

```
1 package com.ProjectClassTask;
2
3 import java.io.File;
19
20 public class NykaTask {
21
22     public static void main(String[] args) throws InterruptedException, IOException {
23
24         System.setProperty("webdriver.chrome.driver", ".\\driver\\chromedriver.exe");
25         WebDriver driver = new ChromeDriver();
26         driver.get("https://www.nykaa.com/");
27         driver.manage().window().maximize();
28         driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
29
30         WebElement findElement = driver.findElement(By.xpath("//button[text()='Sign in']"));
31         findElement.click();
32
33
34         WebElement findElement2 = driver.findElement(By.xpath("//button[contains(text(),'Sign
in with Mobile / Email')]"));
35         findElement2.click();
36
37         Thread.sleep(2000);
38
39         WebElement findElement3 = driver.findElement(By.xpath("//input
[@name='emailMobile']"));
40         findElement3.sendKeys("9876543210");
41
42         WebElement findElement4 = driver.findElement(By.xpath("//button[@type='submit']"));
43         findElement4.click();
44         //SCANNER
45         Scanner sc = new Scanner(System.in);
46         System.out.println("OTP");
47         String otp = sc.nextLine();
48
49         WebElement findElement5 = driver.findElement(By.name("otpValue"));
50         findElement5.sendKeys(otp);
51
52         WebElement findElement6 = driver.findElement(By.xpath("//button[contains
(@class,'button big fill full')]"));
53         findElement6.click();
54
55         WebElement findElement7 = driver.findElement(By.xpath("//div[@id='my-menu']/a
[contains(text(),'make')]"));
56         //MOUSE ACTIONS
57         Actions a = new Actions(driver);
58         a.moveToElement(findElement7).build().perform();
59         WebElement findElement8 = driver.findElement(By.xpath("//div[@id='my-menu']/a
[contains(text(),'Face Primer')]/following::li[1]/a"));
60         findElement8.click();
61
62         Thread.sleep(5000);
63
64         //WINDOWS HANDLING
65         Set<String> windowHandles = driver.getWindowHandles();
66         Iterator iterator = windowHandles.iterator();
67         String firstWin = (String) iterator.next();
```

```
68     String secondWin = (String) iterator.next();
69     driver.switchTo().window(secondWin);
70
71     Thread.sleep(5000);
72
73     //JAVA SCRIPT EXECUTOR
74     JavascriptExecutor js = (JavascriptExecutor) driver;
75     js.executeScript("window.scroll(0,600);", ""); // scrollldown
76
77     //JAVASCRIPT -SCRIPT TIMEOUT EXCEPTION HANDLING
78     try {
79         WebElement findElement9 = driver
80             .findElement(By.xpath("//*[@id=\"product-list-wrap\"]/div[1]/div/div
113 [2]/div/button[2]"));
81
82         js.executeAsyncScript("arguments[0].click();", findElement9);
83
84     } catch (Exception e) {
85         // script timeout exception
86     }
87     System.out.println("preview shade");
88
89     //ADD TO BAG CLICK
90     WebElement findElement10 = driver
91         .findElement(By.xpath("//*[@id=\"product-list-wrap\"]/div[1]/div/div
113 [2]/div/button/span"));
92     findElement10.click();
93     System.out.println("add to bag");
94
95     //SVG --BUTTON CLICK
96     WebElement findElement11 = driver
97         .findElement(By.xpath("//*[@id=\"header_id\"]/div[2]/div/div[2]/div
113 [2]/button"));
98     findElement11.click();
99     System.out.println("moved");
100
101     //TAKESCREENSHOT
102     Thread.sleep(5000);
103     TakesScreenshot ts = (TakesScreenshot) driver;
104     Thread.sleep(5000);
105     File source = ts.getScreenshotAs(OutputType.FILE);
106     File destination = new File("D:\\ec1\\javatask\\SeleniumTask\\screenshot\\Nyka.png");
107     FileUtils.copyFile(source, destination);
108
109
110
111 }
112
113 }
114
```

```
1 package com.ProjectClassTask;
2
3 import java.awt.AWTException;
32
33 public class BaseClass {
34     public static WebDriver driver;
35     // WEBELEMENT METHODS
