BaseClass.java

```java
package com.mavenproject;

import java.awt.AWTException;
import java.awt.Robot;
import java.awt.event.KeyEvent;
import java.io.File;
import java.io.IOException;
import java.util.Set;
import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;

public class BaseClass {
    public static WebDriver driver;

//Browser launch
    public static WebDriver browserlaunch(String browser) {
        if (browser.equalsIgnoreCase("chrome")) {

            System.setProperty("webdriver.chrome.driver",
                    System.getProperty("user dir") + ("\\Driver\
\chromedriver.exe"));
            driver = new ChromeDriver();
        }

        else if (browser.equalsIgnoreCase("internetExplorer")) {
            System.setProperty("webdriver.ie.driver",
                    System.getProperty("user dir") + ("\\Driver\
\IEDriverServer.exe"));
            driver = new InternetExplorerDriver();
        }
        driver.manage().window().maximize();
        return driver;
    }

// Get URL
```

```
    public static void geturl(String url) {
        driver.get(url);
    }

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

//SendKeys
    public static void sendkeys(WebElement element, String input) {
        element.sendKeys("input");
    }

//Close
    public static void close() {
        driver.close();
    }

//Quit
    public static void quit() {
        driver.quit();
    }

//navigate to URL
    public static void navigatetourl(String url) {
        driver.navigate().to(url);
    }

// navigate to forward
    public static void navigatetoforward() {
        driver.navigate().forward();
    }

//Navigate to back
    public static void navigatetoback() {
        driver.navigate().back();
    }

//Navigate refresh
    public static void navigaterefresh() {
        driver.navigate().refresh();
    }

//Alert accept
    public static void alertaccept() {
        driver.switchTo().alert().accept();
```

```java
    }

//Alert dismiss
    public static void alertdismiss() {
        driver.switchTo().alert().dismiss();
    }

//Alert sendKeys
    public static void alertsendkeys(String input) {
        driver.switchTo().alert().sendKeys("input");
    }

//Mouse actions Click
    public static void mouseclick(WebElement element) {
        Actions ac = new Actions(driver);
        ac.click(element).build().perform();
    }

//Mouse actions Right Click
    public static void mouserightclick(WebElement element) {
        Actions ac = new Actions(driver);
        ac.contextClick(element).build().perform();
    }

//Mouse actions Double click
    public static void mousedoubleclick(WebElement element) {
        Actions ac = new Actions(driver);
        ac.doubleClick(element).build().perform();
    }

// Frame By id
    public static void frameid(String id) {
        driver.switchTo().frame(id);
    }

// Frame By index
    public static void frameindex(int index) {
        driver.switchTo().frame(index);
    }

// Frame By WebElement
    public static void framewebelement(WebElement element) {
        driver.switchTo().frame(element);
    }

// From Frame To Main Page
    public static void frametomainpage() {
        driver.switchTo().defaultContent();
    }

//Robot VK UP
    public static void vkup() throws AWTException {
        Robot r = new Robot();
        r.keyPress(KeyEvent.VK_UP);
        r.keyRelease(KeyEvent.VK_UP);
    }

//Robot VK DOWN
    public static void vkdown() throws AWTException {
        Robot r = new Robot();
        r.keyPress(KeyEvent.VK_DOWN);
        r.keyRelease(KeyEvent.VK_DOWN);
    }

//Robot VK ENTER
    public static void vkenter() throws AWTException {
        Robot r = new Robot();
        r.keyPress(KeyEvent.VK_ENTER);
        r.keyRelease(KeyEvent.VK_ENTER);
    }

//Get windows handling
    public static void getwindowshandling() {
        Set<String> windowHandles = driver.getWindowHandles();
        for (String s : windowHandles) {
            String tittle = driver.switchTo().window(s).getWindowHandle();
            System.out.println(tittle);
        }
    }

//Drop down by value
    public static void dropdownbyvalue(WebElement element, String value) {
        Select s = new Select(element);
        s.selectByValue(value);
    }

//Drop down by index
    public static void dropdownbyindex(WebElement element, int index) {
        Select s = new Select(element);
        s.selectByIndex(index);
    }

//Drop down by visible of text
    public static void dropdownbyvisibleoftext(WebElement element, String
text) {
        Select s = new Select(element);
        s.deselectByVisibleText(text);
    }

//Check box
    public static void checkbox(WebElement element) {
        element.click();
    }

//Is enabled
    public static void isenabled(WebElement element) {
        System.out.println(element.isEnabled());
    }

//Is displayed
    public static void isdisplayed(WebElement element) {
        System.out.println(element.isDisplayed());
    }

//Is Selected
    public static void isselected(WebElement element) {
        System.out.println(element.isSelected());
    }

//Get options
    public static void getoptions(WebElement element) {
        Select s = new Select(element);
        java.util.List<WebElement> options = s.getOptions();
        for (WebElement webElement : options) {
            System.out.println(webElement.getText());
        }
    }

//Get title
    public static void gettitle() {
        System.out.println(driver.getTitle());
    }

//Get current URL
    public static void getcurrenturl() {
        System.out.println(driver.getCurrentUrl());
    }

//Get text
    public static void gettext(WebElement element) {
        System.out.println(element.getText());
```

```java
    }

//Get attribute
    public static void getattribute(WebElement element, String input) {
        System.out.println(element.getAttribute(input));
    }

//Implicit wait
    public static void implicitwait(int sec, TimeUnit unit) {
        driver.manage().timeouts().implicitlyWait(sec, unit);
    }

//explicit wait
    public static void explicitwait(int sec, WebElement element) {
        WebDriverWait wait = new WebDriverWait(driver, sec);
        wait.until(ExpectedConditions.visibilityOf(element));
    }

//Take screenshot
    public static void takescreenshot(String location) throws IOException {
        TakesScreenshot sc = (TakesScreenshot) driver;
        File start = sc.getScreenshotAs(OutputType.FILE);
        File end = new File(location);
        FileUtils.copyFile(start, end);
    }

// Scroll intoView
    public static void scrollintoview(WebElement element) {
        JavascriptExecutor js = (JavascriptExecutor) driver;
        js.executeScript("arguments[0].scrollIntoView();", element);
    }

// Scroll By
    public static void scrollby(WebElement element) {
        JavascriptExecutor js = (JavascriptExecutor) driver;
        js.executeScript("window.scrollBy();", element);
    }

//Is Multiple
    public static void ismultiple(WebElement element) {
        Select s = new Select(element);
        System.out.println(s.isMultiple());
    }

// Radio button
    public static void radiobutton(WebElement element) {
        element.click();
    }
}

```