```java
package com.Task;

import java.awt.AWTException;

public class MavenProjectOne {


    public static WebDriver ChromeDrivers;
    //private static int index;
    //private static ByteString option;

//browser launch

    public static WebDriver browserLaunch(String browser) {

        if (browser.equalsIgnoreCase("ChromeDrivers")) {
            System.setProperty("webdriver.chrome.driver", ".\\ChromeDrivers\
\chromedriver.exe");
            ChromeDrivers = new ChromeDriver();

        }
        else if (browser.equalsIgnoreCase("FireFox")) {

            System.setProperty("webdriver.gecko.driver", "null value");
            ChromeDrivers = new FirefoxDriver();
    }
    else {

            System.out.println("print invalid browser");

        }

        ChromeDrivers.manage().window().maximize();

        return ChromeDrivers;

    }

//=============================================================================================
==========================//
//Browser Control Methods
// url launch

    public static void urlLaunch(String url) {

        ChromeDrivers.get(url);

// In this method also using to launch the url
//       if (driver != null) {  // Check if driver is initialized
//             driver.get("https://www.saasgenie.ai/");  // Launch URL
//         } else {
//             System.out.println("WebDriver is not initialized.");
//         }
//


    }
```

```
 83
 84 //=============================================================================================
    ==============================//
 85 //Browser Control Methods
 86 // windows quit
 87
 88     public static void pageClose() {
 89         ChromeDrivers.quit();
 90
 91     }
 92 //=============================================================================================
    ==============================//
 93 // window waiting time
 94     public static void waitTime() throws InterruptedException {
 95          Thread.sleep(5000);
 96
 97     }
 98
 99 //=============================================================================================
    ==============================//
100 // get the title
101      public static String  getText(String element) {
102         element= ChromeDrivers.getTitle();
103         System.out.println(element);
104         return element;
105
106
107 // try catch method getting the title
108 //          try {
109 //          title= ChromeDrivers.getTitle();
110 //          } catch (Exception e) {
111 //          e.printStackTrace();
112 //          }
113
114 //          System.out.println(title);
115 //          return title;
116
117     }
118
119 //=============================================================================================
    ==============================//
120
121 // click the button
122
123     public static void clickElement(WebElement element) {
124            if (element != null) {
125                try {
126                    element.click();
127                    System.out.println("Element clicked successfully.");
128                } catch (Exception e) {
129                    System.out.println("Unable to click the element: " + e.getMessage());
130                }
131            } else {
132                System.out.println("Element is null, cannot click.");
133            }
134        }
135
```

```
136
137
138//   try {
139//       element.click();
140//       System.out.println("Element clicked successfully.");
141//   } catch (NoSuchElementException e) {
142//       System.out.println("Element not found: " + e.getMessage());
143//   } catch (ElementNotInteractableException e) {
144//       System.out.println("Element not interactable: " + e.getMessage());
145//   } catch (Exception e) {
146//       System.out.println("An unexpected error occurred: " + e.getMessage());
147//   }
148
149
150//   try {
151//
152//    waitforvisibilityofelement(element);
153//    element.click();
154//  } catch (Exception e) {
155//      e.printStackTrace();
156//  }
157
158//=============================================================================================///
159//Browser Control Methods
160public static void currentUrl() {
161        ChromeDrivers.getCurrentUrl();
162
163    }
164
165
166public static void navigateForward() {
167    ChromeDrivers.navigate().forward();
168
169}
170
171public static void navigateBack() {
172    ChromeDrivers.navigate().back();
173
174}
175
176
177public static void navigateRefresh() {
178    ChromeDrivers.navigate().refresh();
179
180}
181//=============================================================================================//
182// SendKeys method
183public static void sendkeysfirstName(WebElement element) {
184
185    element.sendKeys("yogeshwari");