36
37     public static void userInput(WebElement element, String value) {
38         element.sendKeys(value);
39     }
40
41     public static void clickTheElement(WebElement element) {
42         element.click();
43     }
44
45     public static void booleanOptions(String name, WebElement element) {
46
47         if (name.equalsIgnoreCase("display")) {
48             boolean displayed = element.isDisplayed();
49             System.out.println(displayed);
50         } else if (name.equalsIgnoreCase("enable")) {
51             boolean enabled = element.isEnabled();
52             System.out.println(enabled);
53         } else if (name.equalsIgnoreCase("select")) {
54             boolean selected = element.isSelected();
55             System.out.println(selected);
56         } else {
57             System.out.println("Invalid Element");
58         }
59     }
60
61     public static void textOnTheElement(WebElement element) {
62         String text = element.getText();
63         System.out.println(text);
64     }
65
66     public static void attributeOfTheElement(WebElement element, String value) {
67         String attribute = element.getAttribute(value);
68         System.out.println(attribute);
69     }
70
71     public static void clearOfTheElement(WebElement element) {
72         element.clear();
73     }
74
75     public static void singleDropDown(String name, WebElement element, String value) {
76
77         Select s = new Select(element);
78         if (name.equalsIgnoreCase("text")) {
79             s.selectByVisibleText(value);
80         } else if (name.equalsIgnoreCase("value")) {
81             s.selectByValue(value);
82         } else if (name.equalsIgnoreCase("index")) {
83             int val = Integer.parseInt(value); // string convert into integer
84             s.selectByIndex(val);
85         } else if (name.equalsIgnoreCase("deSelectValue")) {
```

```
86         s.deselectByValue(value);
87     } else if (name.equalsIgnoreCase("deSelectIndex")) {
88         int vale = Integer.parseInt(value);
89         s.deselectByIndex(vale);
90     } else if (name.equalsIgnoreCase("deSelectText")) {
91         s.deselectByVisibleText(value);
92     } else if (name.equalsIgnoreCase("deSelectAll")) {
93         s.deselectAll();
94     } else {
95         System.out.println("Invalid Selection");
96     }
97 }
98 }
99
100 public static void mouseActions(WebElement element, String name) {
101     Actions a = new Actions(driver);
102
103     if (name.equalsIgnoreCase("click")) {
104         a.click(element).perform();
105     }
106     ;
107     } else if (name.equalsIgnoreCase("doubleClick")) {
108         a.doubleClick(element).perform();
109     }
110     ;
111     } else if (name.equalsIgnoreCase("rightClick")) {
112         a.contextClick(element).perform();
113     }
114     ;
115     } else if (name.equalsIgnoreCase("move")) {
116         a.moveToElement(element).perform();
117     }
118     } else {
119         System.out.println("invalid action");
120     }
121 }
122
123 public static void dragAndDrop(String name, WebElement source, WebElement desti) {
124     Actions aa = new Actions(driver);
125     if (name.equalsIgnoreCase("dragAndDrop")) {
126         aa.dragAndDrop(source, desti).perform();
127     }
128     } else if (name.equalsIgnoreCase("clickAndHold")) {
129         aa.clickAndHold(source).moveToElement(desti).release(desti).perform();
130     }
131 }
132
133 public static void multipleDropDownOptions(WebElement element, String name) {
134     Select ss = new Select(element);
135
136     boolean multiple = ss.isMultiple();
137     System.out.println(multiple);
138
139     if (name.equalsIgnoreCase("optios")) {
140         List<WebElement> options = ss.getOptions();
141
142         for (int i = 0; i < options.size(); i++) {
143             String text = options.get(i).getText();
144             System.out.println(text);
145         }
146     }
```

```
