```java
package com.ProjectClassTask;

import java.io.File;

public class NykaTask  {

    public static void main(String[] args) throws InterruptedException, IOException {

        System.setProperty("webdriver.chrome.driver", ".\\driver\\chromedriver.exe");
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.nykaa.com/");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        WebElement findElement = driver.findElement(By.xpath("//button[text()='Sign in']"));
        findElement.click();


        WebElement findElement2 = driver.findElement(By.xpath("//button[contains(text(),'Sign in with Mobile / Email')]"));
        findElement2.click();

        Thread.sleep(2000);

        WebElement findElement3 = driver.findElement(By.xpath("//input[@name='emailMobile']"));
        findElement3.sendKeys("9876543210");

        WebElement findElement4 = driver.findElement(By.xpath("//button[@type='submit']"));
        findElement4.click();
        //SCANNER
        Scanner sc= new Scanner(System.in);
        System.out.println("OTP");
        String otp = sc.nextLine();

        WebElement findElement5 = driver.findElement(By.name("otpValue"));
        findElement5.sendKeys(otp);

        WebElement findElement6 = driver.findElement(By.xpath("//button[contains(@class,'button big fill full')]"));
        findElement6.click();

        WebElement findElement7 = driver.findElement(By.xpath("//div[@id='my-menu']//a[contains(text(),'make')]"));
        //MOUSE ACTIONS
        Actions a= new Actions(driver);
        a.moveToElement(findElement7).build().perform();
        WebElement findElement8 = driver.findElement(By.xpath("//div[@id='my-menu']//a[contains(text(),'Face Primer')]//following::li[1]//a"));
        findElement8.click();

        Thread.sleep(5000);

        //WINDOWS HANDLING
        Set<String> windowHandles = driver.getWindowHandles();
        Iterator iterator = windowHandles.iterator();
        String firstWin = (String) iterator.next();
        String secondWin = (String) iterator.next();
        driver.switchTo().window(secondWin);

        Thread.sleep(5000);

        //JAVA SCRIPT EXECUTOR
        JavascriptExecutor js = (JavascriptExecutor) driver;
        js.executeScript("window.scroll(0,600);", ""); // scrolldown

        //JAVASCRIPT -SCRIPT TIMEOUT EXCEPTION HANDLING
        try {
            WebElement findElement9 = driver
                    .findElement(By.xpath("//*[@id=\"product-list-wrap\"]/div[1]/div/div[2]/div/button[2]"));

            js.executeAsyncScript("arguments[0].click();", findElement9);

        } catch (Exception e) {
            // script timeout exception
        }
        System.out.println("preview shade");

       //ADD TO BAG CLICK
        WebElement findElement10 = driver
                .findElement(By.xpath("//*[@id=\"product-list-wrap\"]/div[1]/div/div[2]/div/button/span"));
        findElement10.click();
        System.out.println("add to bag");

       //SVG --BUTTON CLICK
        WebElement findElement11 = driver
                .findElement(By.xpath("//*[@id=\"header_id\"]/div[2]/div/div[2]/div[2]/button"));
        findElement11.click();
        System.out.println("moved");

        //TAKESSCREENSHOT
        Thread.sleep(5000);
        TakesScreenshot ts = (TakesScreenshot) driver;
        Thread.sleep(5000);
        File source = ts.getScreenshotAs(OutputType.FILE);
        File destination = new File("D:\\ecl\\javatask\\SeleniumTask\\screenshot\\Nyka.png");
        FileUtils.copyFile(source, destination);



    }

}

```

```java
package com.ProjectClassTask;

import java.awt.AWTException;

public class BaseClass {
    public static WebDriver driver;
    // WEBELEMENT METHODS

    public static void userInput(WebElement element, String value) {
        element.sendKeys(value);
    }

    public static void clickTheElement(WebElement element) {
        element.click();
    }

    public static void booleanOptions(String name, WebElement element) {

        if (name.equalsIgnoreCase("display")) {
            boolean displayed = element.isDisplayed();
            System.out.println(displayed);
