

Day 11 - Test case solution document

Code Used:

BaseTest.java

```
package com.ibm.test;
import java.io.File;
import java.io.IOException;
import java.sql.SQLException;
import java.util.Date;
import org.apache.commons.io.FileUtils;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Optional;
import org.testng.annotations.Parameters;
import org.testng.annotations.Test;
import com.ibm.pages.CategoryPage;
import com.ibm.pages.CouponPage;
import com.ibm.pages.DashboardPage;
import com.ibm.pages.DriverLaunch;
import com.ibm.pages.LoginPage;
import com.ibm.pages.ProductsPage;
import com.ibm.pages>ReturnsPage;
import com.ibm.pages.SettingsPage;
import com.ibm.pages.ShippingPage;
import com.ibm.pages.TabsPage;
import com.ibm.pages.UserPage;
import com.ibm.utilities.DBConnection;
import com.ibm.utilities.ExcelReader;

public class BaseTest extends DriverLaunch
{
    @Test(testName="AddingNewCategory_TC11",dataProvider="AddCategory")
    public void Day11TestCase(String CategoryName,String TagTitle,String
SortOrder,String Status,String Image,String ValidMsg,String SuccessMsg) throws
InterruptedException, IOException, SQLException
    {
        LoginPage login=new LoginPage(driver,wait);
        DashboardPage dash=new DashboardPage(driver,wait);
        CategoryPage category=new CategoryPage(driver,wait);
        UserPage user=new UserPage(driver,wait);
        DBConnection dbconn=new DBConnection();
        try
        {
            //getting the data from the properties file using Hash map
            String adminurl=hashData.get("adminurl");
            String username=hashData.get("username");
```

```

        String password=hashData.get("password");
        String userurl=hashData.get("userurl");
        String showdata=hashData.get("show");
        //String cateImg=hashData.get("image");

        //navigating to the Admin URL of a to z groceries
        driver.get(adminurl);

        //login page is used to provide the input of username and
password

        login.EnterEmail(username);
        login.EnterPassword(password);
        login.clickLogin();

        //In catalog page we will find the catalog element and click on
that to open categories
        dash.OpenCatalog();

        //category page
        category.clickCategory();

        //Sending input for the new category

        category.addNewCategory(CategoryName,TagTitle,SortOrder,Status,Image,ValidMsg)
;

        //validating Success Message
        category.validateSuccessMsg(SuccessMsg);

        //Selecting to show All in categories
        //category.displayAllCategories(showdata);

        //validating the newly added category
        category.ValidateCategory(CategoryName);
        login.ClickLogout();

        //Navigating to User Page
        driver.get(userurl);
        //Validating new category in User Page
        user.CatergoryValidation(CategoryName);

        //Validating new category in Database
        dbconn.DBCategoryVerify(CategoryName);
    }
    catch(Exception e)
    {
        TakesScreenshot ts = (TakesScreenshot) driver;
        File file = ts.getScreenshotAs(OutputType.FILE);
        Date date = new Date();
        String currentDate = date.toString().replace(":", "-");
        FileUtils.copyFile(file, new
File("./Screenshots/Day11_TestCase_Error_" + currentDate + ".jpg"));
        System.out.println(e.getMessage());
        Assert.fail();
    }
}

```

```

        }
        finally {
            driver.quit();
        }
    }
    @DataProvider(name="AddCategory")
    public Object[][] AddCategory() throws IOException
    {
        return ExcelReader.DataTable("./TestData/TestData.xlsx", "AddCategory");
    }
}

```

LoginPage.java

```

package com.ibm.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class LoginPage {
    @FindBy(name="email")
    WebElement emailEle;

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

    @FindBy(xpath="/html/body/div/div/div[2]/form/button")
    WebElement loginEle;

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

    WebDriverWait wait;
    WebDriver driver;

    public LoginPage(WebDriver driver, WebDriverWait wait)
    {
        PageFactory.initElements(driver, this);
    }
    public void EnterEmail(String username)
    {
        emailEle.sendKeys(username);
    }
    public void EnterPassword(String password)
    {
        passwordEle.sendKeys(password);
    }
    public void clickLogin()
    {
        loginEle.click();
    }
}

```

```

        }
    public void ClickLogout()
    {
        logoutEle.click();
    }
}

```

DashboardPage.java

```

package com.ibm.pages;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class DashboardPage {

    //Finding the WebElements
    @FindBy(partialLinkText="Catalog")
    WebElement catalogEle;

    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[5]/a")
    WebElement marketEle;

    @FindBy(xpath="//*[contains(text(),'System')]")
    WebElement systemEle;

    @FindBy(xpath="//*[contains(text(),'Shipping')]")
    WebElement shipEle;

    WebDriver driver;
    WebDriverWait wait;

    //catalog page constructor
    public DashboardPage(WebDriver driver, WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
        PageFactory.initElements(driver, this);
    }

    //Opencatalog method to click on catalog
    public void OpenCatalog()
    {
        wait.until(ExpectedConditions.visibilityOf(catalogEle));
        catalogEle.click();
    }

    //method for opening marketing
    public void OpenMarketing()
    {

```

```

        marketEle.click();
    }
    //method for opening system
    public void OpenSystem()
    {
        systemEle.click();
    }
    public void OpenShipping()
    {
        shipEle.click();
    }
}

CategoryPage.java
package com.ibm.pages;

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.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.Reporter;

public class CategoryPage
{
    //Finding the Elements
    @FindBy(xpath="/html/body/div[1]/div[1]/div/ul/li[2]/ul/li[3]/a")
    WebElement categoryEle;

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

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

    @FindBy(name="tag_title")
    WebElement tagTitleEle;

    @FindBy(name="sort")
    WebElement sortEle;

    @FindBy(name="status")
    WebElement statusEle;

    @FindBy(xpath="/*[@id='files']")
    WebElement imageEle;

    @FindBy(name="dataTableExample2_length")