143     } else if (name.equalsIgnoreCase("allOptions")) {
144         List<WebElement> allSelectedOptions = ss.getAllSelectedOptions();
145
146         for (WebElement allOptionText : allSelectedOptions) {
147
148             String text1 = allOptionText.getText();
149             System.out.println(text1);
150         }
151     } else if (name.equalsIgnoreCase("firstSelected")) {
152         WebElement firstSelectedOption = ss.getFirstSelectedOption();
153         System.out.println(firstSelectedOption.getText());
154     }
155 }
156
157 // WEBDRIVER METHOD
158
159 public static WebDriver browserLaunch(String browser) {
160
161     if (browser.equalsIgnoreCase("chrome")) {
162         System.setProperty("webdriver.chrome.driver",
163             "D:\\ec1\\javatask\\ProjectClassTask\\driver\\chromedriver.exe");
164         driver = new ChromeDriver();
165     } else if (browser.equalsIgnoreCase("firefox")) {
166         System.setProperty("webdriver.gecko.driver",
167             ".\\driver\\geckodriver.exe");
168         driver = new FirefoxDriver();
169     } else if (browser.equalsIgnoreCase("edge")) {
170         String property = System.getProperty("user.dir");
171         System.out.println(property);
172         System.setProperty("webdriver.edge.driver", ""+property +"\\driver\\
173         \msedgedriver.exe");
174         driver = new EdgeDriver();
175     } else {
176         System.out.println("invalid browser");
177     }
178     driver.manage().window().maximize();
179     return driver;
180 }
181
182 public static void launchURL(String option, String url) {
183
184     if (option.equalsIgnoreCase("getURL")) {
185         driver.get(url);
186     }
187
188     else if (option.equalsIgnoreCase("navigateURL")) {
189         driver.navigate().to(url);
190     } else {
191         System.out.println("invalid url");
192     }
193 }
194
195 public static void textPage(String option) {
196     if (option.equalsIgnoreCase("getTitle")) {
197         String title = driver.getTitle();
198         System.out.println(title);
199     }
```

```
199         } else if (option.equalsIgnoreCase("getTitle")) {
200             String urlText = driver.getCurrentUrl();
201             System.out.println(urlText);
202         }
203     }
204 }
205
206 public static void navigatePage(String option) {
207     if (option.equalsIgnoreCase("forward")) {
208         driver.navigate().forward();
209     } else if (option.equalsIgnoreCase("back")) {
210         driver.navigate().back();
211     } else if (option.equalsIgnoreCase("refresh")) {
212         driver.navigate().refresh();
213     }
214 }
215
216 public static void sleepPage(int sec) throws InterruptedException {
217     Thread.sleep(sec);
218 }
219
220 public static void implicitThePage(int sec) {
221     driver.manage().timeouts().implicitlyWait(sec, TimeUnit.SECONDS);
222 }
223
224 public static void explicitWaitOfThePage(int sec, String option, WebElement element) {
225     WebDriverWait wait = new WebDriverWait(driver, 20);
226
227     if (option.equalsIgnoreCase("visible")) {
228         wait.until(ExpectedConditions.visibilityOf(element));
229     } else if (option.equalsIgnoreCase("clickable")) {
230         wait.until(ExpectedConditions.elementToBeClickable(element));
231     }
232 }
233
234 }
235
236 public static void takesScreen(String fileName) throws IOException {
237     TakesScreenshot ts = (TakesScreenshot) driver;
238
239     File source = ts.getScreenshotAs(OutputType.FILE);
240
241     File desti = new File("D:\\ecl\\javatask\\SeleniumTask\\screenshot\\" + fileName +
242         ".png");
243     // FileUtils.copyFile(source, desti);
244     FileHandler.copy(source, desti);
245 }
246
247 public static void keyBord(String option) throws AWTException {
248     Robot r = new Robot();
249
250     if (option.equalsIgnoreCase("clickControl")) {
251         r.keyPress(KeyEvent.VK_CONTROL);
252     } else if (option.equalsIgnoreCase("releaseControl")) {