        } else if (name.equalsIgnoreCase("enable")) {
            boolean enabled = element.isEnabled();
            System.out.println(enabled);
        } else if (name.equalsIgnoreCase("select")) {
            boolean selected = element.isSelected();
            System.out.println(selected);
        } else {
            System.out.println("Invalid Element");
        }
    }

    public static void textOntheElement(WebElement element) {
        String text = element.getText();
        System.out.println(text);
    }

    public static void attributeOfTheElement(WebElement element, String value) {
        String attribute = element.getAttribute(value);
        System.out.println(attribute);
    }

    public static void clearOfTheElement(WebElement element) {
        element.clear();
    }

    public static void singleDropDown(String name, WebElement element, String value) {

        Select s = new Select(element);
        if (name.equalsIgnoreCase("text")) {
            s.selectByVisibleText(value);
        } else if (name.equalsIgnoreCase("value")) {
            s.selectByValue(value);
        } else if (name.equalsIgnoreCase("index")) {
            int val = Integer.parseInt(value); // string convert into integer
            s.selectByIndex(val);
        } else if (name.equalsIgnoreCase("deSelectValue")) {
            s.deselectByValue(value);
        } else if (name.equalsIgnoreCase("deSelectIndex")) {
            int vale = Integer.parseInt(value);
            s.deselectByIndex(vale);
        } else if (name.equalsIgnoreCase("deSelectText")) {
            s.deselectByVisibleText(value);
        } else if (name.equalsIgnoreCase("deSelectAll")) {
            s.deselectAll();
        } else {
            System.out.println("Invalid Selection");
        }

    }

    public static void mouseActions(WebElement element, String name) {

        Actions a = new Actions(driver);

        if (name.equalsIgnoreCase("click")) {
            a.click(element).perform();
            ;
        } else if (name.equalsIgnoreCase("doubleClick")) {
            a.doubleClick(element).perform();
            ;
        } else if (name.equalsIgnoreCase("rightClick")) {
            a.contextClick(element).perform();
            ;
        } else if (name.equalsIgnoreCase("move")) {
            a.moveToElement(element).perform();
        } else {
            System.out.println("invalid action");
        }
    }

    public static void dragAndDrop(String name, WebElement source, WebElement desti) {
        Actions aa = new Actions(driver);
        if (name.equalsIgnoreCase("dragAndDrop")) {
            aa.dragAndDrop(source, desti).perform();
        } else if (name.equalsIgnoreCase("clickAndHold")) {
            aa.clickAndHold(source).moveToElement(desti).release(desti).perform();
        }
    }

    public static void multipleDropDownOptions(WebElement element, String name) {

        Select ss = new Select(element);

        boolean multiple = ss.isMultiple();
        System.out.println(multiple);

        if (name.equalsIgnoreCase("optios")) {
            List<WebElement> options = ss.getOptions();

            for (int i = 0; i < options.size(); i++) {
                String text = options.get(i).getText();
                System.out.println(text);
            }
        } else if (name.equalsIgnoreCase("allOptions")) {
            List<WebElement> allSelectedOptions = ss.getAllSelectedOptions();

            for (WebElement allOptionText : allSelectedOptions) {

                String text1 = allOptionText.getText();
                System.out.println(text1);
            }
        } else if (name.equalsIgnoreCase("firstSelected")) {
            WebElement firstSelectedOption = ss.getFirstSelectedOption();
            System.out.println(firstSelectedOption.getText());
        }
    }

    // WEBDRIVER METHOD

    public static WebDriver browserLaunch(String browser) {

        if (browser.equalsIgnoreCase("chrome")) {
            System.setProperty("webdriver.chrome.driver",
                    "D:\\ecl\\javatask\\ProjectClassTask\\driver\\chromedriver.exe");
