Page Object Model and Page Factory

- In **Page Object Model,** we store the locators of the web elements and the reusable methods of the tests in separate Page Classes
- **Page Object Model** is a design pattern, when applied to the existing frameworks will result in the below:
  - Easy Maintenance
  - Code Readability
- Demo Application - **Zoho.com**
- **Create the framework using the PageObjectModel and PageFactory Design Patterns:**
  - Create a new Maven Project with Artifact id 'PomProj' and Group Id 'pomPack'
  - Delete the auto generated App.java files under the packages
  - Create a new Source Folder src/test/resources
  - Provide the required dependencies in pom.xml file and save the project - View required dependencies here
  - Create the below sub-packages under 'src/main/java' and Create the 'tests' sub-package under 'src/test/java'
    - base, util and pages packages
  - Download the ExtentManager.java file and put it under 'util' sub package - Download from here
  - Create 'reports' folder under the Project as specified in ExtentManager.java
  - Copy the ReportsConfig.xml file under the Project - Download ReportsConfig.xml file here
  - Create a Class say 'LoginTest' under the 'tests' package of 'src/test.java'
  - Create three Classes say 'LaunchPage', 'LoginPage' and 'HomePage' under the 'pages' package of 'src/main/java'
  - Open LaunchPage Class and create the below:
    - Create WebElement for 'Login' link option - View here
    - Create a reusable method for navigating to Login page of Zoho.com say 'goToLoginPage()' - View here
  - Open LoginPage Class and create the WebElements & Reusable methods - View here

---