253         r.keyRelease(KeyEvent.VK_CONTROL);
254     }
```

```
255         r.keyRelease(KeyEvent.VK_CONTROL);
256     } else if (option.equalsIgnoreCase("clickEnter")) {
257         r.keyPress(KeyEvent.VK_ENTER);
258     } else if (option.equalsIgnoreCase("releaseEnter")) {
259         r.keyRelease(KeyEvent.VK_ENTER);
260     } else if (option.equalsIgnoreCase("clickA")) {
261         r.keyPress(KeyEvent.VK_A);
262     } else if (option.equalsIgnoreCase("releaseA")) {
263         r.keyRelease(KeyEvent.VK_A);
264     }
265 }
266
267 public static void alertPage(String option) {
268     Alert alert = driver.switchTo().alert();
269
270     if (option.equalsIgnoreCase("accept")) {
271         alert.accept();
272     }
273
274     else if (option.equalsIgnoreCase("dismiss")) {
275         alert.dismiss();
276     } else if (option.equalsIgnoreCase("text")) {
277         String text = alert.getText();
278         System.out.println();
279     }
280 }
281
282
283
284 public static void alertSendkey(String value) {
285     Alert alert1 = driver.switchTo().alert();
286
287     alert1.sendKeys(value);
288     alert1.accept();
289 }
290
291
292 public static void javaScriptScroll(WebElement element, String option) {
293     try {
294         JavascriptExecutor js = (JavascriptExecutor) driver;
295
296         if (option.equalsIgnoreCase("scroll")) {
297             js.executeScript("arguments[0].scrollIntoView();", element);
298         }
299
300         else if (option.equals("click")) {
301             js.executeScript("arguments[0].click();", element);
302         }
303     } catch (Exception e) {
304         e.printStackTrace();
305         System.out.println(e);
306     }
307 }
308
309
310
311
```

```
312     public static void jsScrollPixel(String xDirection, String yDirection) {
313         try {
314             JavascriptExecutor js1 = (JavascriptExecutor) driver;
315             js1.executeScript("window.scrollBy(" + xDirection + "," + yDirection + ")");
316         }
317
318         catch (Exception e) {
319             e.printStackTrace();
320             System.out.println(e);
321         }
322     }
323
324     public static void scrollBottomPage(String option) {
325
326         try {
327             JavascriptExecutor js2 = (JavascriptExecutor) driver;
328             if (option.equalsIgnoreCase("scrollDown")) {
329                 js2.executeScript("window.scrollBy(0,document.body.ScrollHeight);");// entire
330 page scroll down
331             } else if (option.equalsIgnoreCase("scrollUp")) {
332                 js2.executeScript("window.scrollBy(0,-document.body.ScrollHeight);"); //
333 entire page scroll up
334             }
335         } catch (Exception e) {
336             e.printStackTrace();
337             System.out.println(e);
338         }
339     }
340
341     public static String singleWindowHandle() {
342
343         String windowHandle = driver.getWindowHandle();
344
345         System.out.println(windowHandle);
346
347         return windowHandle;
348     }
349
350     public static void multipleWindowHandle(String firstWindowId) {
351
352         Set<String> windowHandles = driver.getWindowHandles();
353
354         System.out.println(windowHandles);
355
356         System.out.println(windowHandles.size());
357
358         for (String allID : windowHandles) {
359
360             if (!(allID == firstWindowId)) {
361                 driver.switchTo().window(allID);
362                 break;
363             }
364         }
365
366     }
367
368     public static void frameSize(String value) {
```



```
367
368     List<WebElement> findElements = driver.findElements(By.tagName(value));
369
370     System.out.println(findElements.size());
371
372 }
373
374 public static void frameSelection(String option,String value) {
375     if (option.equalsIgnoreCase("index")) {
376         int indexVal = Integer.parseInt(value);
377         driver.switchTo().frame(indexVal);