186
187}
188public static void sendkeyslastName(WebElement element) {
189
190    element.sendKeys("Ramu");
191
192}
```

```
193
194 //=============================================================================================//
195 //Method to find an element by its XPath Name
196 //public static WebElement findElementByName(String name) {
197 //    try {
198 //        return ChromeDrivers.findElement(By.xpath(name));
199 //    } catch (Exception e) {
200 //        System.out.println("Element not found: " + e.getMessage());
201 //        return null; // Return null if the element is not found
202 //    }
203 //}
204
205 //Method to find an element by its XPath
206 //public static  WebElement findElementByXPath(String xpath) {
207 //    try {
208 //        return ChromeDrivers.findElement(By.xpath(xpath));
209 //    } catch (Exception e) {
210 //        System.out.println("Element not found: " + e.getMessage());
211 //        return null; // Return null if the element is not found
212 //    }
213 //}
214 //=============================================================================================//
215
216
217 public static void pageSource() {
218     ChromeDrivers.getPageSource();
219 //  try {
220 //      ChromeDrivers.getPageSource();
221 //  } catch (Exception e) {
222 //      System.out.println("Element not found:"+ e.getMessage());
223 //
224 //  }
225 }
226 //=============================================================================================//
227 //Windows handling methods
228     public static   void WindowsHandlesss(WebElement element){
229
230     }
231
232 //  public static void Windowhandles(String windowhandles) {
233 //
234 //
235 //      //String parentid =ChromeDrivers.getWindowHandle();
236 //      //element.click();
237 //
238 //      Set<String> windowhandless =ChromeDrivers.getWindowHandles();
239 //
240 //
241 //
242 //      System.out.println(windowhandless);
243 //      ChromeDrivers.switchTo().window(windowhandles);
244 //
245
246
247 //
248 //      String childid = ChromeDrivers.getWindowHandle();
249 //       element.click();
```

```
250 //       Set<String> parentid = ChromeDrivers.getWindowHandles();
251 //       String y = null;
252 //       for (String d : parentid) {
253 //       if (!d.equals(childid)) {
254 //       y = d;
255 //       }
256 //       }
257 //    ChromeDrivers.switchTo().window(y);
258
259
260
261
262 //}
263 //=============================================================================================//
264     //Right click mouse
265 public static  void rightClick(WebElement elementToRightClick) {
266     Actions actions = new Actions(ChromeDrivers);
267     actions.contextClick(elementToRightClick).perform();
268
269 }
270 //=============================================================================================/
    /
271 //Robot function using keyboard typing
272 public static void singleDownClick() throws AWTException {
273     Robot r= new Robot();
274     r.keyPress(KeyEvent.VK_DOWN);
275     r.keyRelease(KeyEvent.VK_DOWN);
276     r.keyPress(KeyEvent.VK_ENTER);
277     r.keyRelease(KeyEvent.VK_ENTER);
278
279 }
280 //=============================================================================================//
281 //Time wait function
282 public static void timeOut() {
283     ChromeDrivers.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
284
285 }
286 //=============================================================================================//
287 // Screenshot
288 public static void screenShot(WebElement screenshot) throws IOException {
289     TakesScreenshot ts = (TakesScreenshot) ChromeDrivers;
290      File screenshotAs = ts.getScreenshotAs(OutputType.FILE);
291      File saveFile = new File("C:\\Users\\User\\eclipse-workspace\\MavenNEW\\file\\img.png");
292      Files.copy(screenshotAs, saveFile);
293 }
294
295 //=============================================================================================//
296 // DropDown method
297
298 public static void getOption(WebElement element) {
299
300     Select getOption = new Select(element);
301
302     List<WebElement> option = getOption.getOptions();
303
304     //option.get(2);
305
```

```
306
307
308
309//  if (index >= 0 && index < option.size()) {
310//         // Click the specified option
311//         option.get(index).click();
312//     } else {
313//         System.out.println("Invalid index: " + index);
314//     }
315
316
317    // Select by visible text
318    //dropdown.selectByVisibleText("Option[3]");
319
320    // Select by index
321    //dropdown.selectByIndex(2);
322
323    // Select by value
324    //dropdown.selectByValue("optionValue");
325
326}
327
328
329}
330
331
```