Day 2 Test Case

Day 2 Test Case:

Test Cases	Process	Steps To Execute	Expected Results
TC02	Add a new banner	1. Login to admin portal 2. Add new banner 3. Do not fill the mandatory fields and check	Click the save icon and verify it is on the same page. Negative test case validation

Output:

Step by step execution (Screenshot) document with the actual result in pdf format.

Program

BaseTest.java

```
package com.ibm.test;


import static org.testng.Assert.assertTrue;


import java.io.IOException;

import java.util.HashMap;

import java.util.concurrent.TimeUnit;


import org.junit.Assert;

import org.openqa.selenium.JavascriptExecutor;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

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

import org.testng.annotations.AfterMethod;

import org.testng.annotations.BeforeMethod;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.BeforeTest;
```

```
//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 {
	WebDriver driver;
	WebDriverWait wait;
	PropertiesFileHandler propFileHandler;
	HashMap<String, String> data;
	//Use properties to solve the simple test cases.
	@BeforeSuite
	public void testcase1() throws IOException
	{
		//Locating property file
		String file="./TestData/data.properties";
		propFileHandler = new PropertiesFileHandler();
		data= propFileHandler.getPropertiesAsMap(file);
	}

	@BeforeMethod
	public void Initialization()
	{
		//Open chrome and wait
		System.setProperty("webdriver.chrome.driver","./drivers/chromedriver.exe");
		driver=new ChromeDriver();
```

```
		wait=new WebDriverWait(driver,60);
		driver.manage().window().maximize();
		driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);

}

@AfterMethod
public void closeBrowser() {
		driver.quit();
}

/*@Test(priority=1)
public void Category() throws IOException, InterruptedException
{
		//using data from property file
		String url = data.get("url");
		String userName=data.get("username");
		String passWord=data.get("password");
		driver.get(url);

		//Creating adminPage object and calling AdminPage methods here
		AdminPage adminPage=new AdminPage(driver, wait);
		adminPage.enterEmail(userName);
		adminPage.enterPassword(passWord);

		adminPage.CaptureScreenshot(driver);

		adminPage.clickOnLogin();
```

```
adminPage.Catelog();

adminPage.CaptureScreenshot(driver);

adminPage.selCategory();

adminPage.CaptureScreenshot(driver);

int lastColoumValue=Integer.parseInt(adminPage.tableValue());

if(lastColoumValue<=10)
{
	System.out.println("First 10 Category lists are displayed");
}
else
{
	Assert.fail("First 10 Category lists are not displayed");
}

adminPage.pageTwoNavi();
adminPage.CaptureScreenshot(driver);

adminPage.nextButton();
adminPage.CaptureScreenshot(driver);

adminPage.previousButton();
Thread.sleep(2000);
adminPage.CaptureScreenshot(driver);
```

```
}*/

@Test(priority=2)
public void NegativeTestAddBanner() throws IOException
	{
	//using data from property file
				String url = data.get("url");
				String userName=data.get("username");
				String passWord=data.get("password");
				driver.get(url);

				AdminPage adminPage=new AdminPage(driver, wait);
				adminPage.enterEmail(userName);
				adminPage.enterPassword(passWord);

				adminPage.CaptureScreenshot(driver);

				adminPage.clickOnLogin();

				adminPage.Catelog();

				adminPage.CaptureScreenshot(driver);

				adminPage.clickOnBanners();

				adminPage.CaptureScreenshot(driver);

				adminPage.clickOnAddNewButton();
```

```
                adminPage.CaptureScreenshot(driver);

                adminPage.clickOnSaveButton();

                adminPage.CaptureScreenshot(driver);

                adminPage.validationMessage();

                String titleOfThePage=driver.getTitle();

                String bannerPageTitle=data.get("bannertitle");

                //Verifying current Title to confirm page is not changed
                Assert.assertEquals(titleOfThePage, bannerPageTitle);

                adminPage.bannerName();

                adminPage.clickOnSaveButton();

                adminPage.CaptureScreenshot(driver);

                //Verifying current Title to confirm page is not changed
                Assert.assertEquals(titleOfThePage, bannerPageTitle);
        }

}
```