378     }
379     else if (option.equalsIgnoreCase("id")) {
380         driver.switchTo().frame(value);
381     }
382 }
383
384 }
385 public static void frameSelectionElement(WebElement element) {
386     driver.switchTo().frame(element);
387 }
388
389 public static void frameParentDefault(String option,String value) {
390     if (option.equalsIgnoreCase("parent")) {
391         driver.switchTo().parentFrame();
392     }
393     else if (option.equalsIgnoreCase("defaultContent")) {
394         driver.switchTo().defaultContent();
395     }
396 }
397
398 }
399 public static void multiWindow2() {
400     Set<String> windowHandles = driver.getWindowHandles();
401     Iterator iterator = windowHandles.iterator();
402     String firstWin = (String) iterator.next();
403     String secondWin = (String) iterator.next();
404     driver.switchTo().window(secondWin);
405 }
406
407 }
408
409
```

```
1 package com.pom;
2
3 import org.openqa.selenium.WebDriver;
4
5
6
7
8 public class SignupPage {
9
10 public static WebDriver driver;
11 // WebElement findElement = driver.findElement(By.xpath("//button[text()='Sign in']"));
12 // WebElement findElement2 = driver.findElement(By.xpath("//button[contains(text(),'Sign in with Mobile / Email')]"));
13 // WebElement findElement3 = driver.findElement(By.xpath("//input[@name='emailMobile']"));
14 // WebElement findElement4 = driver.findElement(By.xpath("//button[@type='submit']"));
15
16
17 public SignupPage(WebDriver driver1) {
18     this.driver=driver1;
19     PageFactory.initElements(driver, this);
20 }
21
22 public WebElement getSignUpClick() {
23     return signUpClick;
24 }
25
26 public WebElement getElementClick() {
27     return elementClick;
28 }
29
30 public WebElement getInput() {
31     return input;
32 }
33
34 public WebElement getClickSignUp() {
35     return clickSignUp;
36 }
37
38 @FindBy(xpath = "//button[text()='Sign in']")
39 private WebElement signUpClick;
40
41 @FindBy(xpath = "//button[contains(text(),'Sign in with Mobile / Email')]")
42 private WebElement elementClick;
43
44 @FindBy(xpath = "//input[@name='emailMobile']")
45 private WebElement input;
46
47 @FindBy(xpath = "//button[@type='submit']")
48 private WebElement clickSignUp;
49
50
51
52
53 }
54
```

```
1 package com.pom;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.WebElement;
6 import org.openqa.selenium.support.FindBy;
7 import org.openqa.selenium.support.PageFactory;
8
9 public class OtpPage {
10
11     public static WebDriver driver;
12 //     WebElement findElement3 = driver.findElement(By.xpath("//input[@name='emailMobile']"));
13 //     findElement3.sendKeys("123456789");
14
15 //     WebElement findElement4 = driver.findElement(By.xpath("//button[@type='submit']"));
16 //     findElement4.click();
17
18 //     WebElement findElement5 = driver.findElement(By.name("otpValue"));
19 //     findElement5.sendKeys(otp);
20 //     WebElement findElement6 = driver.findElement(By.xpath("//button[contains(@class,'button big fill full')]"));
21 //     findElement6.click();
22
23     public OtpPage(WebDriver driver2) {
24         this.driver=driver2;
25         PageFactory.initElements(driver, this);
26     }
27
28
29     public WebElement getMobileNumberEntry() {
30         return mobileNumberEntry;
31     }
32
33     public WebElement getMobileNumberClick() {
34         return mobileNumberClick;
35     }
36
37     public WebElement getOtpEntry() {
38         return otpEntry;
39     }
40
41     public WebElement getOtpClik() {
42         return otpClik;
43     }
44
45     @FindBy(xpath = "//input[@name='emailMobile']")
46     private WebElement mobileNumberEntry;
47
48     @FindBy(xpath = "//button[@type='submit']")
49     private WebElement mobileNumberClick;
50
51     @FindBy(name = "otpValue")
52     private WebElement otpEntry;
53
54     @FindBy(xpath = "//button[contains(@class,'button big fill full')]")
55     private WebElement otpClik;
56
```

OtpPage.java

