**Day 1**

**Test Case**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Cases | Process | Steps to Execute | Expected Results |
| TC01 | Edit a tab | 1. Login to admin portal 2. Edit a tab and modify the sort order. | Verify the validation message and Modified Tab name, sort order, status is displayed on the table properly |

**Procedure:**

* Consider the ‘BaseTest.java’ and ‘AdminPage.java’ codes for the test case
* The Data Driven Page Factory method is used
* All the xpath for required fields are created in the ‘AdminPage.java’
* The method is called in the ‘BaseTest.java’
* The page is automated in the way like
* Enter the login mail and password and click on the login button
* Once logged in click on the ‘Catalog’ and click on ‘Tab’
* In that to edit a particular tab select the row of which tab has to be edited and first click on edit and then click on the ‘Edit’
* It opens to a page in there change the tab name, sort order and status
* Once done click on save button
* Once save is clicked it move back to the first page and check whether the change is done

**Code from the java files executed:**

**BaseTest.Java:**

package com.ibm.test;

import java.io.IOException;

import java.util.HashMap;

import java.util.concurrent.TimeUnit;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.support.ui.WebDriverWait;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

import org.testng.annotations.DataProvider;

import org.testng.annotations.Test;

import com.ibm.pages.AdminPage;

import com.ibm.utilities.ExcelUtil;

import com.ibm.utilities.PropertiesFileHandler;

public class BaseTest {

//TestCase 1 given to check the login of admin tab and modifying the tab and its sort order

@Test

public void testcase1() throws IOException, InterruptedException

{

//To take the data given in the 'data.properties' file

String file="./TestData/data.properties";

PropertiesFileHandler propFileHandler = new PropertiesFileHandler();

HashMap<String, String> data= propFileHandler.getPropertiesAsMap(file);

String url= data.get("url");

//To use the chrome driver and get the url, maximize window

System.setProperty("webdriver.chrome.driver", "./drivers/chromedriver.exe");

WebDriver driver = new ChromeDriver();

WebDriverWait wait = new WebDriverWait(driver, 60);

driver.manage().window().maximize();

driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

driver.get(url);

String userName = data.get("username");

String password = data.get("password");

//To run the xpath and method created in AdminPage.java

AdminPage home = new AdminPage(driver);

//To get the mail address from properties file and calling the method from Admin.java page

home.EnetrEmailAddress(userName);Thread.sleep(3000);

//To get the password

home.EnetrPassword(password);Thread.sleep(3000);

//To click on the login button

home.ClickonLoginButton();Thread.sleep(3000);

//To click on the Catalog button

home.ClickonCatalogTabButton();Thread.sleep(3000);

//To click on tab button

home.ClickonTabButton();Thread.sleep(3000);

//To click on Action button of the first data

home.ClickonAction();Thread.sleep(3000);

//To click on the edit button

home.ClickonEditButton();Thread.sleep(3000);

//To edit the tab name

home.EntertheTabName();Thread.sleep(2000);

//To edit the sort order

home.EnetrtheSortOrder();Thread.sleep(2000);

//To change the status

home.EntertheStatus();Thread.sleep(2000);

//To click on save once edit is done

home.ClickonTheSaveButton();Thread.sleep(2000);

//To verify whether after the save button it is going back and changes are implement by checking it with the first page title over there

Assert.assertTrue(driver.findElement(By.xpath("//img[@title='EasyOrdering']")).isDisplayed());Thread.sleep(2000);

home.ClickonLogoutButton();

}

}

**AdminPage.java:**

**package** com.ibm.pages;

**import** java.sql.Driver;

**import** java.util.List;

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

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

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

**import** org.openqa.selenium.support.FindAll;

**import** org.openqa.selenium.support.FindBy;

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

**import** org.openqa.selenium.support.ui.WebDriverWait;

**public** **class** AdminPage {

//Following are the Xpath from the page

//To give the login mail id

@FindBy (xpath="//input[@name='email']")

WebElement Emailele;

//To give the password

@FindBy (xpath="//input[@name='pword']")

WebElement PasswordEle;

//To click on the login button

@FindBy (xpath="//button[@class='btn btn-labeled btn-info m-b-5']")

WebElement LoginButtonEle;

//To click on the Catalog Tab

@FindBy (xpath="//a[@href='#']")

WebElement CatalogTabEle;

//To click on the 'Tab' under 'Catalog'

@FindBy(xpath="//a[text()=' Tabs']")

WebElement TabButtonEle;

//To click on the 'Add new' button

@FindBy(xpath="//a[@title='Add New']")

WebElement AddButtonEle;

//To enter the name while adding a new tab or modifying tab

@FindBy(xpath="//input[@name='name']")

WebElement TabnameEle;

//To enter the sort order while adding or modifying the tab name

@FindBy(xpath="//input[@name='sort']")

WebElement SortELe;

//To change the status

@FindBy(xpath="//select[@name='status']/option[3]")

WebElement StatusEle;

//To save the content

@FindBy(xpath="//button[@title='Save']")

WebElement SaveEle;

//To click on the action in 'Tab'

@FindBy(xpath="//table[@id='dataTableExample2']/tbody/tr/td[5]/div/button[1]")

WebElement ActionEle;

//To click on the Edit button in 'Action'

@FindBy(xpath="//table[@id='dataTableExample2']/tbody/tr/td[5]/div/ul/li/a[1]")

WebElement EditEle;

//To click on the login path

@FindBy(xpath="//a[@title='Logout']")

WebElement LogoutEle;

//For using webdriver

**public** AdminPage(WebDriver driver)

{

PageFactory.*initElements*(driver, **this**);

}

**public** **void** EnetrEmailAddress(String userName)

{

Emailele.sendKeys(userName);

}

**public** **void** EnetrPassword(String password)

{

PasswordEle.sendKeys(password);