            driver = new ChromeDriver();
        } else if (browser.equalsIgnoreCase("firefox")) {
            System.setProperty("webdriver.gecko.driver",
                    ".\\driver\\geckodriver.exe");
            driver = new FirefoxDriver();
        } else if (browser.equalsIgnoreCase("edge")) {
            String property = System.getProperty("user.dir");
            System.out.println(property);
            System.setProperty("webdriver.edge.driver",""+property +"\\driver\
\msedgedriver.exe");
            driver = new EdgeDriver();
        } else {
            System.out.println("ivalid browser");
        }
        driver.manage().window().maximize();
        return driver;
    }

    public static void launchURL(String option, String url) {

        if (option.equalsIgnoreCase("getURL")) {
            driver.get(url);
        }

        else if (option.equalsIgnoreCase("navigateURL")) {
            driver.navigate().to(url);
        } else {
            System.out.println("invalid url");
        }

    }

    public static void textPage(String option) {
        if (option.equalsIgnoreCase("getTitle")) {
            String title = driver.getTitle();
            System.out.println(title);
        } else if (option.equalsIgnoreCase("getTitle")) {
            String urlText = driver.getCurrentUrl();
            System.out.println(urlText);
        }

    }

    public static void navigatePage(String option) {
        if (option.equalsIgnoreCase("forward")) {
            driver.navigate().forward();
        } else if (option.equalsIgnoreCase("back")) {
            driver.navigate().back();
        } else if (option.equalsIgnoreCase("refresh")) {
            driver.navigate().refresh();
        }
    }

    public static void sleepPage(int sec) throws InterruptedException {
        Thread.sleep(sec);
    }

    public static void implicitThePage(int sec) {

        driver.manage().timeouts().implicitlyWait(sec, TimeUnit.SECONDS);
    }

    public static void explicitWaitOfThePage(int sec, String option, WebElement element) {

        WebDriverWait wait = new WebDriverWait(driver, 20);

        if (option.equalsIgnoreCase("visible")) {
            wait.until(ExpectedConditions.visibilityOf(element));
        } else if (option.equalsIgnoreCase("clickable")) {
            wait.until(ExpectedConditions.elementToBeClickable(element));

        }
    }

    public static void takesScreen(String fileName) throws IOException {

        TakesScreenshot ts = (TakesScreenshot) driver;

        File source = ts.getScreenshotAs(OutputType.FILE);

        File desti = new File("D:\\ecl\\javatask\\SeleniumTask\\screenshot\\" + fileName + ".png");
        // FileUtils.copyFile(source, desti);
        FileHandler.copy(source, desti);
    }

    public static void keyBord(String option) throws AWTException {

        Robot r = new Robot();

        if (option.equalsIgnoreCase("clickControl")) {
            r.keyPress(KeyEvent.VK_CONTROL);
        } else if (option.equalsIgnoreCase("releaseControl")) {
            r.keyRelease(KeyEvent.VK_CONTROL);
        } else if (option.equalsIgnoreCase("clickEnter")) {
            r.keyPress(KeyEvent.VK_ENTER);
        } else if (option.equalsIgnoreCase("releaseEnter")) {
            r.keyRelease(KeyEvent.VK_ENTER);
        } else if (option.equalsIgnoreCase("clickA")) {
            r.keyPress(KeyEvent.VK_A);
        } else if (option.equalsIgnoreCase("releaseA")) {
            r.keyRelease(KeyEvent.VK_A);
        }
    }

    public static void alertPage(String option) {

        Alert alert = driver.switchTo().alert();

        if (option.equalsIgnoreCase("accept")) {
            alert.accept();
        }

        else if (option.equalsIgnoreCase("dismiss")) {
            alert.dismiss();
        } else if (option.equalsIgnoreCase("text")) {

            String text = alert.getText();
            System.out.println();
        }

    }

    public static void alertSendkey(String value) {

        Alert alert1 = driver.switchTo().alert();

        alert1.sendKeys(value);
        alert1.accept();
    }