```

```

WebElement showEle;

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

@FindBy(xpath="/html/body/div[1]/div[3]/div/form/div[2]/div/div/div[1]/div/h4")
)
WebElement addCateEle;

WebDriver driver;
WebDriverWait wait;
JavascriptExecutor js;

//Constructor
public CategoryPage(WebDriver driver,WebDriverWait wait)
{
    this.driver=driver;
    PageFactory.initElements(driver, this);
    js=(JavascriptExecutor) driver;
}

//this method is used to click on the Categories under the catalog
public void clickCategory()
{
    categoryEle.click();
}

// This method is used to add New Category to the catalog
public void addNewCategory(String name,String tagtitle,String sortorder,String
status,String img,String ReqMsg)
{
    //clicking on + sign for adding new category
    addNewEle.click();
    //providing the new category details
    cateNameEle.click();
    cateNameEle.sendKeys(name);
    //Validating Error Message after sending Category Name and clicking on
Save
    saveEle.click();
    String TitleReq = js.executeScript("return
document.getElementsByName('tag_title')[0].validationMessage").toString().trim();
    Assert.assertTrue(TitleReq.contains(ReqMsg), "Assertion on Category Tag
Title as mandatory field!!");
    Reporter.log("validation on Error Message - Category Tag Title as
mandatory field!!");
    //Sending data for other fields after validating Error message
    tagTitleEle.sendKeys(tagtitle);
    js.executeScript("arguments[0].scrollIntoView(true)",imageEle);
    sortEle.sendKeys(sortorder);
    Select selectStatus= new Select(statusEle);
    selectStatus.selectByVisibleText(status);
    //Attaching Image and clicking on Save
    imageEle.sendKeys(img);
    driver.findElement(By.id("toTop")).click();
}

```

```

        saveEle.click();

    }
    public String getCategoriesTitle()
    {
        String title= driver.getTitle();
        return title;
    }
    // After saving this will redirect to categories page.

    //showSel will select "ALL" items or display
    public void displayAllCategories(String showvalue)
    {
        Select showSel=new Select(showEle);
        showSel.selectByVisibleText(showvalue);
        //scrolling the window to see the products
        js.executeScript("window.scrollBy(0,500)");
        js.executeScript("window.scrollBy(500,1000)");
        js.executeScript("window.scrollBy(1000,2000)");
    }
    //this method will validate the Success Message
    public void validateSuccessMsg(String success)
    {

        //wait.until(ExpectedConditions.presenceOfElementLocated(By.xpath("//div[@clas
s='alert alert-success alert-dismissible']")));
        String message=js.executeScript("return
document.getElementsByClassName('alert alert-success alert-
dismissible')[0].textContent").toString().trim();
        Assert.assertTrue(message.contains(success), "Assertion on Success
Message after adding new Category!!!");
        Reporter.log("Validation of Success Message after adding new
Category!!!");
    }
    public String dataPageSource()
    {
        String dataPgSrc=driver.getPageSource();
        return dataPgSrc;

    }
    //this method will edit a category after clearing the existing data
    public void editCategory(String old_name,String new_name,String
tagtitle,String sort,String status,String imgPath)
    {
        List<WebElement>
rowEle=driver.findElements(By.xpath("//table[@id='dataTableExample2']/tbody/tr"));
        int rows=rowEle.size();
        for(int i=1;i<=rows;i++)
        {
            //getting the category name to edit
            String
name=driver.findElement(By.xpath("//table[@id='dataTableExample2']/tbody/tr["+i+"]/td
[2])).getText().trim();

            if(name.equals(old_name))

```



```

        if(nameTxt.equals(catName))
        {
            Assert.assertTrue(nameTxt.contains(catName), "Assertion of
newly added Category presence in Admin Page categories Table!!");
            Reporter.log("Validation of newly added Category in Admin
Panel Table!!");
            break;
        }
    }
}

```

}

UserPage.java

```

package com.ibm.pages;

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import org.testng.Assert;
import org.testng.Reporter;

public class UserPage
{
    //Finding the elements of the User Page
    @FindBy(xpath="//*[@class='click-categories flaticon-bars'][2]")
    WebElement catBarEle;
    @FindBy(xpath="//*[@class='category-drop-list-inner'][2]")
    WebElement CateList;
    @FindBy(xpath="/html/body/div[1]/div[5]/div/div[1]/div/nav/ul/li[2]/span")
    WebElement HdrEle;
    @FindBy(xpath="//input[@id='search-box']")
    WebElement mainSrchEle;

    WebDriver driver;
    WebDriverWait wait;
    JavascriptExecutor js;
    //Constructor
    public UserPage(WebDriver driver,WebDriverWait wait)
    {
        this.driver=driver;
        this.wait=wait;
        js=(JavascriptExecutor) driver;
        PageFactory.initElements(driver, this);
    }
    //this method validates the category presence in User Page under "Shop By
Category"
    public void CatergoryValidation(String catename)

```

```

{
    //clicking on the bars of Shop by category
    catBarEle.click();
    wait.until(ExpectedConditions.visibilityOf(CateList));
    //opening the category created
    WebElement newcateEle=driver.findElement(By.partialLinkText(catename));
    newcateEle.click();
    //passing the header text to a string for validation
    String validCate=HdrEle.getText().trim();
    Assert.assertTrue(validCate.contains(catename), "Assertion on new
category in User Page!!!");
    Reporter.log("Validation of Category header in User Page!!!");
}
//This method validates the edited Phone number present in the footer
information in User Page
public void footerPhnoValidation(String Phno) throws InterruptedException
{
    WebElement
footPhno=driver.findElement(By.xpath("//p[contains(text(), '"+Phno+"')]"));
    js.executeScript("arguments[0].scrollIntoView(true);", footPhno);
    String footerPh=footPhno.getText().trim();
    Assert.assertTrue(footerPh.contains(Phno), "Assertion on Phone number in
footer section!!!");
    Reporter.log("Assertion on Phone number in footer section!!!");
}
//This method validates the edited Phone number present in the header
information in User Page
public void headerPhnoValidation(String Phno) throws InterruptedException
{
    WebElement hdrPhno=driver.findElement(By.xpath("(//div[@class='header-
mid-right-content'])[1]"));
    js.executeScript("arguments[0].scrollIntoView(true);", hdrPhno);
    String headerPh=hdrPhno.getText().trim();
    Assert.assertTrue(headerPh.contains(Phno), "Assertion on Phone number in
header section!!!");
    Reporter.log("Assertion on Phone number in header section!!!");
}
//this method validates the edited Product in User page
public void ProductsValidation(String ProNewName) throws InterruptedException
{

    mainSrchEle.click();
    Thread.sleep(10000);
    //sending keys in the search

    driver.findElement(By.xpath("//*[@id=\"searchModal\"]//div/div/div/div[1]/input
")).sendKeys(ProNewName);
    Thread.sleep(5000);
    //selecting the prooduct from the dropdown

    driver.findElement(By.xpath("//span[text()='"+ProNewName+"']/ancestor::div[@id
='searchproducts-div']")).click();
    String validProduct=HdrEle.getText().trim();
    Assert.assertTrue(validProduct.contains(ProNewName), "Assertion on
Edited Product in User Page!!!");
}

