**Implementing Posting a Vehicle – Creating Action Classes:**

We have created object repositories in the previous chapter. We have to create action classes now. The same way as object repositories, we have to create two action classes adminnavigation.class and postvehicle.class.

**adminnavigation.class action class:**

**package** actions;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**public** **class** adminnavigation {

WebDriver driver;

**public** adminnavigation(WebDriver driver){

**this**.driver = driver;

}

**public** **void** adminclickdashboardlink(WebElement dashboardlink) {

dashboardlink.click();

}

**public** **void** adminclickbrandslink(WebElement brands) {

brands.click();

}

**public** **void** adminclickcreatebrand(WebElement createbrand) {

createbrand.click();

}

**public** **void** adminmanagebrands(WebElement managebrands) {

managebrands.click();

}

**public** **void** adminvehicles(WebElement vehicles) {

vehicles.click();

}

**public** **void** adminpostavehicle(WebElement postvehicle) {

postvehicle.click();

}

**public** **void** adminmanagevehicles(WebElement managevehicles) {

managevehicles.click();

}

**public** **void** adminmanagebookings(WebElement managebooking) {

managebooking.click();

}

**public** **void** adminmanagetestimonials(WebElement managetestimonials) {

managetestimonials.click();

}

**public** **void** adminmanagecontactusquery(WebElement contactusquery) {

contactusquery.click();

}

**public** **void** adminregusers(WebElement regusers) {

regusers.click();

}

**public** **void** adminmanagepages(WebElement managepages) {

managepages.click();

}

**public** **void** adminupdatecontactinfo(WebElement updatecontactinfo) {

updatecontactinfo.click();

}

**public** **void** adminmanagesubscribers(WebElement managesubscribers) {

managesubscribers.click();

}

}

**postvehicle.class action class:**

**package** actions;

**import** java.util.List;

**import** org.openqa.selenium.By;

**import** org.openqa.selenium.JavascriptExecutor;

**import** org.openqa.selenium.WebDriver;

**import** org.openqa.selenium.WebElement;

**import** org.openqa.selenium.interactions.Actions;

**import** org.testng.Assert;

**public** **class** postvehicle {

WebDriver driver;

**public** postvehicle(WebDriver driver) {

**this**.driver = driver;

}

**public** **void** adminvehicletitle(WebElement vehicletitle, String title) {

vehicletitle.sendKeys(title);

}

**public** **void** adminselectbrand(WebElement brands, String brand) {

List<WebElement> links = brands.findElements(By.*tagName*("li"));

**for** (**int** i = 1; i < links.size(); i++)

{

System.***out***.println(links.get(i).getText());

**if**(brand.equalsIgnoreCase(links.get(i).getText())) {

brands.findElements(By.*tagName*("li")).get(i).click();

**break**;

}

}

}

**public** **void** adminvehicleoverview(WebElement vehicleoverview, String overview) {

vehicleoverview.sendKeys(overview);

}

**public** **void** adminvehpriceperday(WebElement vehpriceperday, String priceperday) {

vehpriceperday.sendKeys(priceperday);

}

**public** **void** adminselectfueltype(WebElement fueltypes, String fueltype) {

List<WebElement> fueltypes\_li = fueltypes.findElements(By.*tagName*("li"));

**for** (**int** i = 1; i < fueltypes\_li.size(); i++)

{

System.***out***.println(fueltypes\_li.get(i).getText());

**if**(fueltype.equalsIgnoreCase(fueltypes\_li.get(i).getText())) {

fueltypes.findElements(By.*tagName*("li")).get(i).click();

**break**;

}

}

}

**public** **void** adminvehmodelyear(WebElement vehmodelyear, String modelyear) {

vehmodelyear.sendKeys(modelyear);

}

**public** **void** adminvehselectcapcity(WebElement vehcapacity, String capacity) {

vehcapacity.sendKeys(capacity);

}

**public** **void** adminuploadimage1(WebElement upimage1, String image1) {

upimage1.sendKeys(image1);

}

**public** **void** adminuploadimage2(WebElement upimage2, String image2) {

upimage2.sendKeys(image2);

}

**public** **void** adminuploadimage3(WebElement upimage3, String image3) {

upimage3.sendKeys(image3);

}

**public** **void** adminuploadimage4(WebElement upimage4, String image4) {

upimage4.sendKeys(image4);

}

**public** **void** adminuploadimage5(WebElement upimage5, String image5) {

upimage5.sendKeys(image5);

}

**public** **void** adminselairconditioning(WebElement selairconditioner, String airconditioner) {

**if**(airconditioner.equalsIgnoreCase("yes"))

selairconditioner.click();

}

**public** **void** adminselpowerdoorlocks(WebElement selpowerdoorlocks, String powerdoorlocks) {

**if**(powerdoorlocks.equalsIgnoreCase("yes"))

selpowerdoorlocks.click();

}

**public** **void** adminselantibreakingsystem(WebElement selantibreakingsystem, String antilockbreakingsystem) {

**if**(antilockbreakingsystem.equalsIgnoreCase("yes"))

selantibreakingsystem.click();

}

**public** **void** adminselbreakassist(WebElement selbreakassist, String breakassist) {

**if**(breakassist.equalsIgnoreCase("yes"))

selbreakassist.click();

}

**public** **void** adminselpowersteering(WebElement selpowersteering, String powersteering) {

**if**(powersteering.equalsIgnoreCase("yes"))

selpowersteering.click();

}

**public** **void** adminseldriverairbag(WebElement seldriverairbag, String driverairbag) {

**if**(driverairbag.equalsIgnoreCase("yes"))

seldriverairbag.click();

}

**public** **void** adminselpassengerairbag(WebElement selpassengerairbag, String passengerairbag) {

**if**(passengerairbag.equalsIgnoreCase("yes"))

selpassengerairbag.click();

}

**public** **void** adminselpowerwindows(WebElement selpowerwindows, String powerwindows) {

**if**(powerwindows.equalsIgnoreCase("yes"))

selpowerwindows.click();

}

**public** **void** adminselcdplayer(WebElement selcdplayer, String cdplayer) {

**if**(cdplayer.equalsIgnoreCase("yes"))

selcdplayer.click();

}

**public** **void** adminselcentrallocking(WebElement selcentrallocking, String centrallocking) {

**if**(centrallocking.equalsIgnoreCase("yes"))

selcentrallocking.click();

}

**public** **void** adminselcrashsensor(WebElement selcrashsensor, String crashsensor) {

**if**(crashsensor.equalsIgnoreCase("yes"))

selcrashsensor.click();

}

**public** **void** adminselleatherseats(WebElement selleatherseats, String leatherseats) {

**if**(leatherseats.equalsIgnoreCase("yes"))

selleatherseats.click();

}

**public** **void** adminsubmitvehicle(WebElement submitvehicle) **throws** InterruptedException {

JavascriptExecutor js = (JavascriptExecutor) driver;

js.executeScript("window.scrollBy(0,1000)");

submitvehicle.click();

}

**public** **void** admingetsuccessmsgafterpostvehicle(WebElement message) {

String successmessage = message.getText();

String expectedmessage = "SUCCESS:Vehicle posted successfully";

System.***out***.println(successmessage);

Assert.*assertEquals*(successmessage, expectedmessage);

}

}

In the adminsubmitvehicle() method, clicking the button is not working if it is not in visible view. So we are moving the scrollbar using javascriptexecutor then we are clicking.

In admingetsuccessmsgafterpostvehcile, we are using asserts to verify the successmessage.