AdminTest.Java program

```
package com.ibm.pages;
```

```
import static org.testng.Assert.assertEquals;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Date;
import java.util.List;
import java.util.Properties;
import java.util.concurrent.TimeUnit;

import org.apache.commons.io.FileUtils;
import org.openqa.selenium.By;
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.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.annotations.Test;

import com.ibm.test.LoginPage;

public class AdminPage {
```

```
//xpaths for locating webelements in webpage
@FindBy(name="email")
WebElement emailAdd;

@FindBy(name="pword")
WebElement enterPass;

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

@FindBy(xpath="//*[text()=' Catalog']")
WebElement catalogEle;

@FindBy(xpath="//*[text()=' Categories']")
WebElement categorySel;

@FindBy(xpath="//table[@id='dataTableExample2']/tbody/tr[10]/td[1]")
WebElement tablelastColValue;

@FindBy(xpath="(//*[@aria-controls='dataTableExample2'])[16]")
WebElement pageTwoButton;

@FindBy(xpath="(//*[@aria-controls='dataTableExample2'])[18]")
WebElement nextButtoninPage;

@FindBy(xpath="(//*[@aria-controls='dataTableExample2'])[14]")
WebElement previousButtonInPage;
```

```
@FindBy(xpath="//*[text()=' Banners']")
WebElement bannersSel;

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

@FindBy(xpath="//button[@type='Submit']")
WebElement clickOnSave;

@FindBy(name="name")
WebElement bannerNameField;

WebDriverWait wait;
WebDriver driver;

public AdminPage(WebDriver driver, WebDriverWait wait)
{
        PageFactory.initElements(driver, this);
        this.driver=driver;
        this.wait=wait;
}

//Enter email address
public void enterEmail(String username)
{
        emailAdd.sendKeys(username);
}
//To enter password
public void enterPassword(String password)
```

```
{
	enterPass.sendKeys(password);
}
//click on Login button
public void clickOnLogin()
{
	loginEle.click();
}

//Capture screen shot method.. including date in between filename
public void CaptureScreenshot(WebDriver driver) throws IOException
{
	File file=((TakesScreenshot) driver).getScreenshotAs(OutputType.FILE);

	//Created date method to include date in screen shot file name
	Date date=new Date();

	//Changing : into - to understand easily and seperate day-month-year
	String currentDate=date.toString().replace(":", "-");

	FileUtils.copyFile(file, new File("./ProjectOneSS/LoginPage-"+currentDate+".png"));
}
//Selecting catelog
 public void Catelog()
 {
	catalogEle.click();
 }
 //Selecting category
public void selCategory()
```

```
{
	categorySel.click();
}
//TableColValue from Last row
public String tableValue()
{
	return tablelastColValue.getText();
}

//Method to verify Page 2 navigation button is displayed and working or not
public void pageTwoNavi()
{
	if(pageTwoButton.isDisplayed() && pageTwoButton.isEnabled())
	{
		System.out.println("Page navigation is displayed and enabled");
		pageTwoButton.click();
	}
	else
	{
		//Condition to fail test case if next page number 2 not displayed or not enabled
		Assert.fail("Page Navigation is failed");
	}

}
//Method to verify Next Button is displayed and working or not
public void nextButton()
{
	if(nextButtoninPage.isDisplayed() && nextButtoninPage.isEnabled())
	{
```

```
                System.out.println("Next Button is displayed and Enabled");
                nextButtoninPage.click();
        }
        else
        {
                //Condition to fail test case if Next button not displayed or not enabled
                Assert.fail("Next Button is disabled");
        }

}
//Method to verify Previous button is displayed and working or not
public void previousButton()
{
        if(previousButtonInPage.isDisplayed() && previousButtonInPage.isEnabled())
        {
                System.out.println("Previous Button is displayed and Enabled");
                previousButtonInPage.click();
        }
        else
        {
                //Condition to fail test case if Previous button not displayed or not enabled
                Assert.fail("Previous button is disabled");
        }

}
//To click on Banners on LHS
public void clickOnBanners()
{
        bannersSel.click();
```

```
	}

	//To click on Add new button in Banners
	public void clickOnAddNewButton()
	{
		addNewButton.click();
	}

	//Click on Save button
	public void clickOnSaveButton()
	{
		clickOnSave.click();
	}

	//Verifying validation message
	public void validationMessage()
	{
		JavascriptExecutor js=(JavascriptExecutor)driver;
		String sText=js.executeScript("return
document.getElementsByName('name')[\"0\"].validationMessage").toString();
		System.out.println(sText);

	}
	//Enter in Banner Name field
	public void bannerName()
	{
		bannerNameField.sendKeys("BannerTestCase");
	}
}
```

