Execution is managed via runners, and reports are generated automatically.
This modular approach makes the framework maintainable, reusable, and CI/CD friendly.