Wednesday, March 8, 2023, 12:07 AM

```
57 }  
58
```

```
1 package com.pom;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.WebElement;
6 import org.openqa.selenium.support.FindBy;
7 import org.openqa.selenium.support.PageFactory;
8
9 public class ProductSearch {
10 public static WebDriver driver;
11 // WebElement findElement7 = driver.findElement(By.xpath("//div[@id='my-menu']/a[contains
    (text(),'make')]"));
12 //
13 // WebElement findElement8 = driver.findElement(By.xpath("//div[@id='my-menu']/a[contains
    (text(),'Face Primer')]//following:li[1]//a"));
14 // findElement8.click();
15
16 public ProductSearch(WebDriver driver3) {
17
18     this.driver=driver3;
19     PageFactory.initElements(driver, this);
20 }
21
22 @FindBy(xpath="//div[@id='my-menu']/a[contains(text(),'make')]")
23 private WebElement moveAction;
24
25 @FindBy(xpath="//div[@id='my-menu']/a[contains(text(),'Face Primer')]//following:li
    [1]//a")
26 private WebElement actionClick;
27
28
29
30
31 public WebElement getMoveAction() {
32     return moveAction;
33 }
34
35 public WebElement getActionClick() {
36     return actionClick;
37 }
38
39
40
41
42 }
43
```

```
1 package com.pom;
2
3 import org.openqa.selenium.By;
4 import org.openqa.selenium.WebDriver;
5 import org.openqa.selenium.WebElement;
6 import org.openqa.selenium.support.FindBy;
7 import org.openqa.selenium.support.PageFactory;
8
9 public class AddToBag {
10
11     public static WebDriver driver;
12
13 // WebElement findElement9 = driver
14 //     .findElement(By.xpath("//*[@id=\"product-list-wrap\"]/div[1]/div/div[2]/div/button
15 // [2]"));
16 // WebElement findElement10 = driver
17 //     .findElement(By.xpath("//*[@id=\"product-list-wrap
18 // \"]/div[1]/div/div[2]/div/button/span"));
19 // findElement10.click();
20 // WebElement findElement11 = driver
21 //     .findElement(By.xpath("//*[@id=\"header_id\"]/div[2]/div/div[2]/div[2]/button"));
22 // findElement11.click();
23
24     public AddToBag(WebDriver driver4) {
25
26         this.driver=driver4;
27
28         PageFactory.initElements(driver, this);
29     }
30
31
32     @FindBy(xpath="//*[@id='product-list-wrap']/div[1]/div/div[2]/div/button[2]")
33 private WebElement jsClick1;
34
35     @FindBy(xpath="//*[@id='product-list-wrap']/div[1]/div/div[2]/div/button/span")
36 private WebElement addClick2;
37
38     @FindBy(xpath="//*[@id='header_id']/div[2]/div/div[2]/div[2]/button")
39 private WebElement svgClick3;
40
41
42
43     public WebElement getJsClick1() {
44         return jsClick1;
45     }
46
47     public WebElement getAddClick2() {
48         return addClick2;
49     }
50
51     public WebElement getSvgClick3() {
52         return svgClick3;
53     }
54
55 }
```

AddToBag.java

Tuesday, March 7, 2023, 11:50 PM

56

```
1 package com.pom;
2
3 import org.openqa.selenium.WebDriver;
4 import org.openqa.selenium.support.PageFactory;
5
6 public class POMmanager {
7
8     public static WebDriver driver;
9
10    public POMmanager(WebDriver driver5) {
11        this.driver=driver5;
12        PageFactory.initElements(driver, this);
13    }
14
15
16    public SignupPage signUp () {
17
18        SignupPage sp= new SignupPage(driver);
19
20        return sp;
21    }
22
23    public OtpPage otp() {
24
25        OtpPage op= new OtpPage (driver);
26
27        return op;
28    }
29
30    public ProductSearch prSearchrh() {
31
32        ProductSearch ps= new ProductSearch (driver);
33
34        return ps;
35    }
36
37    public AddToBag addBag() {
38
39        AddToBag ab= new AddToBag (driver);
40
41        return ab;
42    }
43
44
45 }
46
```



```
1 package com.propertyfile;
2
3 import java.io.IOException;
4
5 public class ConfigurationHelper {
6
7     private ConfigurationHelper() {
8     }
9
10    public static ConfigurationReader getInstance() throws IOException {
11
12        ConfigurationReader cr=new ConfigurationReader();
13        return cr;
14    }
15
16 }
17
```

```