```

```

        Reporter.log("Assertion on Edited Product in User Page!!");
    }
}

```

DBConnection.java

```

package com.ibm.utilities;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

import org.testng.Assert;
import org.testng.Reporter;

public class DBConnection {

    //method to validate category in Database
    public void DBCategoryVerify(String newCatName) throws SQLException
    {
        //Establishing the DB connection
        Connection c =
DriverManager.getConnection("jdbc:mysql://foodsonfinger.com:3306/foodsonfinger_atozgr
oceries",
                "foodsonfinger_atoz","welcome@123");
        Statement s = c.createStatement();
        //Executing the required statement for getting the new category..
        sending the new catergory name in where condition
        ResultSet rs = s.executeQuery("SELECT * from as_category where
name=\""+newCatName+"\"");
        //the below while loop checks each and every line of the above query
        result
        while(rs.next())
        {
            //doing assertion on New category name when matches the name of
            the category in DB

            Assert.assertTrue(newCatName.equals(rs.getString("name")),"Assertion on New
Category in Database!!");
            Reporter.Log("Assertion on New Category in Database!!");

        }
    }
}

```

DriverLaunch.java

```

package com.ibm.pages;

import java.io.IOException;
import java.util.HashMap;
import java.util.concurrent.TimeUnit;

import org.openqa.selenium.WebDriver;

```

```

import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
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.Optional;
import org.testng.annotations.Parameters;

import com.ibm.utilities.PropertiesFileHandler;

public class DriverLaunch {

    public WebDriver driver;
    public WebDriverWait wait;
    public PropertiesFileHandler propFileHandle;
    public HashMap<String, String> hashData;

    @BeforeSuite
    public void preSetForTest() throws IOException
    {
        String file=".//TestData/data.properties";
        propFileHandle =new PropertiesFileHandler();
        hashData=propFileHandle.getPropertiesAsMap(file);
    }
    @BeforeMethod
    @Parameters({"browser"})
    public void initialisation(@Optional("ch")String browser)
    {
        browserInitialization(browser);
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(30, TimeUnit.SECONDS);
        wait=new WebDriverWait(driver, 60);
    }
    @AfterMethod
    public void closingBrowser()
    {
        driver.quit();
    }

    public void browserInitialization(String browser)
    {
        switch(browser.toLowerCase())
        {
            case "ff":
                System.setProperty("webdriver.gecko.driver",
".//drivers/geckodriver.exe");
                driver=new FirefoxDriver();
                break;
            case "ch":
                System.setProperty("webdriver.chrome.driver",
".//drivers/chromedriver.exe");
                driver=new ChromeDriver();
        }
    }
}

```

```

        break;
    case "ie":
        System.setProperty("webdriver.ie.driver",
        "./drivers/IEDriverServer.exe");
        driver=new InternetExplorerDriver();
        break;
    }
}
}

```

PropertiesFileHandler.java

```

package com.ibm.utilities;

import java.io.FileInputStream;
import java.io.IOException;
import java.util.HashMap;
import java.util.Properties;
import java.util.Set;

public class PropertiesFileHandler {

    public HashMap<String, String> getPropertiesAsMap(String file) throws
IOException {
        HashMap<String, String> magentoMap = new HashMap<String, String>();

        FileInputStream fileIn = new FileInputStream(file);
        Properties prop = new Properties();
        prop.load(fileIn);

        Set<Object> keysProp = prop.keySet();
        for (Object key : keysProp) {
            magentoMap.put(key.toString(), prop.getProperty(key.toString()));
        }
        prop.clear();
        return magentoMap;
    }
}

```

data.properties

```

adminurl=https://atozgroceries.com/admin
userurl=https://atozgroceries.com
username=demo@atozgroceries.com
password=456789

```

testng.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test thread-count="5" name="Test">
        <parameter name="browser" value="ff"></parameter>
        <classes>
            <class name="com.ibm.test.BaseTest">
                <methods><include name="Day10TestCase"></include></methods>
            </class>
        
```

```

</classes>
</test> <!-- Test -->
</suite> <!-- Suite -->

ExcelReader.java
package com.ibm.utilities;

import java.io.FileInputStream;
import java.io.IOException;

import org.apache.poi.ss.usermodel.DataFormatter;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;