    public static void javaScriptScroll(WebElement element, String option) {

        try {
            JavascriptExecutor js = (JavascriptExecutor) driver;

            if (option.equalsIgnoreCase("scroll")) {
                js.executeScript("arguements[0].scrollIntoView();", element);
            }

            else if (option.equals("click")) {
                js.executeScript("arguments[0].click();", element);
            }

        } catch (Exception e) {
            e.printStackTrace();
            System.out.println(e);
        }
    }

    public static void jsScrollPixel(String xDirection, String yDirection) {
        try {
            JavascriptExecutor js1 = (JavascriptExecutor) driver;
            js1.executeScript("window.scrollBy(" + xDirection + "," + yDirection + ")");
        }

        catch (Exception e) {
            e.printStackTrace();
            System.out.println(e);
        }
    }

    public static void scrollBottomPage(String option) {

        try {
            JavascriptExecutor js2 = (JavascriptExecutor) driver;
            if (option.equalsIgnoreCase("scrollDown")) {
                js2.executeScript("window.scrollBy(0,document.body.ScrollHeight);");// entire page scroll down
            } else if (option.equalsIgnoreCase("scrollUp")) {
                js2.executeScript("window.scrollBy(0,-document.body.ScrollHeight);"); // entire page scroll up
            }
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println(e);
        }
    }

    public static String singleWindowHandle() {

        String windowHandle = driver.getWindowHandle();

        System.out.println(windowHandle);

        return windowHandle;

    }

    public static void multipleWindowHandle(String firstWindowId) {

        Set<String> windowHandles = driver.getWindowHandles();

        System.out.println(windowHandles);

        System.out.println(windowHandles.size());

        for (String allID : windowHandles) {

            if (!(allID == firstWindowId)) {
                driver.switchTo().window(allID);
                break;
            }
        }
    }

    public static void frameSize(String value) {

        List<WebElement> findElements = driver.findElements(By.tagName(value));

        System.out.println(findElements.size());

    }

    public static void frameSelection(String option,String value) {

        if (option.equalsIgnoreCase("index")) {
            int indexVal = Integer.parseInt(value);
            driver.switchTo().frame(indexVal);
        }
        else if (option.equalsIgnoreCase("id")) {
            driver.switchTo().frame(value);

        }
    }
    public static void frameSelectionElement(WebElement element) {

            driver.switchTo().frame(element);
        }
    public static void frameParentDefault(String option,String value) {

        if (option.equalsIgnoreCase("parent")) {
            driver.switchTo().parentFrame();
        }
        else if (option.equalsIgnoreCase("defaultContenet")) {
            driver.switchTo().defaultContent();
        }
    }

    public static void multiWindow2() {
        Set<String> windowHandles = driver.getWindowHandles();
        Iterator iterator = windowHandles.iterator();
        String firstWin = (String) iterator.next();
        String secondWin = (String) iterator.next();
        driver.switchTo().window(secondWin);
    }

}
```

```java
package com.pom;

import org.openqa.selenium.WebDriver;

public class SignupPage {

public static WebDriver driver;
//  WebElement findElement = driver.findElement(By.xpath("//button[text()='Sign in']"));
//  WebElement findElement2 = driver.findElement(By.xpath("//button[contains(text(),'Sign in with Mobile / Email')]"));
//  WebElement findElement3 = driver.findElement(By.xpath("//input[@name='emailMobile']"));
//  WebElement findElement4 = driver.findElement(By.xpath("//button[@type='submit']"));


    public SignupPage(WebDriver driver1) {
        this.driver=driver1;
        PageFactory.initElements(driver, this);
    }

    public WebElement getSignUpClick() {
        return signUpClick;
    }

    public WebElement getElementClick() {
        return elementClick;
