How to explain Test Automation Framework to the interviewer.

We need to specify in and out of our Test Automation Framework such as programming **language** used, **Type of framework** used, Test Base Class (Initializing WebDriver, Implicit Waits), How we separate Element locators and tests (Page Objects, Page Factory), Utility functions file, Property files, TestNG annotations, How we parameterize tests using Excel files, How we capture error screenshots, Generating reports(Extent Reports), Emailing reports, Version Control System used and Continues Integration Tool used.

**Language:**In our Selenium Project we are using Java language. Even though Selenium supports multiple languages, we are using Java language is just because most of the automation developers have knowledge on Selenium with Java.

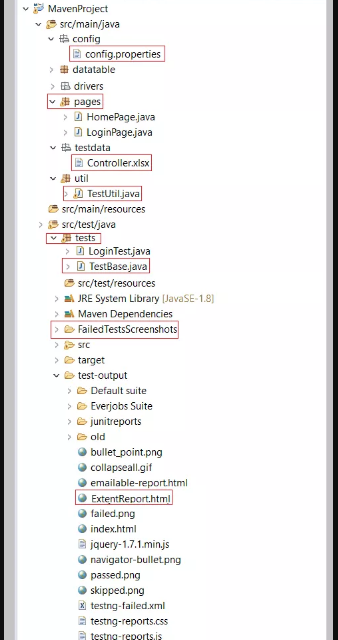
**Type of Framework:** In our project, we are using Data-driven Framework by using Page Object Model design pattern with Page Factory.

**POM:**As per the Page Object Model, we have maintained a class for every web page. Each web page has a separate class and that class holds the functionality and members of that web page. Separate classes for every individual test.

**Packages:** We have separate packages for Pages and Tests. All the web page related classes come under **Pages** package and all the tests related classes come under **Tests** package.

For example, Home Page and Login Page have a separate classes to store element locators. For the login test there would be a separate class which calls the methods from the Home Page class and Login Page class.

I will explain based on the below mentioned test automation framework structure.



Above screenshot illustrates standardized maven project. As per the above maven project, all the tests are kept in the ‘***src/test/java***‘ and remaining files (such as config.properties, element locators (POM classes), utility files, test data, etc.,) kept under ‘***src/main/java***‘.

**Test Base Class:**Test Base class (TestBase.java) deals with all the common functions used by all the pages. This class is responsible for loading the configurations from properties files, Initializing the WebDriver, Implicit Waits, Extent Reports and also to create the object of FileInputStream which is responsible for pointing towards the file from which the data should be read.

**Utility Class ():**Utility class (TestUtil.java) stores and handles the functions (The code which is repetitive in nature such as waits, actions, capturing screenshots, accessing excels, sending email etc.,) which can be commonly used across the entire framework. The reason behind creating utility class is to achieve reusability. This class extends the TestBase class to inherit the properties of TestBase in TestUtil.

**Properties file:**This file (**config.properties**) stores the information that remains static throughout the framework such as browser specific information, application URL, screenshots path etc.

All the details which change as per the environment and authorization such as URL, Login Credentials are kept in the config.properties file. Keeping these details in a separate file makes easy to maintain.

**Screenshots:** Screenshots will be captured and stored in a separate folder and also the screenshots of a failed test cases will be added in the extent reports.

**Test Data:** All the historical test data will be kept in excel sheet (controller.xlsx). By using ‘controller.xlsx’, we pass test data and handle data driven testing. We use Apache POI to handle excel sheets.

**TestNG:** Using TestNG for Assertions, Grouping and Parallel execution.

**Maven:** Using Maven for build, execution and dependency purpose. Integrating the TestNG dependency in POM.xml file and running this POM.xml file using Jenkins.