1 package com.propertyfile;
2
3 import java.io.File;
4 import java.io.FileInputStream;
5 import java.io.FileNotFoundException;
6 import java.io.IOException;
7 import java.util.Properties;
8
9 public class ConfigurationReader {
10
11     Properties p;
12     public ConfigurationReader() throws IOException {
13
14         File f= new File(".\\src\\main\\java\\com\\propertyfile\\nyka.properties");
15
16         FileInputStream fis= new FileInputStream(f);
17
18         p= new Properties();
19
20         p.load(fis);
21     }
22
23     public String getBrowser() {
24
25 String browser = p.getProperty("browser");
26 return browser;
27     }
28
29     public String getUrl() {
30
31         String url = p.getProperty("url");
32         return url;
33     }
34
35 public String getMobileNumber() {
36
37         String mobileNumber = p.getProperty("mobileNumber");
38         return mobileNumber;
39     }
40
41 public String getScreenr() {
42
43         String screenshot = p.getProperty("screenshot");
44         return screenshot;
45     }
46 public String getJsX() {
47
48         String jsX = p.getProperty("jsX");
49         return jsX;
50     }
51
52 public String getJsY() {
53
54         String jsY = p.getProperty("jsY");
55         return jsY;
56     }
57 }
```



```
1 Feature: Nyka Application Testing
2
3 #eature: Adactin Application Testing
4 #Scenario: Login Page
5 #
6 #Given user Launch The Url
7 #When user Enter The Username In Username Field
8 #And user Enter the Password In Password Field
9 #Then user Click Login Button
10
11
12
13 Scenario: Signup Page
14 Given launch Url
15 When wait For Few Minutes
16 When move To SignIn Button
17 Then click Signin button Navigate the Otp Page
18
19 Scenario: Otp page
20 When user Enter The Mobile Number
21 And user Click the Submit Button
22 And scanner Using To Put The Otp
23 And user Enter The Otp In Otp Field
24 Then user Click The Verification button
25
26 Scenario: Product Search
27 When move to The Make
28 Then click The Concealer Navigate Next Window
29
30 Scenario: Add To Bag
31 When switch The Window
32 And scroll Down The Particular Element
33 And click The Preview Shades
34 And click The Add To Bag
35 And move To Svg
36 And take Snap
37
38
39
40
```

```
1 package com.testrunner;
2
3 import java.io.IOException;
16
17
18 @RunWith(Cucumber.class)
19 @CucumberOptions(features = "src\\test\\java\\com\\cuc\\
    \\nyka.feature", glue = "com.stepdefinition", monochrome = true,
20 plugin =
    {"pretty", "html:Report1/nykahtmlreport.html", "json:Report1/nykajsonreport.json", "junit:Report1/
    nykaxmlreport.xml"})
21
22 // "com.cucumber.listener.ExtentCucumberFormatter:Report/extentreport.html" //not working io
    cucumber
23 public class RunnerTest {
24     public static WebDriver driver;
25
26     @BeforeClass
27     public static void setUp() throws IOException {
28
29         String browser = ConfigurationHelper.getInstance().getBrowser();
30
31         driver = Base2.browserLaunch(browser);
32         // driver = Base2.browserLaunch("chrome");
33
34         // System.setProperty("webdriver.chrome.driver", ".\\driverss\\chromedriver.exe");
35         // driver = new ChromeDriver();
36     }
37
38     @AfterClass
39     public static void tearDown() {
40         driver.quit();
41     }
42 }
43
```

```
1 package com.stepdefinition;
2
3 import java.io.IOException;
18
19
20 public class StepDefinition extends Base2 {
21
22     POMmanager pm= new POMmanager(driver);
23
24 // public WebDriver driver = RunnerTest.driver;
25 //
26 // @Given("^user Launch The Url$")
27 // public void user_Launch_The_Url() throws Throwable {
28 //
29 //     driver.get("https://adactinhotelapp.com/");
30 // }
31 //
32 // @When("^user Enter The Username In Username Field$")
33 // public void user_Enter_The_Username_In_Username_Field() throws Throwable {
34 //     WebElement findElement = driver.findElement(By.id("username"));
35 //     findElement.sendKeys("absbd");
36 //