    }

    public WebElement getInput() {
        return input;
    }

    public WebElement getClickSignUp() {
        return clickSignUp;
    }

    @FindBy(xpath = "//button[text()='Sign in']")
    private WebElement signUpClick;

    @FindBy(xpath ="//button[contains(text(),'Sign in with Mobile / Email')]")
    private WebElement elementClick;

    @FindBy(xpath = "//input[@name='emailMobile']")
    private WebElement input;

    @FindBy(xpath = "//button[@type='submit']")
    private WebElement clickSignUp;




}

```

```java
package com.pom;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class OtpPage {

    public static WebDriver driver;
//      WebElement findElement3 = driver.findElement(By.xpath("//input[@name='emailMobile']"));
//      findElement3.sendKeys("123456789");

//      WebElement findElement4 = driver.findElement(By.xpath("//button[@type='submit']"));
//      findElement4.click();

//  WebElement findElement5 = driver.findElement(By.name("otpValue"));
//  findElement5.sendKeys(otp);
//  WebElement findElement6 = driver.findElement(By.xpath("//button[contains(@class,'button big fill full')]"));
//  findElement6.click();

    public OtpPage(WebDriver driver2) {
        this.driver=driver2;
        PageFactory.initElements(driver, this);
    }


    public WebElement getMobileNumberEntry() {
        return mobileNumberEntry;
    }

    public WebElement getMobileNumberClick() {
        return mobileNumberClick;
    }

    public WebElement getOtpEntry() {
        return otpEntry;
    }

    public WebElement getOtpClik() {
        return otpClik;
    }

    @FindBy(xpath = "//input[@name='emailMobile']")
    private WebElement mobileNumberEntry;

    @FindBy(xpath = "//button[@type='submit']")
    private WebElement mobileNumberClick;

    @FindBy(name = "otpValue")
    private WebElement otpEntry;

    @FindBy(xpath = "//button[contains(@class,'button big fill full')]")
    private WebElement otpClik;

}

```

```java
package com.pom;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class ProductSearch {
public static WebDriver driver;
//  WebElement findElement7 = driver.findElement(By.xpath("//div[@id='my-menu']//a[contains(text(),'make')]"));
//
//  WebElement findElement8 = driver.findElement(By.xpath("//div[@id='my-menu']//a[contains(text(),'Face Primer')]//following::li[1]//a"));
//  findElement8.click();

    public ProductSearch(WebDriver driver3) {

        this.driver=driver3;
        PageFactory.initElements(driver, this);
    }

    @FindBy(xpath="//div[@id='my-menu']//a[contains(text(),'make')]")
    private WebElement moveAction;

    @FindBy(xpath="//div[@id='my-menu']//a[contains(text(),'Face Primer')]//following::li[1]//a")
    private WebElement actionClick;




    public WebElement getMoveAction() {
        return moveAction;
    }

    public WebElement getActionClick() {
        return actionClick;
    }




}
```

```java
package com.pom;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;

public class AddToBag {

    public static WebDriver driver;

//  WebElement findElement9 = driver
//          .findElement(By.xpath("//*[@id=\"product-list-wrap\"]/div[1]/div/div[2]/div/button[2]"));

//  WebElement findElement10 = driver
//          .findElement(By.xpath("//*[@id=\"product-list-wrap\"]/div[1]/div/div[2]/div/button/span"));
//  findElement10.click();

//  WebElement findElement11 = driver
//          .findElement(By.xpath("//*[@id=\"header_id\"]/div[2]/div/div[2]/div[2]/button"));
//  findElement11.click();

    public AddToBag(WebDriver driver4) {

        this.driver=driver4;

        PageFactory.initElements(driver, this);
    }


    @FindBy(xpath="//*[@id='product-list-wrap']/div[1]/div/div[2]/div/button[2]")
private WebElement jsClick1;

    @FindBy(xpath="//*[@id='product-list-wrap']/div[1]/div/div[2]/div/button/span")
    private WebElement addClick2;

    @FindBy(xpath="//*[@id='header_id']/div[2]/div/div[2]/div[2]/button")
    private WebElement svgClick3;



    public WebElement getJsClick1() {
        return jsClick1;
    }

    public WebElement getAddClick2() {