}

**public** **void** ClickonLoginButton()

{

LoginButtonEle.click();

}

**public** **void** ClickonCatalogTabButton()

{

CatalogTabEle.click();

}

**public** **void** ClickonTabButton()

{

TabButtonEle.click();

}

**public** **void** ClickonAddButton()

{

AddButtonEle.click();

}

**public** **void** EntertheTabName()

{

TabnameEle.clear();

TabnameEle.sendKeys("##AABB");

}

**public** **void** EnetrtheSortOrder()

{

SortELe.clear();

SortELe.sendKeys("1");

}

**public** **void** EntertheStatus()

{

StatusEle.click();

}

**public** **void** ClickonTheSaveButton()

{

SaveEle.click();

}

**public** **void** ClickonAction()

{

ActionEle.click();

}

**public** **void** ClickonEditButton()

{

EditEle.click();

}

**public** **void** ClickonLogoutButton()

{

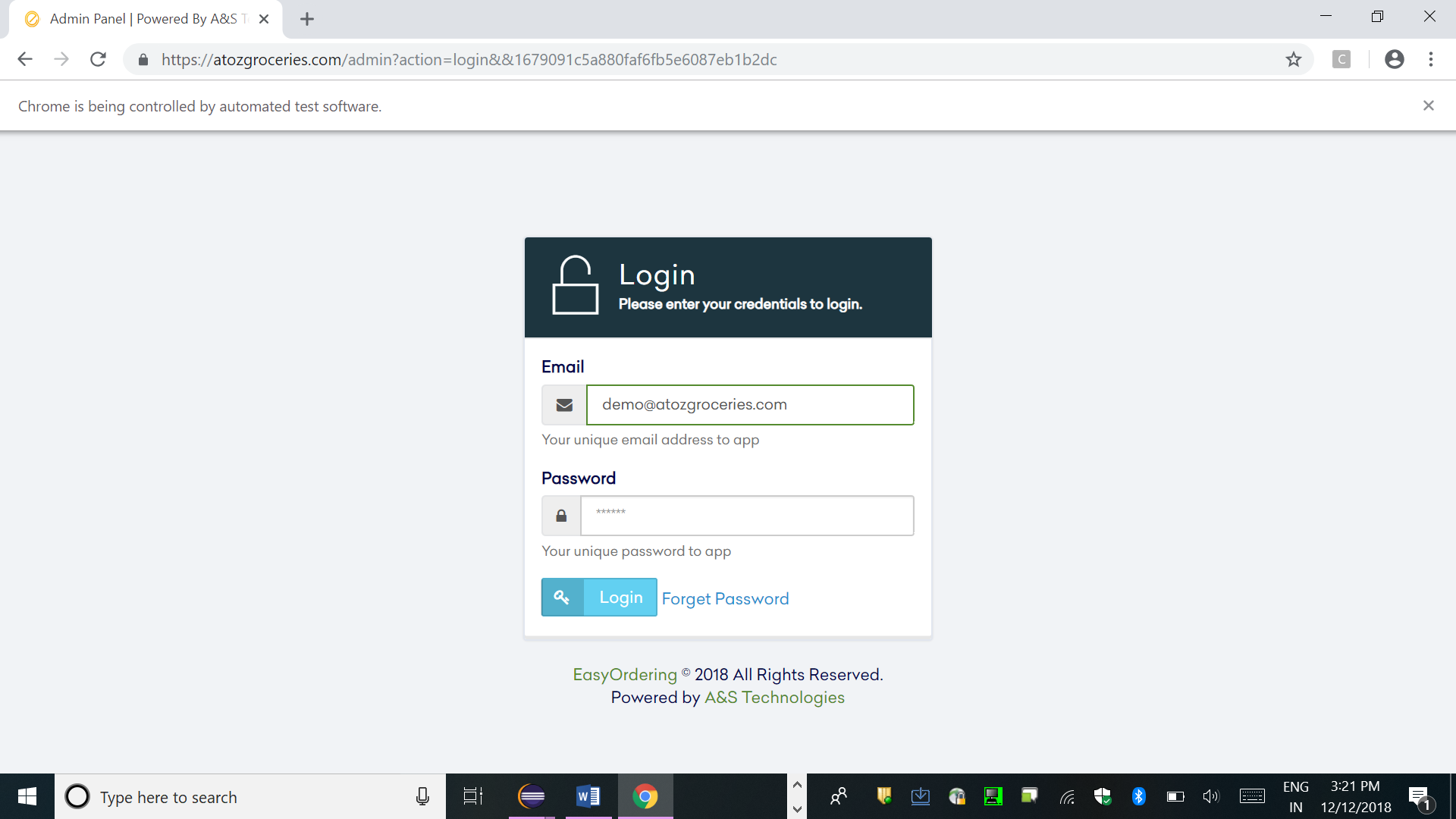
LogoutEle.click();