public class ExcelReader {
    public static Object[][] DataTable(String WBLoc, String sheetName) throws
IOException
    {
        XSSWorkbook TestDataWB=new XSSWorkbook(new FileInputStream(WBLoc));
        XSSSheet ProductsSheet=TestDataWB.getSheet(sheetName);
        Object[][] ProductInfo=new Object[ProductsSheet.getPhysicalNumberOfRows()-
1][ProductsSheet.getRow(0).getPhysicalNumberOfCells()];
        int rowcount=ProductsSheet.getPhysicalNumberOfRows();
        int cellcount=ProductsSheet.getRow(0).getPhysicalNumberOfCells();
        for(int i=1;i<rowcount;i++)
        {
            for(int j=0;j<cellcount;j++)
            {
                DataFormatter format=new DataFormatter();
                String
cellvalue=format.formatCellValue(ProductsSheet.getRow(i).getCell(j));
                ProductInfo[i-1][j]=cellvalue;
            }
        }
        return ProductInfo;
    }
}

```

TestData.xlsx

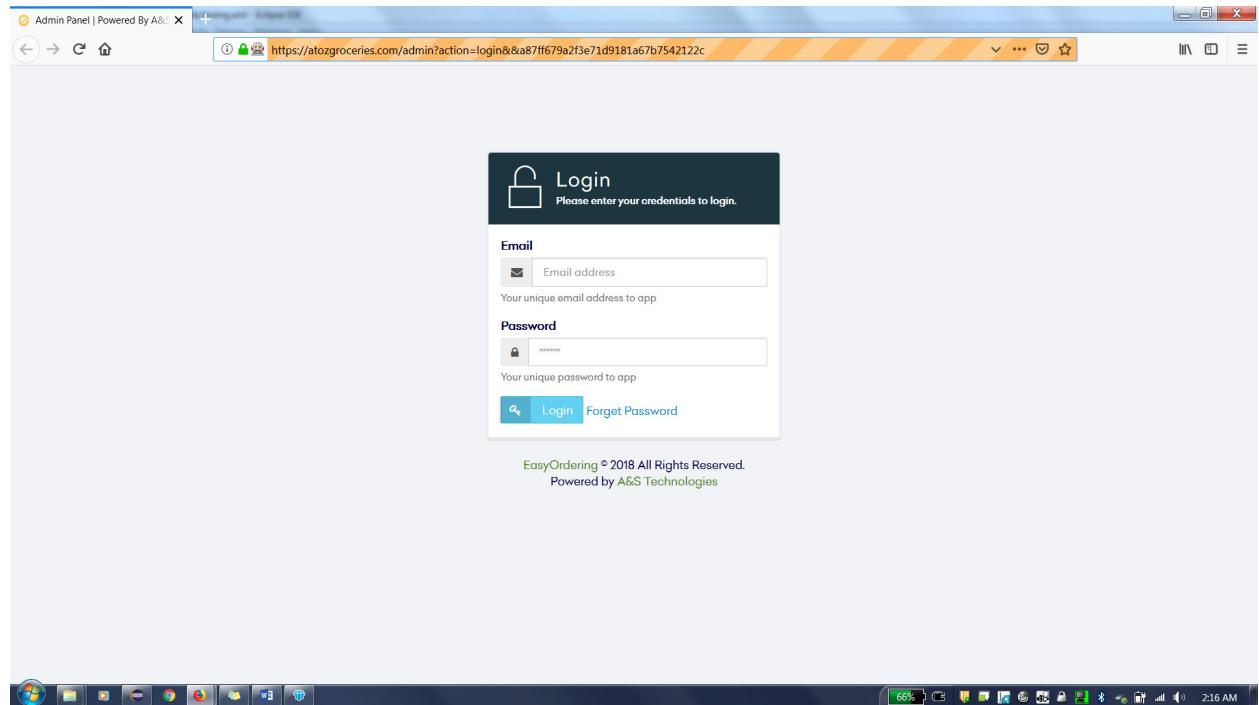
The screenshot shows the Eclipse IDE interface with the title "EclipseWorkSpace - SeleniumFramework/TestData/TestData.xlsx - Eclipse IDE". The central area displays an Excel spreadsheet titled "TestData.xlsx" with the following data:

	A	B	C	D	E	F	G	H
1	CategoryName	TagTitle	SortOrder	Status	Image	ValidMsg	SuccessMsg	
2	Noodles	Noodles	3	Enabled	C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\TestData\noodle_category.jpg	fill out this field	successfully added category	
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								
26								

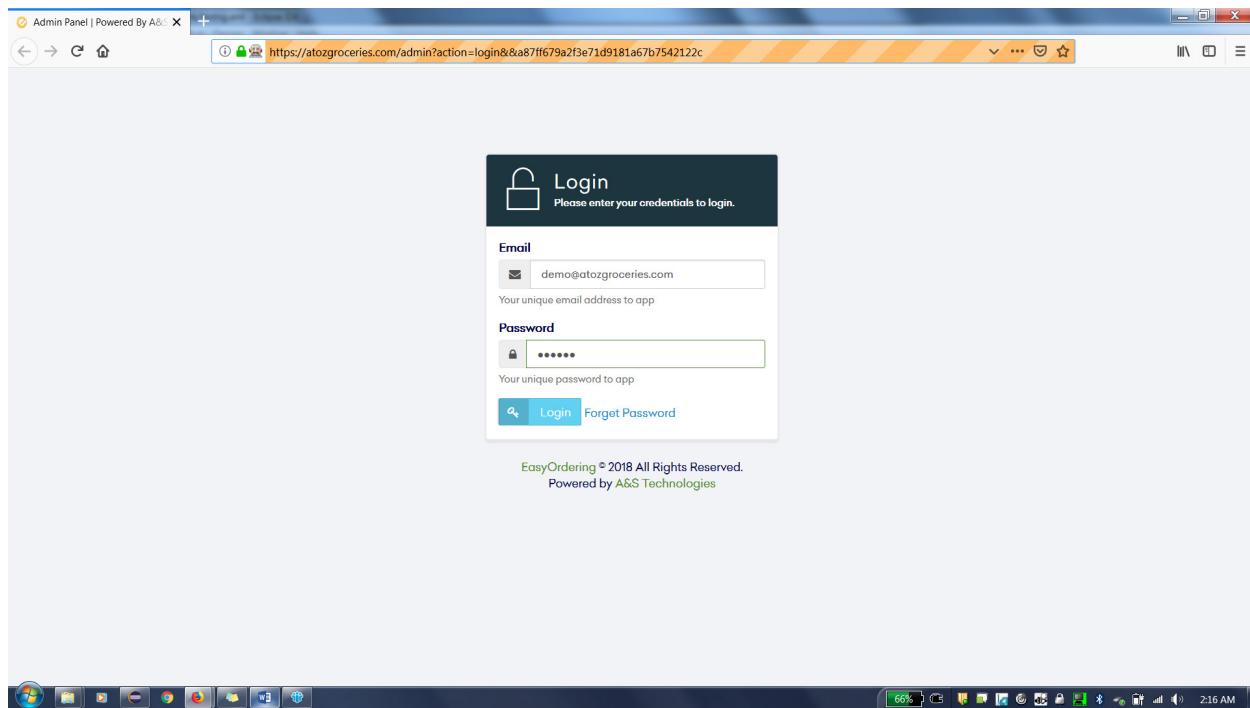
The left sidebar shows the project structure with "Test Data" selected, containing files like "almonds.jpg", "Bas_rice.jpg", etc. The bottom status bar shows "81%" and the time "2:38 AM".

Step by step output:

1. Navigating to the Admin Page of “atozgroceries”



2. Providing login credentials



3. Dashboard after logging into the site

A screenshot of the Admin Panel dashboard. The URL is https://atozgroceries.com/admin/dashboard?token=8f14e45fcee167a5a36dedd4bea2543. The dashboard has a sidebar with 'Dashboard', 'Catalog', 'Sales', 'Customers', 'Marketing', 'System', and 'Reports'. The main area shows four cards: 'Total Orders' (0), 'Total Sales' (₹4,138.82), 'Total Customers' (4), and 'Total Products' (33). Below is a section titled 'Latest Orders' with a table:

#	Order ID	Customer	Status	Total	Date Added	Action
1	#10	Rajendra Sahoo	Pending	₹21,750.00	2018-12-21	
2	#6	TestNew	Pending	₹3,950.00	2018-12-20	
3	#5		Pending	₹8,700.00	2018-12-20	
4	#3		Pending	₹3,570.00	2018-12-18	
5	#2		Pending	₹3,570.00	2018-12-18	

At the bottom, the taskbar shows various icons.

4. Opening Catalog

The screenshot shows the Admin Panel dashboard for easyordering. On the left, a sidebar menu is open under the 'Catalog' section, listing 'Tabs', 'Banners', 'Categories', 'Products', 'Product Image', and 'Information'. The main content area is titled 'Dashboard' and features four large cards: 'Total Orders' (6), 'Total Sales' (₹27,965.00), 'Total Customers' (32), and 'Total Products' (230). Below these cards is a table titled 'Latest Orders' with columns for #, Order ID, Customer, Status, Total, Date Added, and Action. The table contains five rows of order data.

#	Order ID	Customer	Status	Total	Date Added	Action
1	#10	Rajendra Sahoo	Pending	₹21,750.00	2018-12-21	
2	#6	TestNew	Pending	₹3,950.00	2018-12-20	
3	#5		Pending	₹8,700.00	2018-12-20	
4	#3		Pending	₹3,570.00	2018-12-18	
5	#2		Pending	₹3,570.00	2018-12-18	

5. Opening Categories