        return addClick2;
    }

    public WebElement getSvgClick3() {
        return svgClick3;
    }

}
```

```java
package com.pom;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.PageFactory;

public class POMmanager {

    public static WebDriver driver;

    public POMmanager(WebDriver driver5) {
        this.driver=driver5;
        PageFactory.initElements(driver, this);
    }


    public SignupPage signUp () {

        SignupPage sp= new SignupPage(driver);

        return sp;
    }

    public OtpPage otp() {

        OtpPage op= new OtpPage (driver);

        return op;
    }

    public ProductSearch prSearcrh() {

        ProductSearch ps= new ProductSearch (driver);

        return ps;
    }

    public AddToBag addBag() {

        AddToBag ab= new AddToBag (driver);

        return ab;
    }


}

```

```java
package com.propertyfile;

import java.io.IOException;

public class ConfigurationHelper {

private ConfigurationHelper() {
}

public static ConfigurationReader getInstance() throws IOException {

    ConfigurationReader cr=new ConfigurationReader();
    return cr;
}

}

```

```java
package com.propertyfile;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Properties;

public class ConfigurationReader {

    Properties p;
    public ConfigurationReader() throws IOException {

        File f= new File(".\\src\\main\\java\\com\\propertyfile\\nyka.properties");

        FileInputStream fis= new FileInputStream(f);

         p= new Properties();

        p.load(fis);
    }

    public  String getBrowser() {

String browser = p.getProperty("browser");
return browser;
    }

    public  String getUrl() {

        String url = p.getProperty("url");
        return url;
            }

public  String getMobileNumber() {

        String mobileNumer = p.getProperty("mobileNumber");
        return mobileNumer;
            }

public  String getScreenr() {

    String screenshot = p.getProperty("screenshot");
    return screenshot;
        }
public  String getJsX() {

    String jsX = p.getProperty("jsX");
    return jsX;
        }

public  String getJsY() {

    String jsY = p.getProperty("jsY");
    return jsY;
        }
}
```

```
Feature: Nyka Application Testing

#eature: Adactin Application Testing
#Scenario: Login Page
#
#Given user Launch The Url
#When user Enter The Username In Username Field
#And user Enter the Password In Password Field
#Then user Click Login Button



Scenario: Signup Page
Given launch Url
When wait For Few Minutes
When move To SignIn Button
Then click Sigin button Navigate the Otp Page

Scenario: Otp page
When user Enter The Mobile Number
And user Click the Submit Button
And scanner Using To Put The Otp
And user Enter The Otp In Otp Field
Then user Click The Verification button

Scenario: Product Search
When move to The Make
Then click The Concealer Navigate Next Window

Scenario: Add To Bag
When switch The Window
And scroll Down The Particular Element
And click The Preview Shades
And click The Add To Bag
And move To Svg
And take Snap
```

```java
package com.testrunner;

import java.io.IOException;


@RunWith(Cucumber.class)
@CucumberOptions(features = "src\\test\\java\\com\\cuc\\nyka.feature",glue="com.stepdefinition",monochrome = true,
plugin = {"pretty","html:Report1/nykahtmlreport.html","json:Report1/nykajsonreport.json","junit:Report1/nykaxmlreport.xml"})

//"com.cucumber.listener.ExtentCucumberFormatter:Report/extentreport.html"  //not working io cucumber
public class RunnerTest {
    public static WebDriver driver;

    @BeforeClass
    public static void setUp() throws IOException {

        String browser = ConfigurationHelper.getInstance().getBrowser();

        driver=Base2.browserLaunch(browser);
        //driver=Base2.browserLaunch("chrome");

//      System.setProperty("webdriver.chrome.driver", ".\\driverss\\chromedriver.exe");
//      driver = new ChromeDriver();
    }

    @AfterClass
    public static void tearDown() {
        driver.quit();
    }
}

```

```java
package com.stepdefinition;

import java.io.IOException;


public class StepDefinition extends Base2 {

    POMmanager pm= new POMmanager(driver);