}

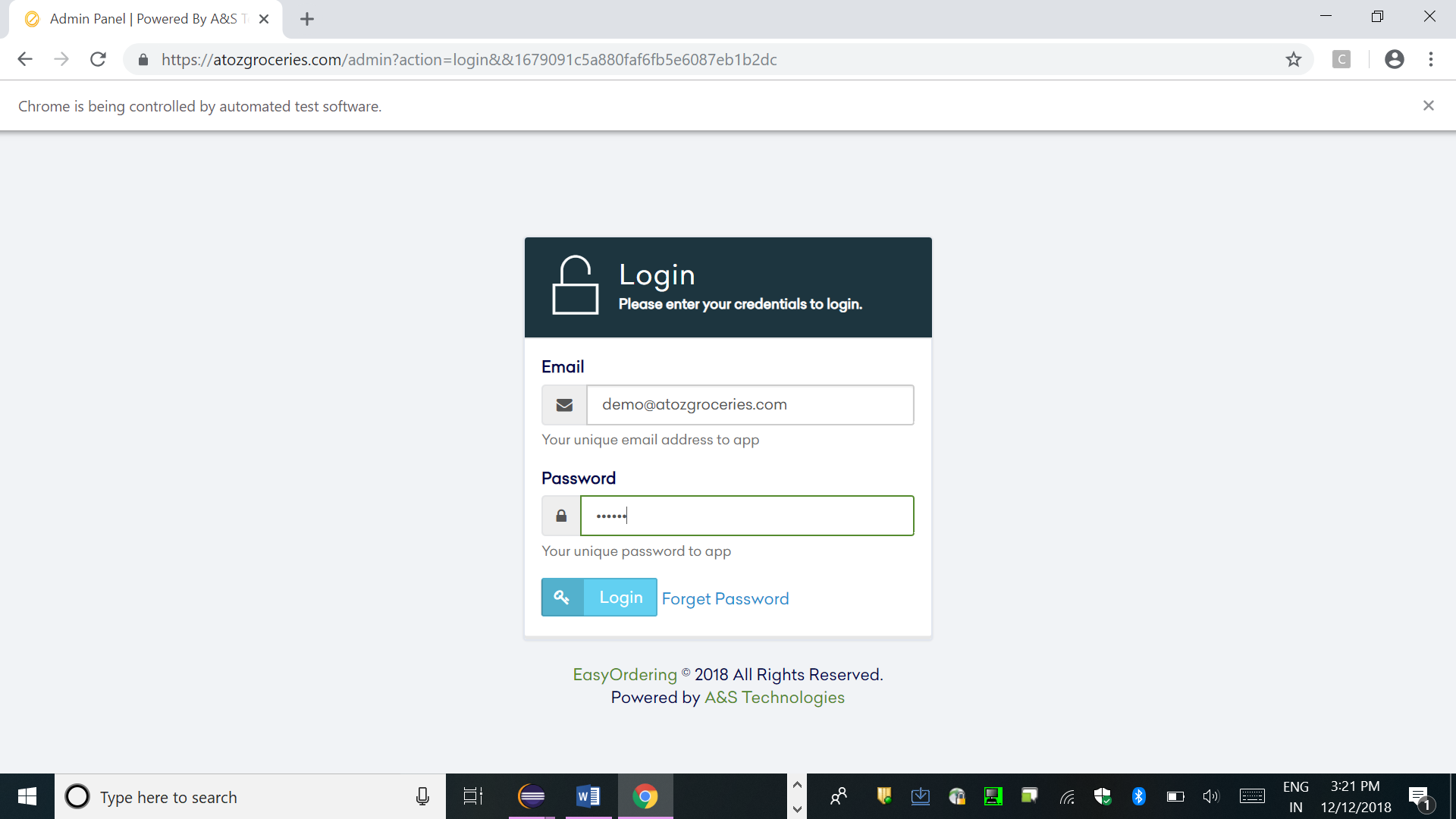
}

**Screenshot from the webpage automated:**

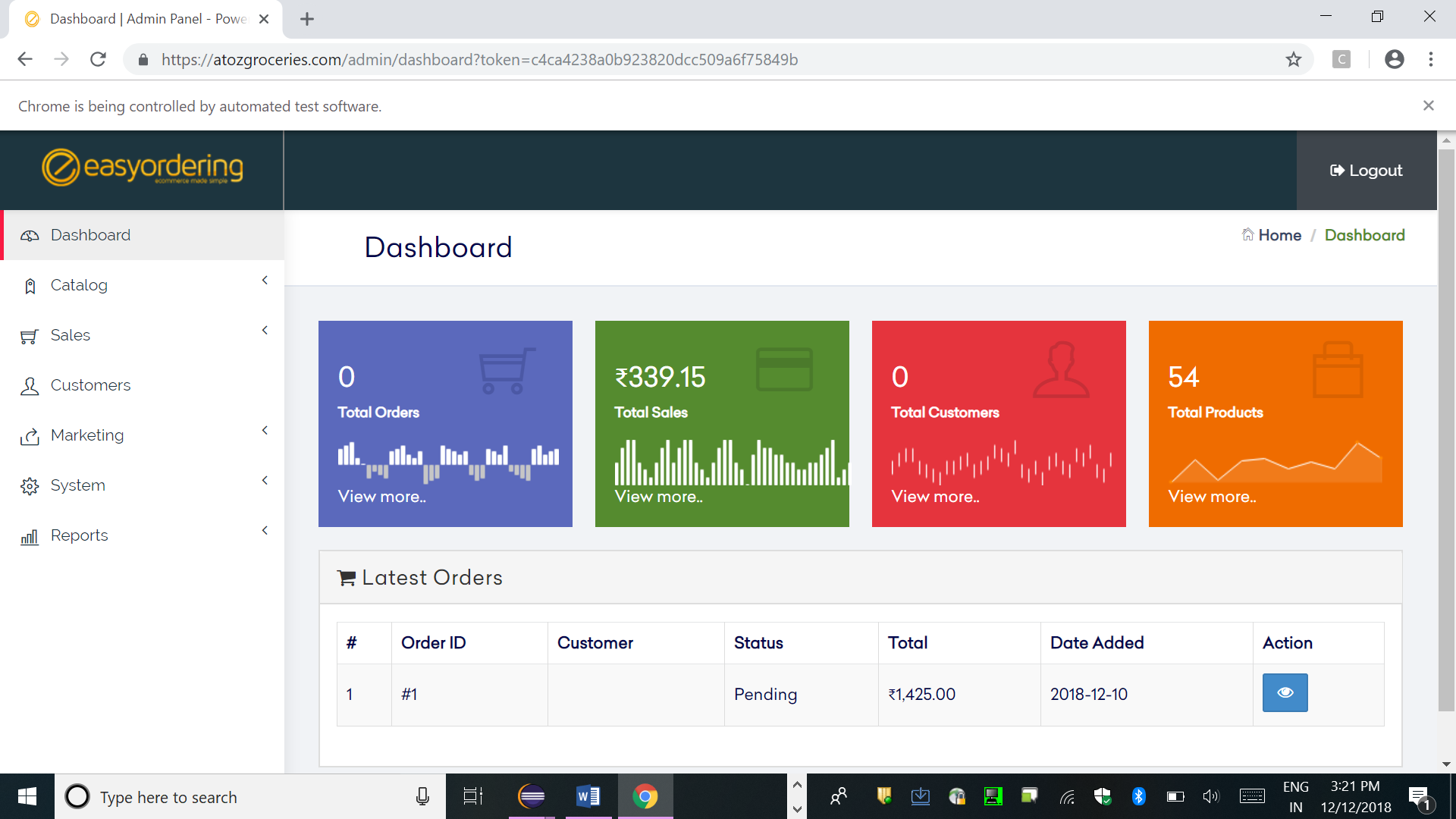
