SmartCliff

CAREER MOBILITY SOLUTIONS

We are on a mission to address the digital skills gap for 10 Million+ young professionals, train and empower them to forge a career path into future tech

Page Object Model & Page Factory

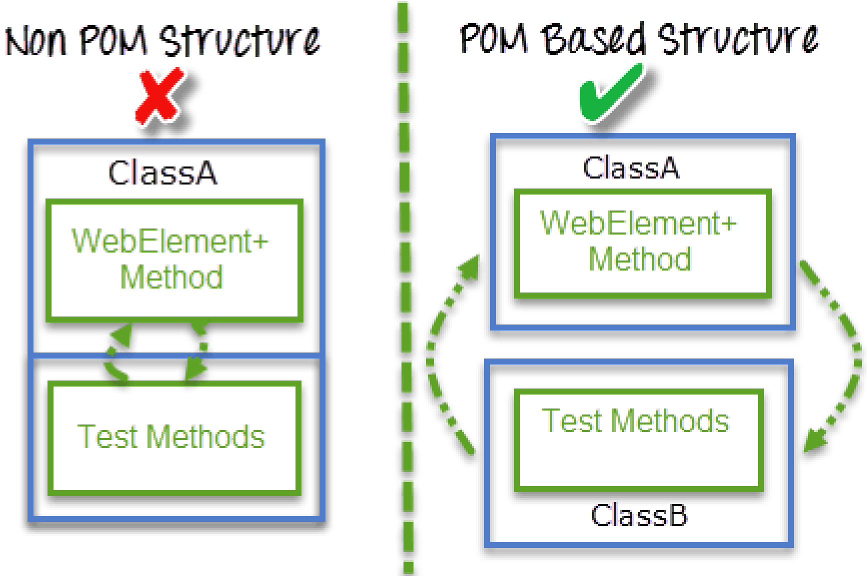
Contents

- Introduction to Page Object Model
- Annotation
- Advantages and Disadvantages of Page Object Model
- Quiz

Introduction of Page Object Model

Introduction of Page Object Model

- Page Object Model (POM) is a design pattern, popularly used in test automation that creates Object Repository for web UI elements.
- The advantage is it reduces code duplication .
- In this model each web page of the application is considered as a separate class file
- In this model each web page of the application is considered as a separate class file



Introduction of Page Object Model (Contd.)

- The Page class will identity the web elements of that web page and also contains page methods which perform operations on that web elements

- **Page Object Model (POM)** is a design pattern, popularly used in test automation that creates Object Repository for web UI elements.

- The advantage of the model is that it reduces code duplication and improves test maintenance.

Introduction of Page Object Model (Contd.)

- Under this model, for each web page in the application, there should be a corresponding Page Class.
- This **Page class will identify the WebElements of that web page** and also contains Page methods which perform operations on those WebElements.
- Name of these methods should be given as per the task they are performing, i.e., if a loader is waiting for the payment gateway to appear, POM method name can be waitForPaymentScreenDisplay().

Introduction of Page Object Model (Contd.)

- Under this model, for each web page in the application, there should be a corresponding Page Class.
- This **Page class will identify the WebElements of that web page** and also contains Page methods which perform operations on those WebElements.
- Name of these methods should be given as per the task they are performing, i.e., if a loader is waiting for the payment gateway to appear, POM method name can be waitForPaymentScreenDisplay().

Annotations

Annotations

@FindBy Annotation

- The @FindBy annotation is used in Page Objects in Selenium tests to specify the object location strategy for a WebElement or a list of WebElements
- Using the PageFactory, these WebElements are usually initialized when a Page Object is created.
- The @FindBy annotation is used to locate one or more WebElements using a single criterion

@FindBy Annotation

Example:

```
public class GooglePage
{     //WebElemewnt searchBox= driver.findElement(By.name(“q”));
       @FindBy(how = How.NAME, using = "q")
       private WebElement searchBox;
       public void searchFor(String text)
       {
              searchBox.sendKeys(text);
              searchBox.submit();
        }
}
```

@FindBy Annotation type 2

Example:

```
public class GooglePage
{      //WebElemewnt searchBox= driver.findElement(By.name(“q”));
               @FindBy(xpath="//*[@name=‘q']")
       private WebElement searchBox;
       public void searchFor(String text)
       {
              searchBox.sendKeys(text);
              searchBox.submit();
        }
}
```

Advantages and Disadvantages of Page Object Model

Advantages and Disadvantages of Page Object Model

Advantages of Page Object Model

1.Object Repository:

- You can create an Object Repository of the fields segmented page-wise.
- This as a result provides a Page Repository of the application as well.
- Each page will be defined as a java class.
- All the fields in the page will be defined in an interface as members. The class will then implement the interface.

