```
package rahulshettyacademy.TestComponents;

import org.testng.annotations.AfterMethod;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.time.Duration;
import java.util.HashMap;
import java.util.List;
import java.util.Properties;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.Dimension;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.edge.EdgeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;

import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.ObjectMapper;

import io.github.bonigarcia.wdm.WebDriverManager;
import rahulshettyacademy.pageobjects.LandingPage;

public class BaseTest {

        public WebDriver driver;
        public LandingPage landingPage;

        public WebDriver initializeDriver() throws IOException

        {
                // properties class

                 Properties prop = new Properties();
                FileInputStream fis = new
FileInputStream(System.getProperty("user.dir")
                                    +
"//src//main//java//rahulshettyacademy//resources//GlobalData.properties");
                prop.load(fis);

                String browserName = System.getProperty("browser")!=null ?
```

```
System.getProperty("browser") :prop.getProperty("browser");
                //prop.getProperty("browser");

                if (browserName.contains("chrome")) {
                        ChromeOptions options = new ChromeOptions();
                        WebDriverManager.chromedriver().setup();
                        if(browserName.contains("headless")){
                        options.addArguments("headless");
                        }
                        driver = new ChromeDriver(options);
                        driver.manage().window().setSize(new
Dimension(1440,900));//full screen

                } else if (browserName.equalsIgnoreCase("firefox")) {
                        System.setProperty("webdriver.gecko.driver",

"/Users/rahulshetty//documents//geckodriver");
                        driver = new FirefoxDriver();
                        // Firefox
                } else if (browserName.equalsIgnoreCase("edge")) {
                        // Edge
                        System.setProperty("webdriver.edge.driver", "edge.exe");
                        driver = new EdgeDriver();
                }

                driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
                driver.manage().window().maximize();
                return driver;

        }

        public List<HashMap<String, String>> getJsonDataToMap(String filePath)
throws IOException
        {
                //read json to string
        String jsonContent =    FileUtils.readFileToString(new File(filePath),
                        StandardCharsets.UTF_8);

        //String to HashMap- Jackson Databind

        ObjectMapper mapper = new ObjectMapper();
          List<HashMap<String, String>> data = mapper.readValue(jsonContent, new
TypeReference<List<HashMap<String, String>>>() {
      });
          return data;

        //{map, map}

        }
```

```
        public String getScreenshot(String testCaseName,WebDriver driver) throws
IOException
        {
                TakesScreenshot ts = (TakesScreenshot)driver;
                File source = ts.getScreenshotAs(OutputType.FILE);
                File file = new File(System.getProperty("user.dir") + "//reports//"
+ testCaseName + ".png");
                FileUtils.copyFile(source, file);
                return System.getProperty("user.dir") + "//reports//" +
testCaseName + ".png";


        }

        @BeforeMethod(alwaysRun=true)
        public LandingPage launchApplication() throws IOException
        {

                 driver = initializeDriver();
                  landingPage = new LandingPage(driver);
                landingPage.goTo();
                return landingPage;


        }

        @AfterMethod(alwaysRun=true)

        public void tearDown()
        {
                driver.close();
        }
}
```