1.Login page: email



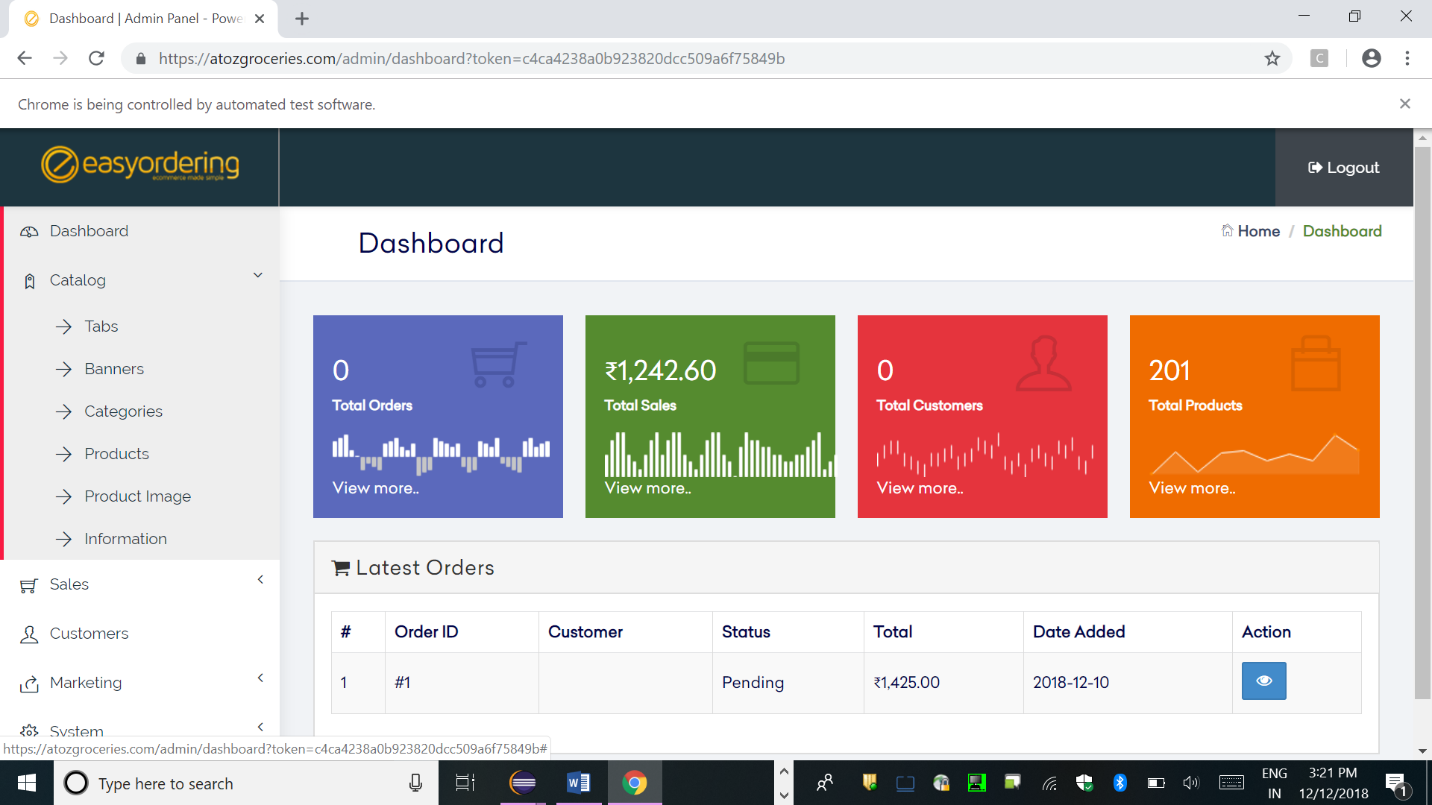
2.Loginpage: Password



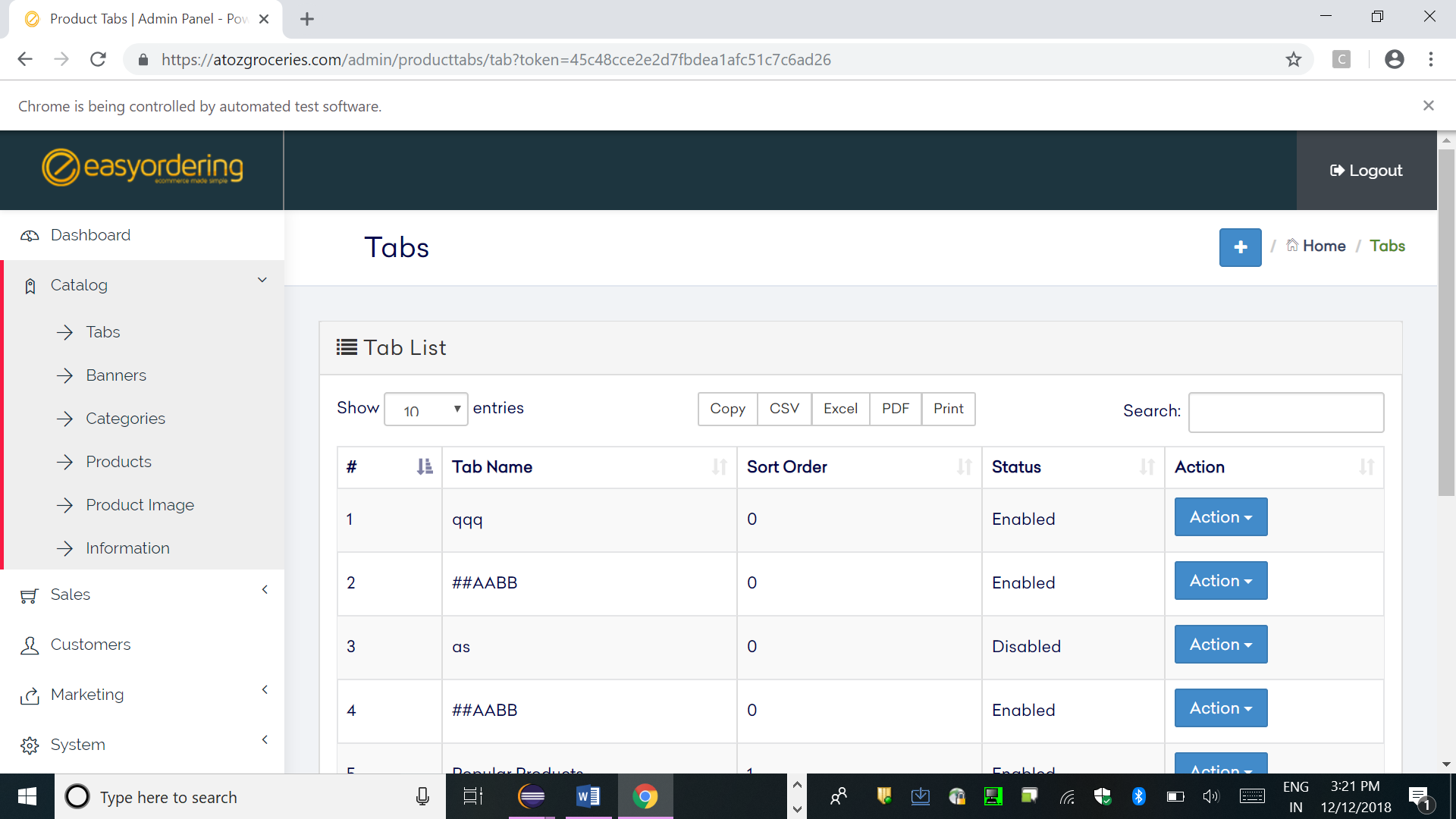