37 // }
38 //
39 // @When("^user Enter the Password In Password Field$")
40 // public void user_Enter_the_Password_In_Password_Field() throws Throwable {
41 //     WebElement findElement = driver.findElement(By.id("password"));
42 //     findElement.sendKeys("12344");
43 // }
44 //
45 // @Then("^user Click Login Button$")
46 // public void user_Click_Login_Button() throws Throwable {
47 //
48 //     WebElement findElement = driver.findElement(By.id("login"));
49 //     findElement.click();
50 // }
51
52
53 // Signup Page
54 @Given("launch Url")
55 public void launch_url() throws IOException {
56
57     String url = ConfigurationHelper.getInstance().getUrl();
58     LaunchURL("getURL", url);
59
60     //launchURL("getURL", "https://www.nykaa.com/");
61 }
62
63 @When("wait For Few Minutes")
64 public void wait_for_few_minutes() {
65     implicitThePage(10);
66 }
67 @When("move To SignIn Button")
68 public void move_to_sign_in_button() {
69     clickTheElement(pm.signUp().getSignUpClick());
70 }
71 @Then("click Signin button Navigate the Otp Page")
```

```

72     public void click_signin_button_navigate_the_otp_page() {
73         clickTheElement(pm.signUp().getElementClick());
74     }
75
76 //Otp Page
77     @When("user Enter The Mobile Number")
78     public void user_enter_the_mobile_number() throws IOException {
79
80         String mobileNumber = ConfigurationHelper.getInstance().getMobileNumber();
81         userInput(pm.otp().getMobileNumberEntry(), mobileNumber);
82
83         //userInput(pm.otp().getMobileNumberEntry(), "123456789");
84     }
85     @When("user Click the Submit Button")
86     public void user_click_the_submit_button() {
87
88     }
89     @When("scanner Using To Put The Otp")
90     public void scanner_using_to_put_the_otp() {
91         clickTheElement(pm.otp().getMobileNumberClick());
92     }
93     @When("user Enter The Otp In Otp Field")
94     public void user_enter_the_otp_in_otp_field() {
95         Scanner sc= new Scanner(System.in);
96         System.out.println("OTP");
97         String otp = sc.nextLine();
98         userInput(pm.otp().getOtpEntry(), otp);
99     }
100    @Then("user Click The Verification button")
101    public void user_click_the_verification_button() {
102        clickTheElement(pm.otp().getOtpClick());
103    }
104 //Product Search Page
105    @When("move to The Make")
106    public void move_to_the_make() {
107        mouseActions(pm.prSearchrh().getMoveAction(), "move");
108    }
109    @Then("click The Concealer Navigate Next Window")
110    public void click_the_concealer_navigate_next_window() {
111        clickTheElement(pm.prSearchrh().getActionClick());
112    }
113
114 //Add to Bag
115    @When("switch The Window")
116    public void switch_the_window() {
117        multiWindow2();
118    }
119    @When("scroll Down The Particular Element")
120    public void scroll_down_the_particular_element() throws IOException {
121        String jsX = ConfigurationHelper.getInstance().getJsX();
122        String jsY = ConfigurationHelper.getInstance().getJsY();
123        jsScrollPixel(jsX, jsY);
124
125        //jsScrollPixel("0", "600");
126    }
127    @When("click The Preview Shades")
128    public void click_the_preview_shades() {

```

```
129         //try {
130             javascriptScroll(pm.addBag().getJsClick1(), "click");
131         //} catch (Exception e) {
132         //}
133     }
134     @When("click The Add To Bag")
135     public void click_the_add_to_bag() {
136         clickTheElement(pm.addBag().getAddClick2());
137     }
138     @When("move To Svg")
139     public void move_to_svg() {
140         clickTheElement(pm.addBag().getSvgClick3());
141     }
142     @When("take Snap")
143     public void take_snap() throws IOException {
144         String screenr = ConfigurationHelper.getInstance().getScreenr();
145         takesScreen(screenr);
146
147         //takesScreen("nykaScreen");
148     }
149
150
151 }
```