Data.properties

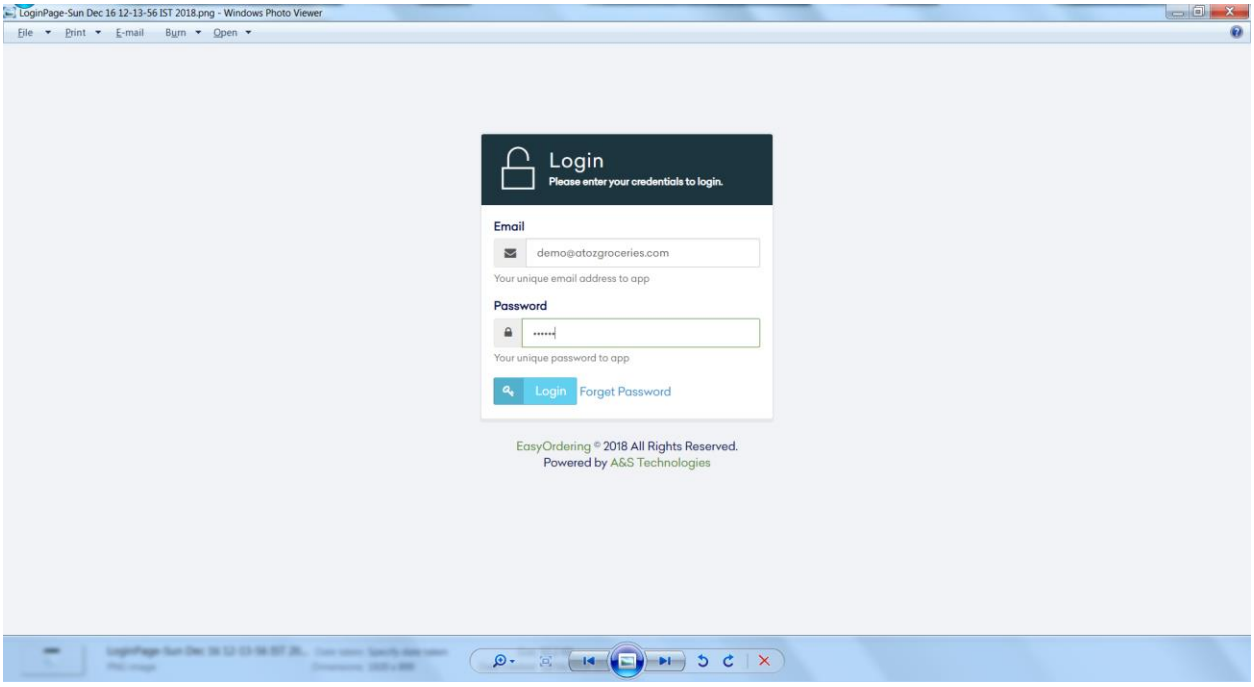
```
url=https://atozgroceries.com/admin
username=demo@atozgroceries.com
password=456789
bannertitle=Banner | Admin Panel - Powered By A&S
```