3.Page opened once logis is done:



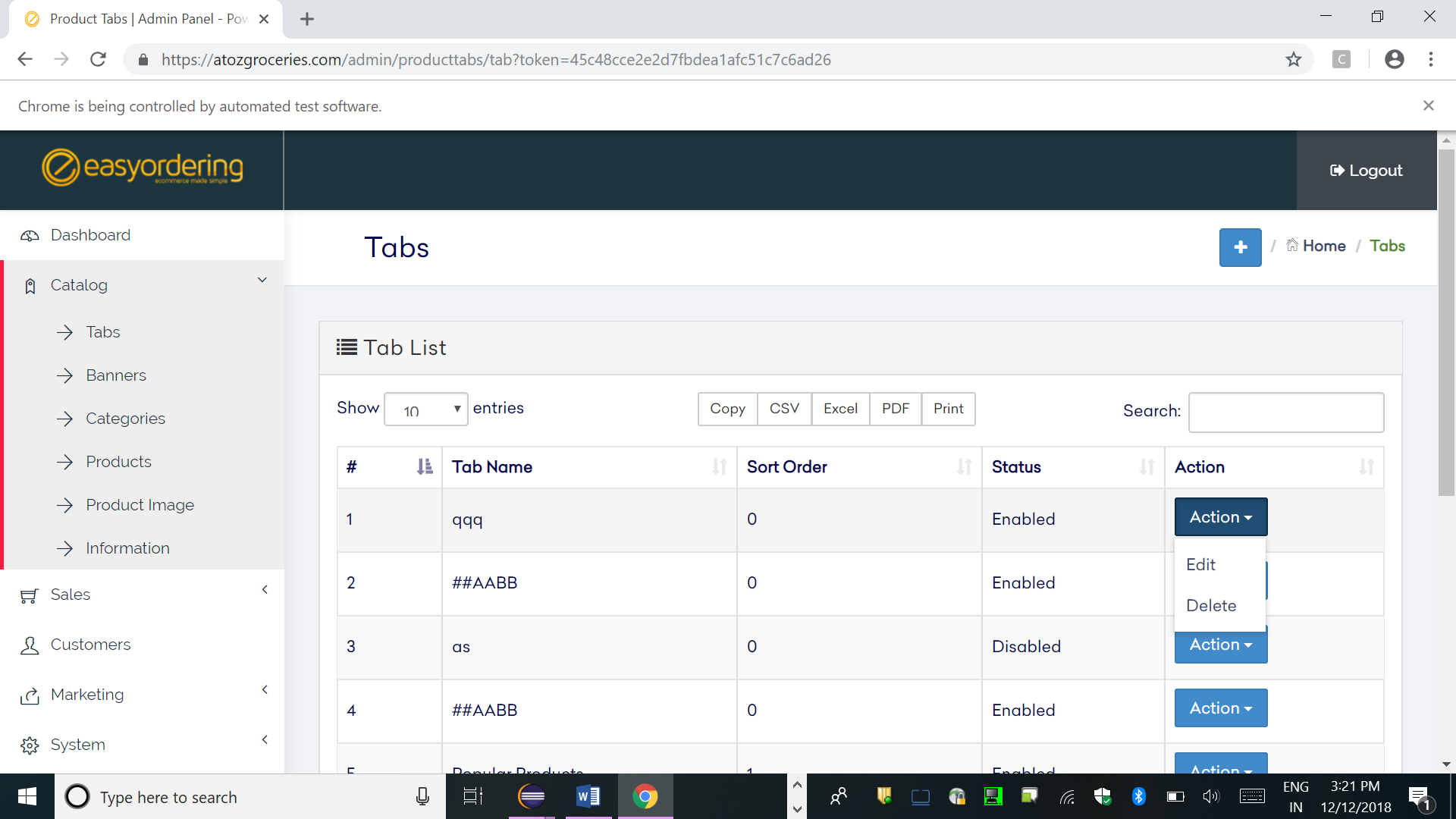
4.Click made on the ‘Catalog’