//  public WebDriver driver = RunnerTest.driver;
//
//  @Given("^user Launch The Url$")
//  public void user_Launch_The_Url() throws Throwable {
//
//      driver.get("https://adactinhotelapp.com/");
//  }
//
//  @When("^user Enter The Username In Username Field$")
//  public void user_Enter_The_Username_In_Username_Field() throws Throwable {
//      WebElement findElement = driver.findElement(By.id("username"));
//      findElement.sendKeys("absbd");
//
//  }
//
//  @When("^user Enter the Password In Password Field$")
//  public void user_Enter_the_Password_In_Password_Field() throws Throwable {
//      WebElement findElement = driver.findElement(By.id("password"));
//      findElement.sendKeys("12344");
//  }
//
//  @Then("^user Click Login Button$")
//  public void user_Click_Login_Button() throws Throwable {
//
//      WebElement findElement = driver.findElement(By.id("login"));
//      findElement.click();
//  }


// Signup Page
    @Given("launch Url")
    public void launch_url() throws IOException {

        String url = ConfigurationHelper.getInstance().getUrl();
        launchURL("getURL", url);

        //launchURL("getURL", "https://www.nykaa.com/");
    }

    @When("wait For Few Minutes")
    public void wait_for_few_minutes() {
        implicitThePage(10);
    }
    @When("move To SignIn Button")
    public void move_to_sign_in_button() {
        clickTheElement(pm.signUp().getSignUpClick());
    }
    @Then("click Sigin button Navigate the Otp Page")
    public void click_sigin_button_navigate_the_otp_page() {
        clickTheElement(pm.signUp().getElementClick());
    }

//Otp Page
    @When("user Enter The Mobile Number")
    public void user_enter_the_mobile_number() throws IOException {

        String mobileNumber = ConfigurationHelper.getInstance().getMobileNumber();
        userInput(pm.otp().getMobileNumberEntry(), mobileNumber);

        //userInput(pm.otp().getMobileNumberEntry(), "123456789");
    }
    @When("user Click the Submit Button")
    public void user_click_the_submit_button() {

    }
    @When("scanner Using To Put The Otp")
    public void scanner_using_to_put_the_otp() {
        clickTheElement(pm.otp().getMobileNumberClick());
    }
    @When("user Enter The Otp In Otp Field")
    public void user_enter_the_otp_in_otp_field() {
        Scanner sc= new Scanner(System.in);
        System.out.println("OTP");
        String otp = sc.nextLine();
        userInput(pm.otp().getOtpEntry(), otp);
    }
    @Then("user Click The Verification button")
    public void user_click_the_verification_button() {
        clickTheElement(pm.otp().getOtpClik());
    }
//Product Search  Page
    @When("move to The Make")
    public void move_to_the_make() {
        mouseActions(pm.prSearcrh().getMoveAction(),"move");
    }
    @Then("click The Concealer Navigate Next Window")
    public void click_the_concealer_navigate_next_window() {
        clickTheElement(pm.prSearcrh().getActionClick());
    }

//Add to Bag
    @When("switch The Window")
    public void switch_the_window() {
        multiWindow2();
    }
    @When("scroll Down The Particular Element")
    public void scroll_down_the_particular_element() throws IOException {
        String jsX = ConfigurationHelper.getInstance().getJsX();
        String jsY = ConfigurationHelper.getInstance().getJsY();
        jsScrollPixel(jsX,jsY);

        //jsScrollPixel("0","600");
    }
    @When("click The Preview Shades")
    public void click_the_preview_shades() {
        //try {
            javaScriptScroll(pm.addBag().getJsClick1(), "click");
        //} catch (Exception e) {
        //}
    }
    @When("click The Add To Bag")
    public void click_the_add_to_bag() {
        clickTheElement(pm.addBag().getAddClick2());
    }
    @When("move To Svg")
    public void move_to_svg() {
        clickTheElement(pm.addBag().getSvgClick3());
    }
    @When("take Snap")
    public void take_snap() throws IOException {
        String screenr = ConfigurationHelper.getInstance().getScreenr();
        takesScreen(screenr);

        //takesScreen("nykaScreen");
    }


}
```