Testng.xml

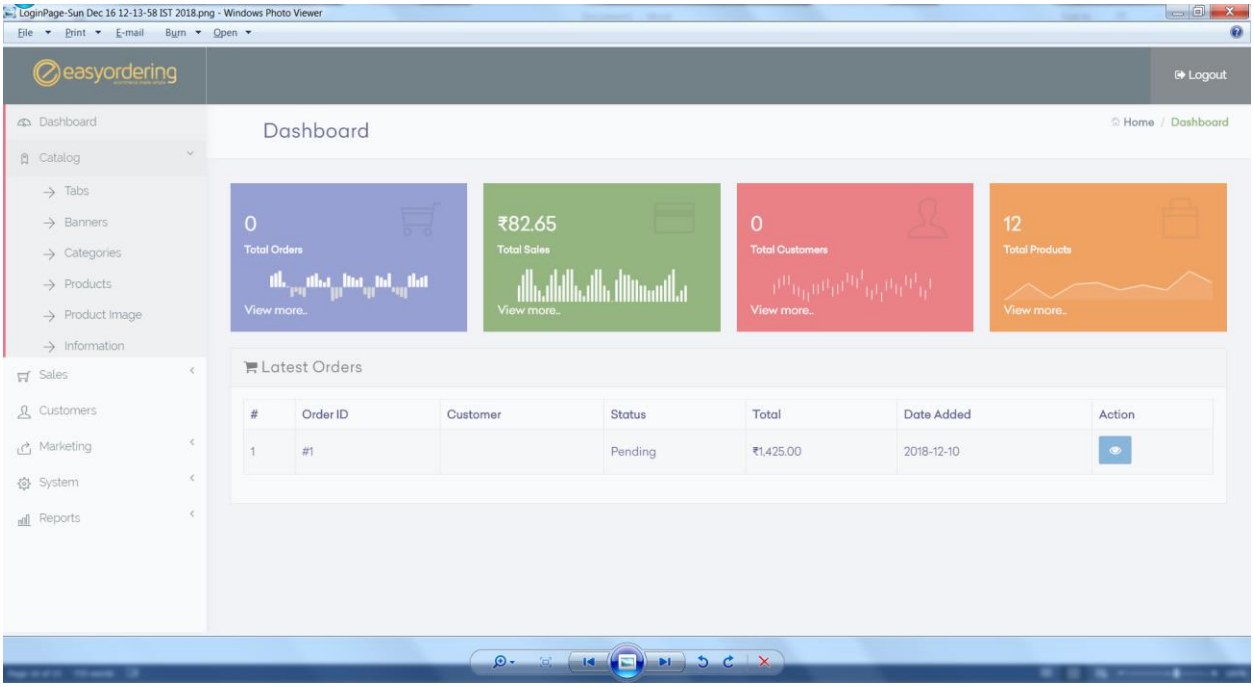
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite" parallel="tests">
	<test thread-count="5" name="Test">
		<classes>
			<class name="com.ibm.test.BaseTest">
			</class>
		</classes>
	</test> <!-- Test -->
</suite> <!-- Suite -->
```

Output screen shot

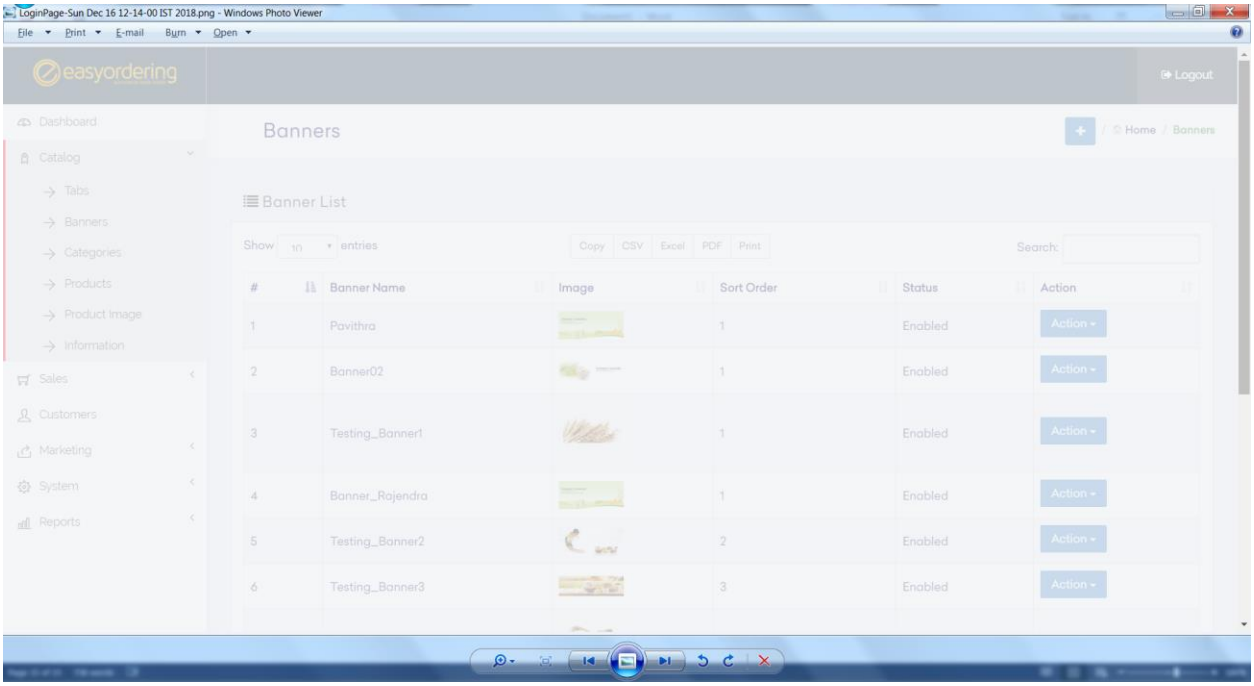
Login screen