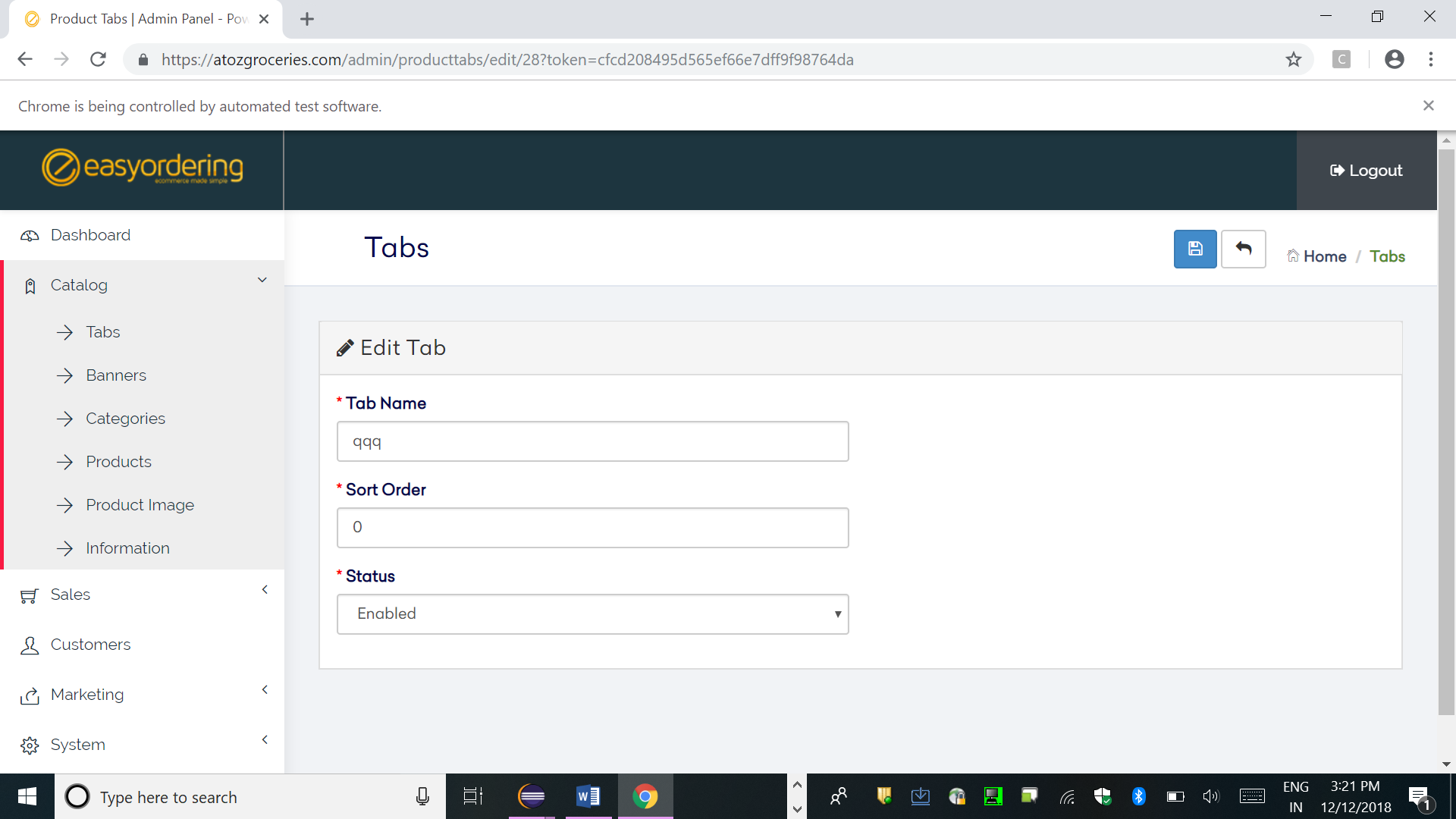
5.’Tabs’ is selected



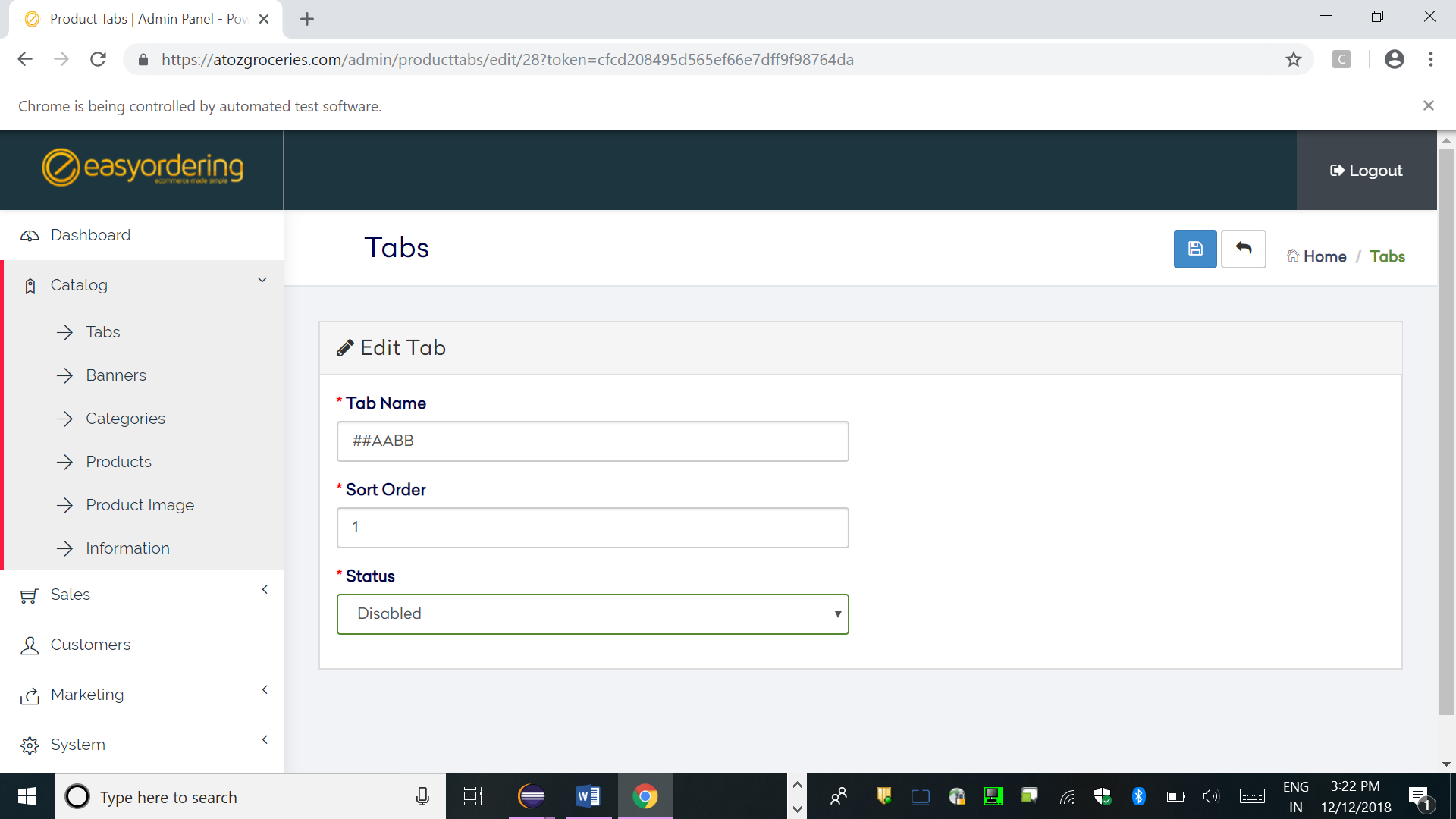
6.’Edit’ is done to