2.Functional Encapsulation:

- All possible functionality or operations that can be performed on a page can be defined and contained within the same class created for each page.
- This allows for clear definition and scope of each page's functionality.

Advantages of Page Object Model

3.Low maintenance:

- Any User Interface changes can swiftly be implemented into the interface as well as class.

4.Programmer Friendly:

- Robust and more readable.
- The Object-oriented approach makes the framework programmer friendly.

5.Low Redundancy:

- Helps reduce duplication of code.
- If the architecture is correctly and sufficiently defined, the POM gets more done in less code.

6.Efficient & Scalable:

- Faster than other keyword-driven/data-driven approaches where Excel sheets are to be read/written.

Disadvantages of Page Object Model

1.High Setup Time & Effort:

- Initial effort investment in development of Automation Framework is high.
- This is the biggest weight of POM in case of web applications with hundreds/thousands of pages.
- It is highly suggested that if this model is decided to be implemented, then it should be done parallel to development of the application. Refer V-Model for Software Development Life Cycle.

2.Skilled labor:

- Testers not technically sound or aware of programming best practices are a nightmare in this case. Perhaps this is the biggest mistake to make, employing unskilled labor in hopes of training them during implementation.
- Unskilled testers need to undergo a Training Boot Camp to be ready for such an undertaking.
- Also the Architecture of the framework should be defined clearly and completely before development in order to avoid any loopholes in later stages.

Disadvantages of Page Object Model

- Every application is different and it may require the automation framework to be significantly tailored towards it.

3.Specific:

- Not a generic model. Automation Framework developed using POM approach is specific to the application. Unlike keyword-driven/data-driven frameworks, it is not a generic framework.

Page Object Model

Driver class page – Testng1.java

```
public class Testng1 {


                    //creating object for Webdriver and Loginpage1
  WebDriver driver;
  Loginpage1 objlogin;
  Homepage1 objHome;
                    @Test
  public void logintest() {
objlogin=new Loginpage1(driver);
objlogin.perfomLoginAction("Admin", "admin123");
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(3));
Assert.assertEquals(driver.getCurrentUrl(),
"https://opensource-demo.orangehrmlive.com/web/index.php/dashboard/index");
objHome=new Homepage1(driver);
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(3));
objHome.clickadmintab();
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(3));

                    Assert.assertEquals(driver.getCurrentUrl(),
"https://opensource-demo.orangehrmlive.com/web/index.php/admin/viewSystemUsers");
                    }
@BeforeTest
public void beforetestmethod() {
WebDriverManager.chromedriver().setup();
driver=new ChromeDriver();
driver.get("https://opensource-demo.orangehrmlive.com/web/index.php/auth/login");
driver.manage().window().maximize();
driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(3));
}}
```

Page Object Model

Elements Object page - Loginpage1.java

```
public class Loginpage1 {
WebDriver driver;
//By object for Web elements by corresponding locators
By username=By.name("username");
By password= By.name("password");
By loginbtn=By.xpath("//button[@type='submit']");

//constructor to instantiate driver
  public Loginpage1(WebDriver driver) {
this.driver=driver;
}
  // setting the username
  public void setusername(String uname) {
                    driver.findElement(username).sendKeys(uname);
}
```

```
 public void setpassword(String pwd) {
  driver.findElement(password).sendKeys(pwd);
}
  public void clickloginbtn() {
                    driver.findElement(loginbtn).click();
}

  public void perfomLoginAction(String uname,String pwd) {
                    this.setusername(uname);
                    this.setpassword(pwd);
                    this.clickloginbtn();

}

}
}}
```

Page Factory

Definition

Page Factory is a class provided by Selenium WebDriver to support Page Object Design patterns. In Page Factory, testers use @FindBy annotation.

The **initElements** method is used to initialize web elements.

@FindBy: An annotation used in Page Factory to locate and declare web elements using different locators. Below is an example of declaring an element using **@FindBy**@FindBy(id="elementId") WebElement element;

Below are locators that can be used:

- **ClassName**
- **CSS**
- **Name**
- **Xpath**
- **TagName**
- **LinkText**
- **PartialLinkText**