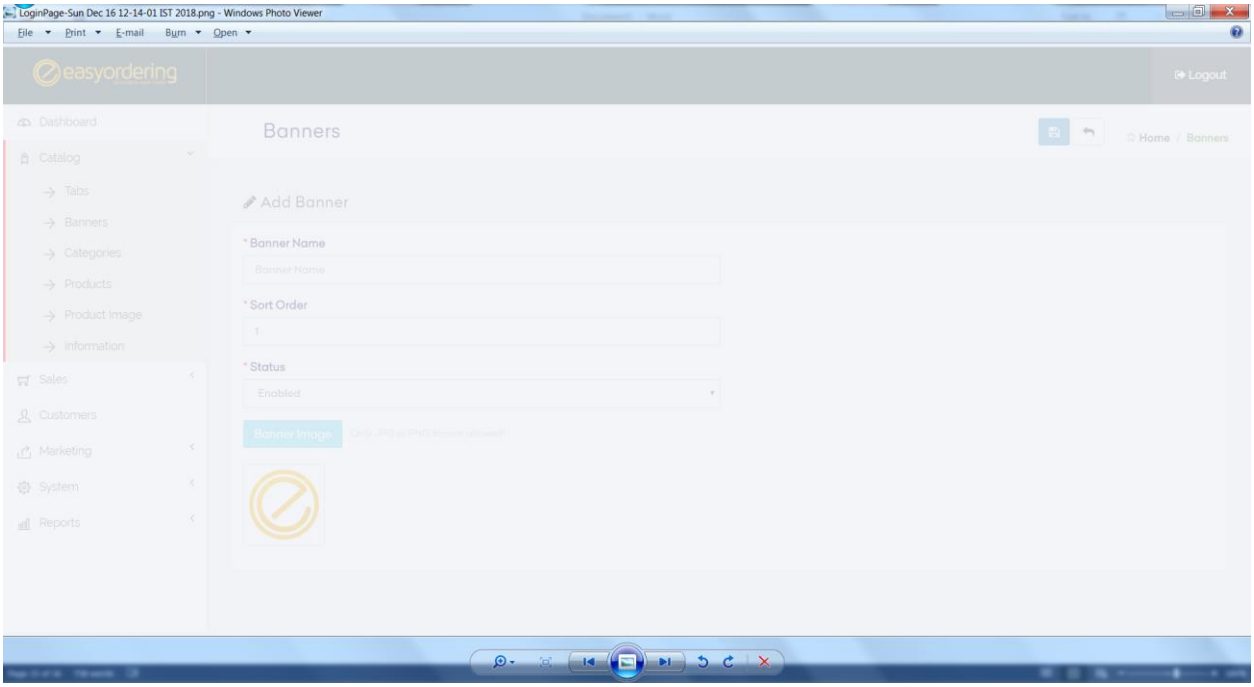
DashBoard



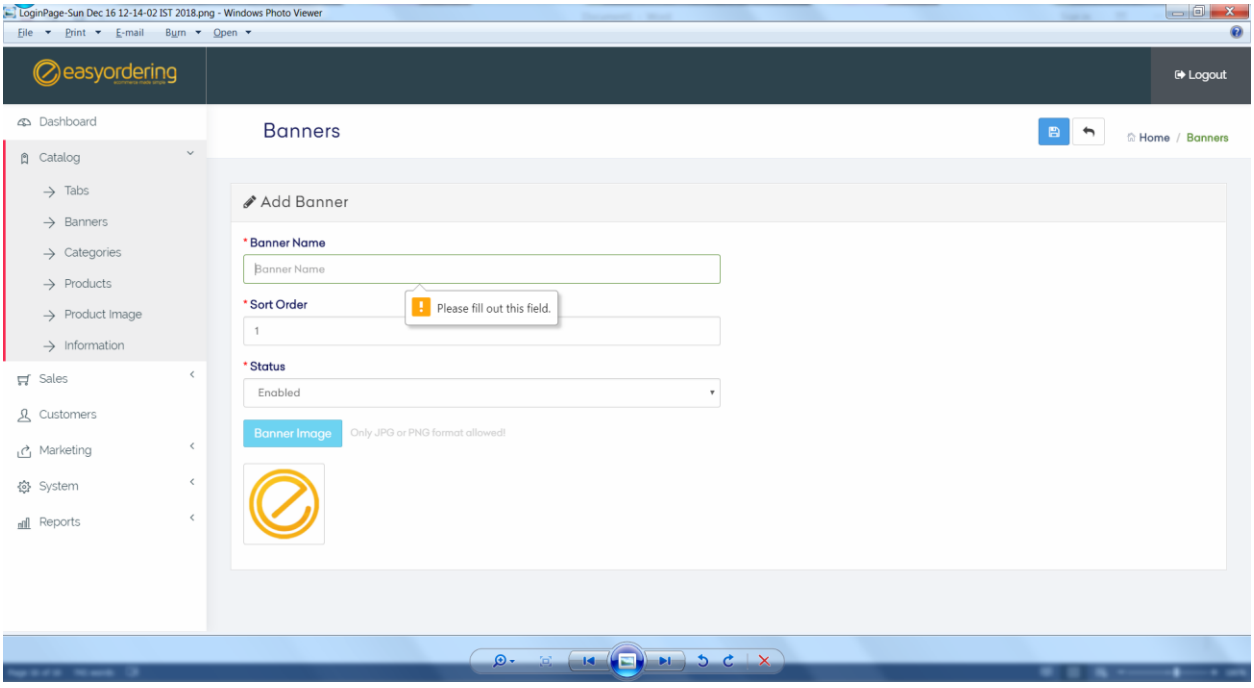
Banners page



Add banner screen

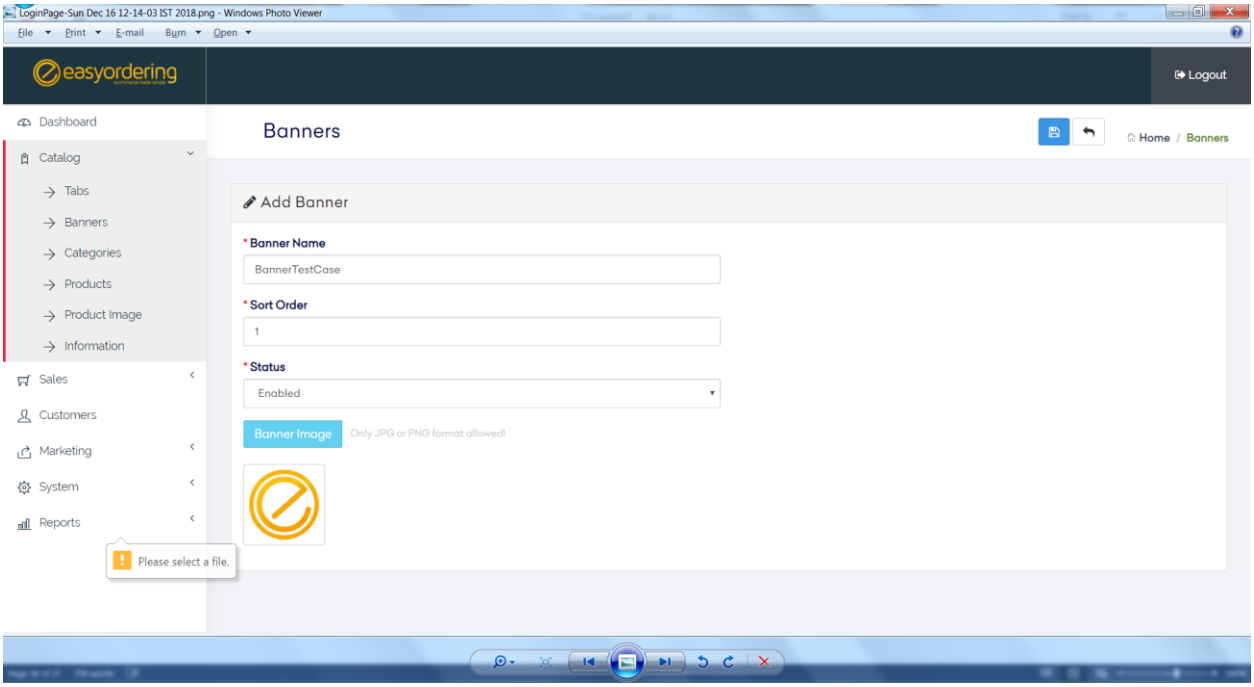


Click on save without entering details



Verified page on same page (without entering details and click on save)

Enter banner Name and click on save



Still image not uploaded and giving error to save banner..

Negative validation is working as expected

Find Emailable Output of above program

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
Test	1	0	0	13,827		

Class	Method	Start	Time (ms)
Suite			
Test — passed			
com.ibm.test.BaseTest	NegativeTestAddBanner	1544942634101	9595

Test

com.ibm.test.BaseTest#NegativeTestAddBanner

back to summary

Test Case pass