the first value is the table



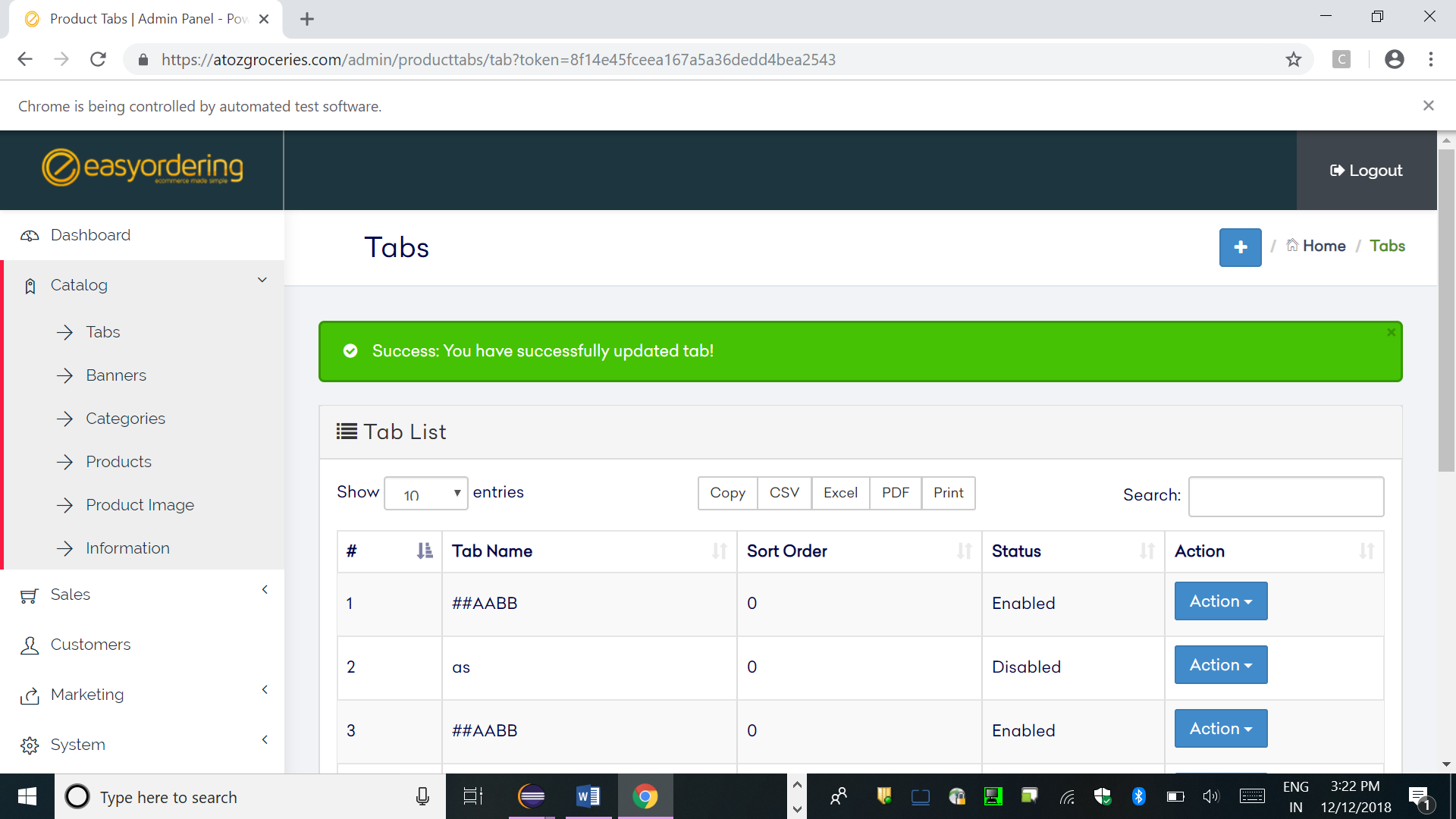
7.The first row of the table is taken for editing