Page Factory

Program 1

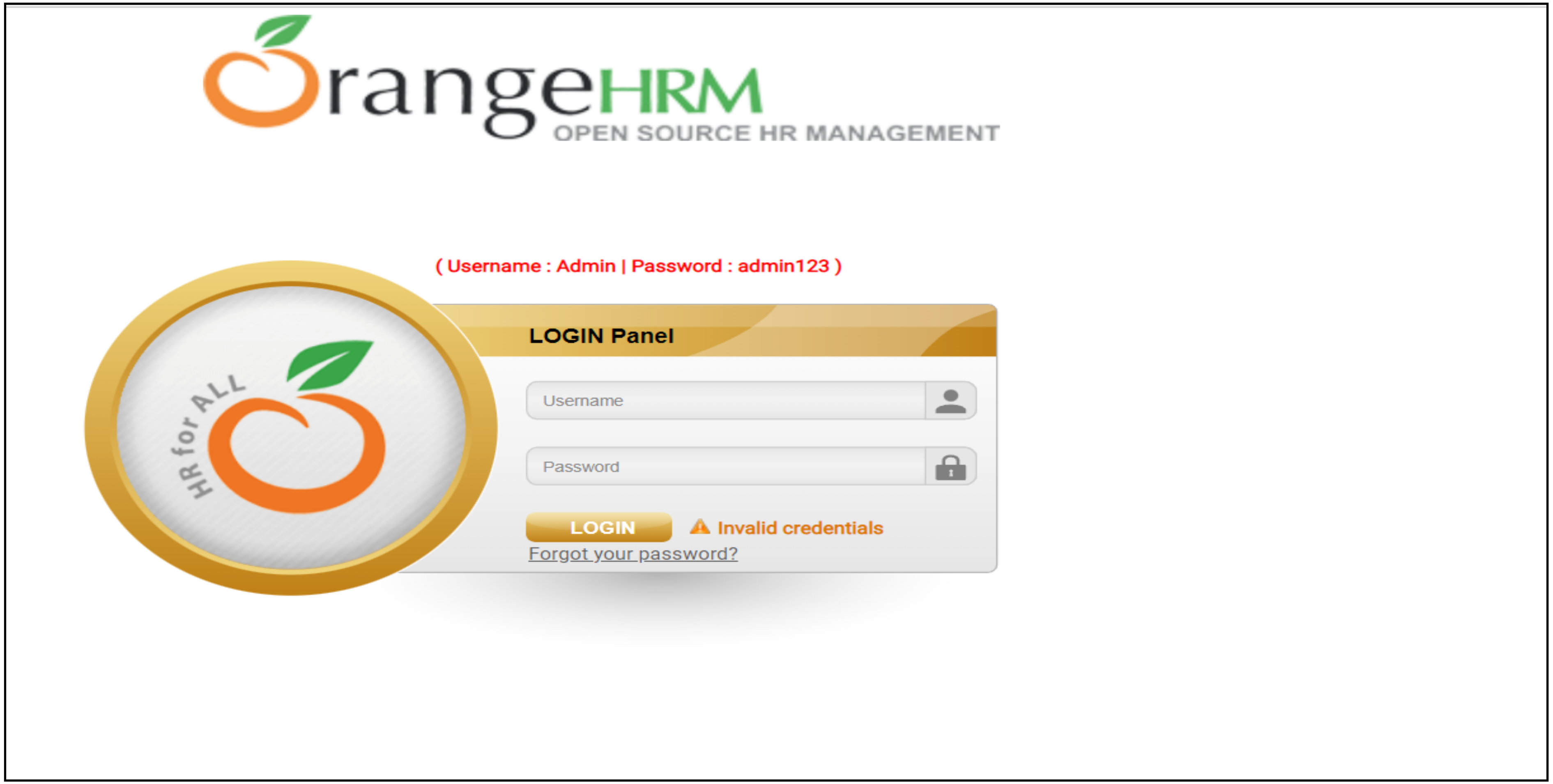
```
//driver class
public class LoginwithPageFactory {
@Test
public void Login1() throws InterruptedException {
System.setProperty("webdriver.chrome.driver","E:\\Testing\\chromedriver.exe");
WebDriver driver = new ChromeDriver();
driver.get("https://opensource-demo.orangehrmlive.com/index.php/validatecredentials");

PageFactory.initElements(driver, Loginpageobjects1.class);
Thread.sleep(7000);
Loginpageobjects1.username.sendKeys("Admin");
        Loginpageobjects1.password.sendKeys("admin123");
        Loginpageobjects1.btnLogin.click();
//driver.quit();
}}
```

Page Factory

Program 1

```
// Loginpageobjects1.java
public class Loginpageobjects1 {

@FindBy(id="txtUsername")
public static WebElement username;

@FindBy(id="txtPassword")
public static WebElement password;

@FindBy(id="btnLogin")
public static WebElement btnLogin;



}
```