The screenshot shows the 'Categories' page in the Admin Panel. The sidebar menu is open under the 'Catalog' section, listing 'Tabs', 'Banners', 'Categories', 'Products', 'Product Image', and 'Information'. The main content area is titled 'Category List' and displays a table of categories. Each category row includes the category name, an image thumbnail, sort order, status (Enabled or Disabled), and an 'Action' button. The categories listed are: test15Category, Pulses Items123, Fruits, Grains, Vegetables, and Dry Red Chilli.

#	Category Name	Image	Sort Order	Status	Action
1	test15Category		0	Enabled	
2	Pulses Items123		1	Disabled	
3	Fruits		2	Enabled	
4	Grains		3	Enabled	
5	Vegetables		4	Enabled	
6	Dry Red Chilli		7	Enabled	

6. Clicking on “+” sign to add new category

The screenshot shows the 'Categories' section of the easyordering Admin Panel. On the left, there's a sidebar with links like Dashboard, Catalog (which is selected), Tabs, Banners, Categories, Products, Product Image, Information, Sales, Customers, Marketing, System, and Reports. The main area has a title 'Categories' and a sub-section 'Add Category'. It contains fields for 'Category Name' (with placeholder 'Category Name'), 'Parent' (a dropdown menu), and 'Description' (a rich text editor). At the bottom of the 'Add Category' form, there's a 'Save' button.

7. Sending Category Name and Clicking on Save to validate Error message

This screenshot is similar to the previous one but shows an error. The 'Category Name' field now contains 'Noodles', and a red error message 'Please fill out this field.' appears below it. The rest of the form and sidebar are identical to the first screenshot.

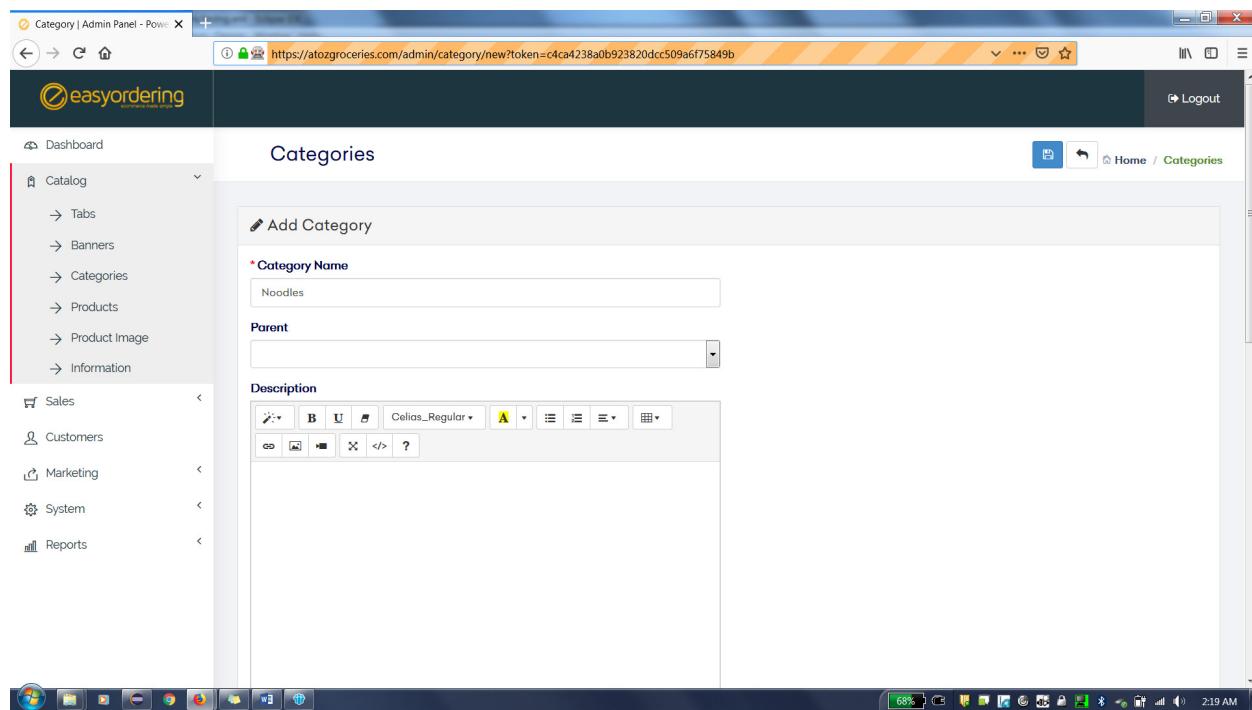
8. After validation, sending all the required data for adding a category

The screenshot shows the 'Category | Admin Panel - Pow' window. On the left, a sidebar lists navigation items: Banners, Categories (which is selected and highlighted in red), Products, Product Image, Information, Sales, Customers, Marketing, System, and Reports. The main content area contains fields for 'Category Name' (Noodles), 'Parent' (a dropdown menu), 'Description' (with a rich text editor toolbar), and 'Meta Tag Title' (Noodles). The status bar at the bottom shows system icons and the time 2:28 AM.

9. Attaching an image file

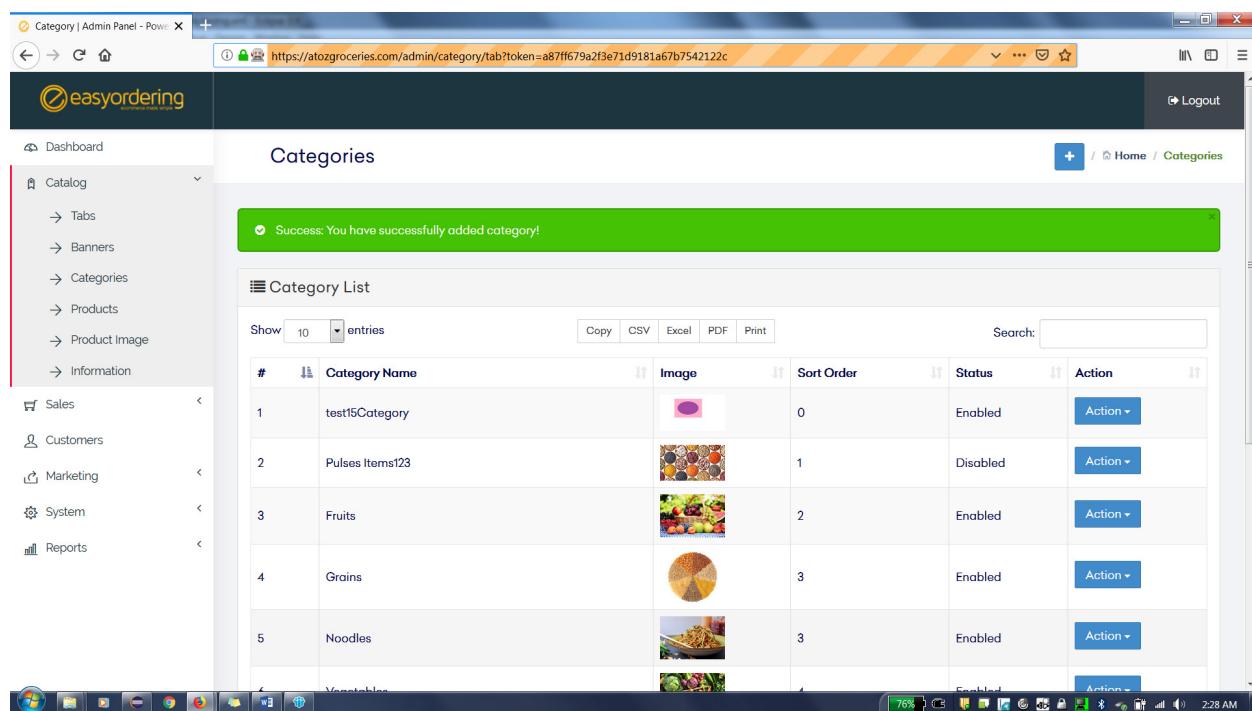
The screenshot shows the same 'Category | Admin Panel - Pow' window. In addition to the previous fields, it now includes 'Meta Tag Keywords' (Meta Tag Keywords) and 'Sort Order' (3). A 'Status' dropdown is set to 'Enabled'. A 'Category Image' field is present, with a thumbnail preview of a bowl of noodles and the note 'Only JPG or PNG format allowed!'. The status bar at the bottom shows system icons and the time 2:28 AM.

10. Clicking on Save.



The screenshot shows the 'Categories' section of the easyordering Admin Panel. On the left, there's a sidebar with links like Dashboard, Catalog (selected), Sales, Customers, Marketing, System, and Reports. The main content area has a title 'Categories' and a sub-section 'Add Category'. It includes fields for 'Category Name' (containing 'Noodles'), 'Parent' (empty dropdown), and 'Description' (rich text editor). The URL in the browser is <https://atozgroceries.com/admin/category/new?token=c4ca4238a0b923820dcc509a6f75849b>.

11. Validating the Success message of adding category