8.The value given in the ‘Admin.java’ is taken



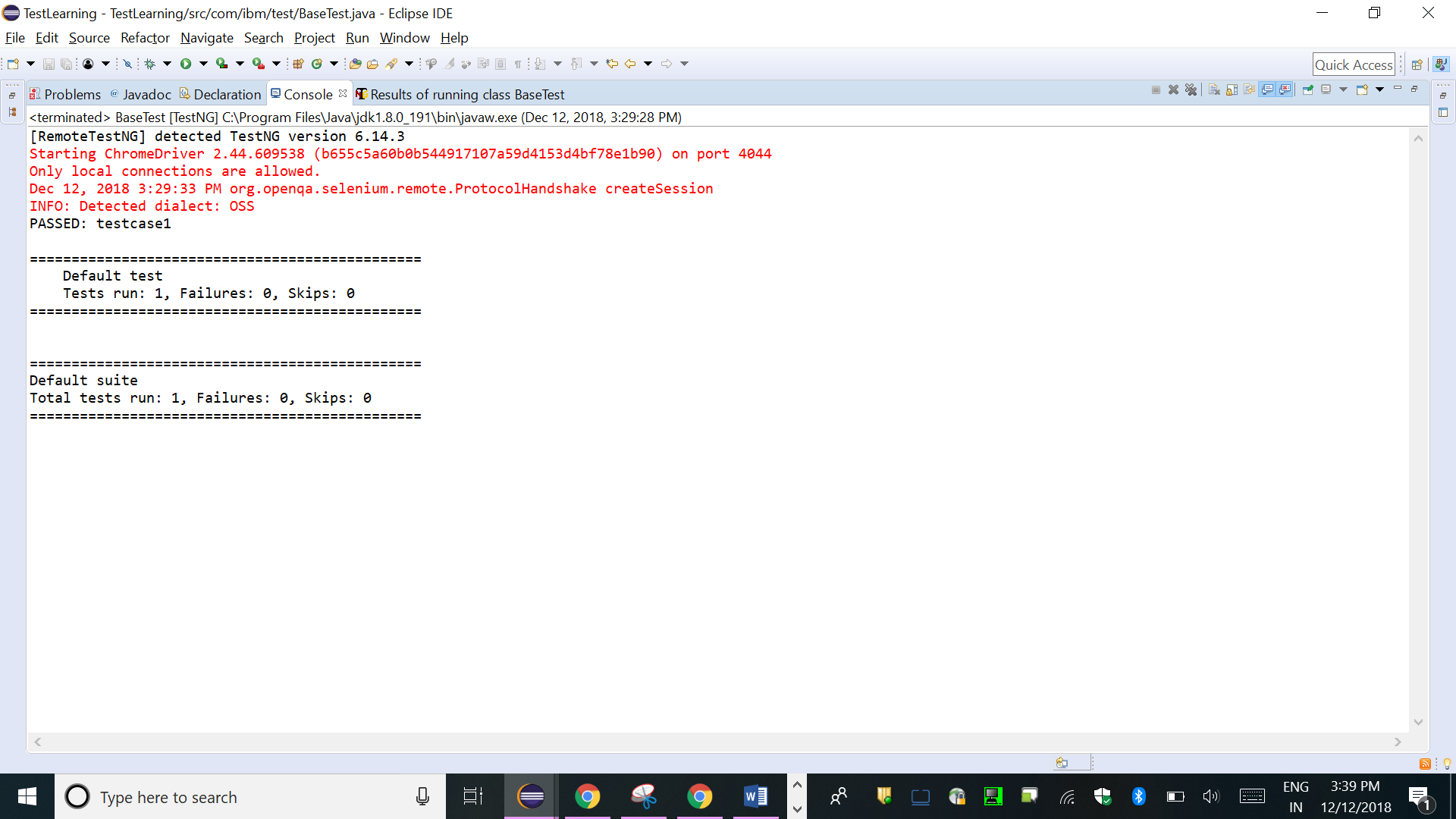
9.Once edit is done and saved it go backs to the first page



10.The screenshot of new element added



**Screenshot from Eclipse output:**



**Attachment of Java file and class file:**