Output

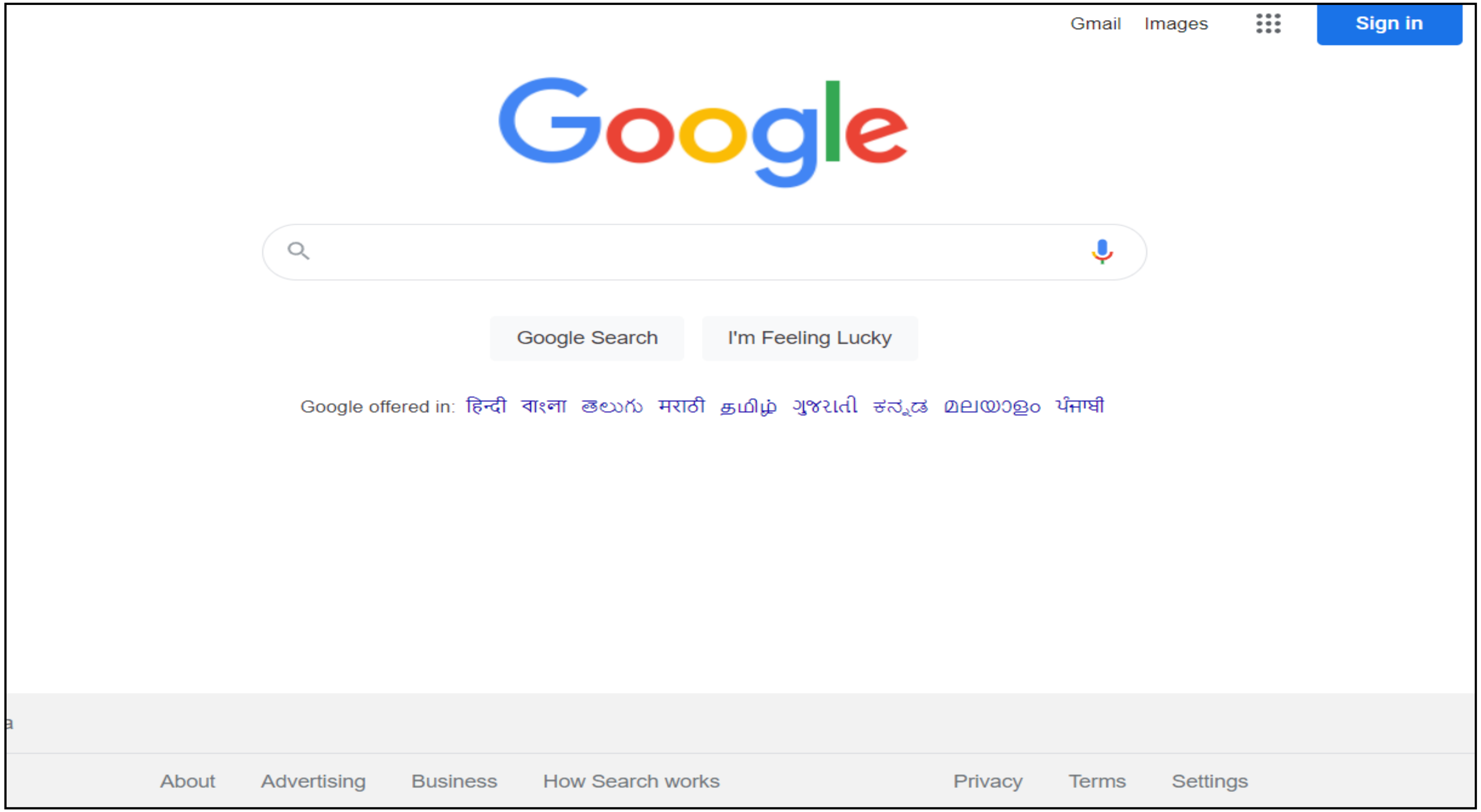

(Username : Admin | Password : admin123)

⚠ Invalid credentials

Program 1

```
public class Screenshot1 {
    @Test
    public void TakeScreenShot1() throws Exception{
WebDriver driver ;
    System.setProperty("webdriver.chrome.driver","c:\\Drivers\\chromedriver.exe");
    driver = new ChromeDriver();
           driver.get("https://www.google.com");
          String fileWithPath="c://swarna/ss.png";
            TakesScreenshot scrShot =(TakesScreenshot)driver;
                File SrcFile=scrShot.getScreenshotAs(OutputType.FILE);
                File DestFile=new File(fileWithPath);
                FileUtils.copyFile(SrcFile, DestFile);
                   }
}
```

Output


Gmail
Images
Sign in
Google
Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
a
About
Advertising
Business
How Search works
Privacy
Terms
Settings

Quiz

1) Which locator is used by @FindBy annotation?

a) css

b) xpath

c) TagName

d) All the above

Answer : Option d)

Quiz

2) POM is a

a) Design Pattern

b) Interface

c) Package

d) Function

Answer : Option a)

Quiz

3) Which method is used to initialize driver in Page Object Model?

a) init()

b) initDriver()

c) initElement()

d) initElements()

Answer : Option d)

THANK YOU

SmartCliff
Career Mobility Solutions