The screenshot shows the 'Categories' section of the easyordering Admin Panel. The sidebar is identical to the previous screenshot. The main content area now displays a 'Category List' table with 5 rows. The table has columns for #, Category Name, Image, Sort Order, Status, and Action. The categories listed are: test15Category (Image: pink square, Sort Order: 0, Status: Enabled), Pulses Items123 (Image: bowl of cereal, Sort Order: 1, Status: Disabled), Fruits (Image: various fruits, Sort Order: 2, Status: Enabled), Grains (Image: pie chart of grains, Sort Order: 3, Status: Enabled), and Noodles (Image: bowl of noodles, Sort Order: 3, Status: Enabled). Above the table, a green success message box says 'Success: You have successfully added category!'. The URL in the browser is <https://atozgroceries.com/admin/category/tab?token=a87ff679a2f3e71d9181a67b7542122c>.

12. Selecting "ALL" from Show, to display all categories

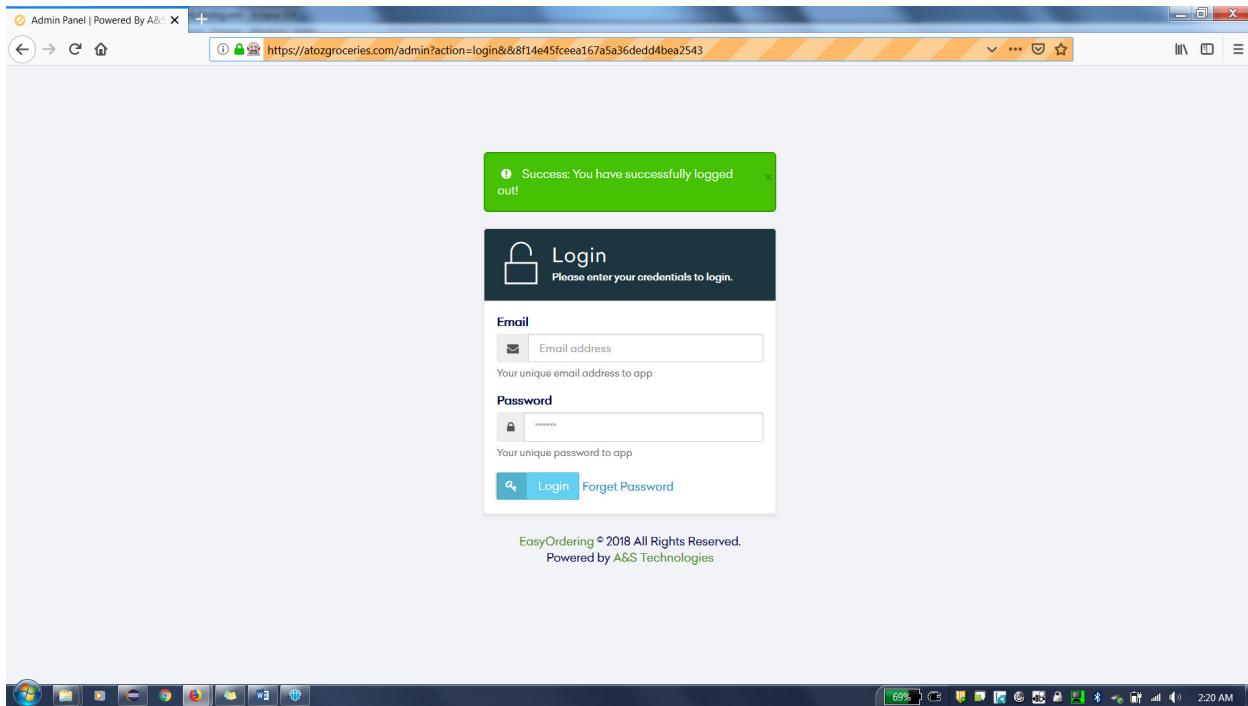
Screenshot of the Admin Panel showing the Categories page. A success message "Success: You have successfully added category!" is displayed. The Category List table shows five categories: test15Category, Pulses Items123, Fruits, Grains, and Noodles. The "Show" dropdown is set to "All".

#	Category Name	Image	Sort Order	Status	Action
1	test15Category		0	Enabled	Action ▾
2	Pulses Items123		1	Disabled	Action ▾
3	Fruits		2	Enabled	Action ▾
4	Grains		3	Enabled	Action ▾
5	Noodles		3	Enabled	Action ▾

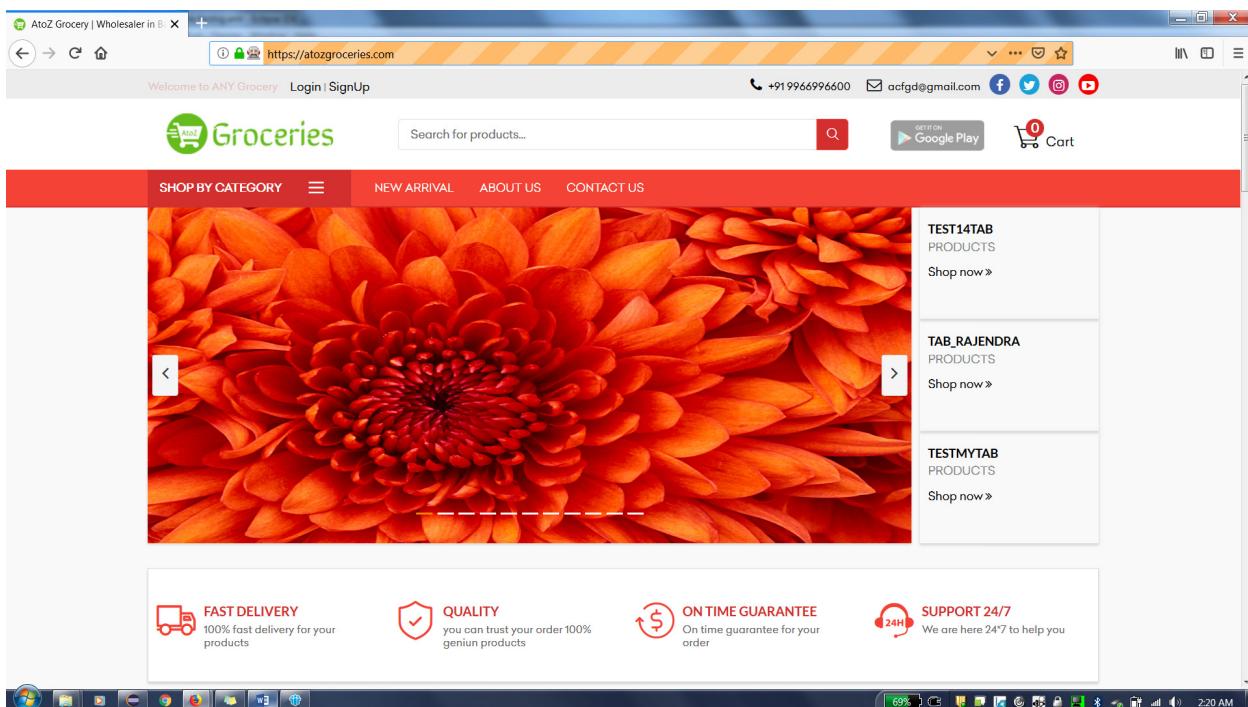
Screenshot of the Admin Panel showing the Categories page. The Category List table displays a list of categories including Vegetables, Soveiries(Mixtures) & Sweets, Masala Items, Papad all types, Vermicel,maggi all fast cookings, Masala , Powders& Sauce, and Colours(Edible). The "Show" dropdown is set to "All".

#	Category Name	Image	Sort Order	Status	Action
27	Vegetables		22	Enabled	Action ▾
28	Soveiries(Mixtures) & Sweets		23	Enabled	Action ▾
29	Masala Items		24	Enabled	Action ▾
30	Papad all types		25	Enabled	Action ▾
31	Vermicel,maggi all fast cookings		25	Enabled	Action ▾
32	Masala , Powders& Sauce		27	Enabled	Action ▾
33	Colours(Edible)		28	Enabled	Action ▾

13. Logging out from Admin Page



14. Navigating to User Page



15. Opening Categories under Shop By Category and selecting newly added category

The screenshot shows the homepage of the AtoZ Grocery website. On the left, there is a sidebar menu with categories: Fruits, Grains, Noodles, Vegetables, Dry Red Chilli, Coriander/Daniya, and Xxx. Below this is a "More Categories" link. The main content area features a large image of various grains and legumes in bowls on a wooden surface. To the right of the image, there are three promotional boxes: "TEST14TAB PRODUCTS" (Shop now), "TAB_RAJENDRA PRODUCTS" (Shop now), and "TESTMYTAB PRODUCTS" (Shop now). At the bottom of the page, there are four service icons: FAST DELIVERY (100% fast delivery for your products), QUALITY (you can trust your order 100% genuine products), ON TIME GUARANTEE (On time guarantee for your order), and SUPPORT 24/7 (We are here 24/7 to help you).

16. Validating the new category header

The screenshot shows the 'Category - Noodles' page. The sidebar menu on the left has 'Noodles' selected. The main content area displays a grid of five product cards with the following details:

test14tab	Tab_Rajendra	TestMyTab	Test01	Test02
Test03	Test04	Test055		

17. Output from console

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows various Java files and resources, including DashboardPage.java, LoginPage.java, ProductsPage.java, ReturnsPage.java, SettingsPage.java, ShippingPage.java, TabsPage.java, and UserPage.java.
- TestNG XML Configuration:** The `testng.xml` file contains the following XML code:


```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd">
<suite name="Suite">
<test thread-count="5" name="test">
<parameter name="browser" value="ff"/>
<classes>
<class name="com.ibm.test.BaseTest">
<methods><include name="Day11TestCase"/></include></methods>
</class>
</classes>
</test>
</suite>
      
```
- Console Output:** The console window displays the output of the test execution, including error messages and test results. Key errors include:
 - JavaScript error: `TypeError: document.getElementById(...) is null`
 - Java error: `SeleniumFramework.Testing` terminated due to a Java exception.
 - Multiple pipe errors (e.g., `pipe error: 109`) from child threads.
 - MessageChannel errors: `Channel closing: too late to send/recv, messages will be lost` and `Channel error: cannot send/recv`.
- Output Summary:** The console shows the total test run statistics: `Total tests run: 1, Failures: 0, Skips: 0`.

18. Automated Index.html report

The screenshot shows a web browser displaying the `index.html` report generated by TestNG. The report includes:

- Header:** TestNG reports
- Navigation:** Back, Forward, Stop, Home, Address bar (file:///C:/Harisha/Selenium_Training/EclipseWorkSpace/SeleniumFramework/test-output/index.html#), and various browser tabs.
- Main Content Area:**
 - All suites:** Shows the suite structure with **Reporter output for Suite** containing the following message:


```

Day11TestCase(Noodles, Noodles, 3, Enabled, C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\ TestData...)
validation on Error Message - Category Tag Title as mandatory field!! Validation of Success Message after adding new Category!! Validation of newly added Category in Admin Panel Table!! Validation of Category header in User Page!! Assertion on New Category in Database!!
          
```
 - Info:** Details about the suite configuration, including the XML file path (`C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\testing.xml`), number of groups (0), times (Times), reporter output (selected), ignored methods, and chronological view.
 - Results:** Summary of test results: 1 method passed, 0 failed, 0 skipped. A link to the detailed results page (`Day11TestCase(Noodles, Noodles, 3, Enabled, C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\ TestData..)`) is provided.
- Bottom:** Browser toolbar and system tray.

19. Automated Emailable report

The screenshot shows a web browser window with the title "TestNG Report". The URL in the address bar is "file:///C:/Harisha/Selenium_Training/EclipseWorkSpace/SeleniumFramework/test-output/emailable-report.html". The page displays the results of a Java test case named "Day11TestCase".

Test Summary:

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
Test	1	0	0	35,707		

Test Case Details:

Class	Method	Start	Time (ms)
Suite			
com.ibm.test.BaseTest	Day11TestCase	1546203811693	28445

Parameter Values:

Parameter #1	Parameter #2	Parameter #3	Parameter #4	Parameter #5	Parameter #6	Parameter #7
Noodles	Noodles	3	Enabled	C:\Harisha\Selenium_Training\EclipseWorkSpace\SeleniumFramework\TestData\noodles.jpg	fill out this field	successfully added category

Messages:

```
validation on Error Message - Category Tag Title as mandatory field!!
Validation of Success Message after adding new Category!!
Validation of newly added Category in Admin Panel Table!!
Validation of Category header in User Page!!
Assertion on New Category in Database!!
```

[back to summary](#)

*** End of the document ***