

```
package com.automationexercise.tests;
```

```
import java.io.IOException;
```

```
import java.time.Duration;
```

```
import org.openqa.selenium.NoAlertPresentException;
```

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

```
import org.testng.annotations.Test;
```

```
import com.automationexercise.Base.BaseTest;
```

```
import com.automationexercise.pages.TC_Ecom_ContactUs_Func;
```

```
import com.automationexercise.utilities.ScreenshotUtilities;
```

```
import com.aventstack.extentreports.ExtentTest;
```

```
public class TC_Ecom_ContactUs_Functional_Test extends BaseTest {
```

```
    String projectpath=System.getProperty("user.dir");
```

```
    // TC01. Verify Home Page URL
```

```
    /* @Test
```

```
    public void VerifyUrlNavigatesHomepage() throws IOException {
```

```
        ExtentTest test=extent.createTest("To verify and validate that the user can navigate  
to the URL: 'http://automationexercise.com'");
```

```
        driver.get("https://automationexercise.com");
```

```
        String expectedURL = "https://automationexercise.com/";
```

```
        String actualURL = driver.getCurrentUrl();
```

```
        if(actualURL.equals(expectedURL)) {
```

```
            test.pass("TC01 Passed: URL navigates to the home  
page").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Homepage"));
```

```
        }
```

```
    else {
```

```

        test.fail("TC01 Failed: Url is not navigated to home
page").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Homepage fail"));
    }
}

```

// TC02. Verify on clicking contactus its should open contact us page

@Test

public void VerifyHomeNavigatesContactuspage() throws IOException {

```

        ExtentTest test=extent.createTest("To verify and validate that the user can navigate
to the Contact us page");

```

```

        driver.get("https://automationexercise.com");

```

```

        TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);

```

```

        contactus.clickContactUs();

```

```

        String actualTitle = driver.getTitle();

```

```

        if(actualTitle.equalsIgnoreCase("Automation Exercise - Contact us")) {

```

```

            test.pass("Navigated to contact us page");

```

```

        }

```

```

        else {

```

```

            test.fail("Not navigated to contact us
page").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Navigation to
contact us page fail"));

```

```

        }

```

```

    }

```

// TC03. Verify form submission with valid details and file upload

@Test

public void verifyContactFormSubmission() throws IOException {

```

        ExtentTest test = extent.createTest("To verify and validate the Contact Us form
submission with valid details and file upload");

```

```

        driver.get("https://automationexercise.com");

```

```

TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);

contactus.clickContactUs();

contactus.enterName("Sneha");

contactus.enterEmail("03snehan@gmail.com");

contactus.enterSubject("Capstone project");

contactus.enterMessage("Testing automation exercise");

String filePath = projectpath +
"\src\test\resources\automationexercise_TestData\CapstoneProject.txt";

contactus.uploadFile(filePath);

contactus.clickSubmit();


//to handle alert after submission
try {
driver.switchTo().alert().accept();
}catch (NoAlertPresentException e) {
}

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

if(contactus.isSuccessMessageDisplayed()) {
    test.pass("Form submitted successfully");
}

else {
    test.fail("Form not
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form
submission failed"));
}

}

// TC04. Verify form submission with valid details without file upload

@Test

public void verifyContactFormSubmissionWithoutFile() throws IOException {

```

```
ExtentTest test = extent.createTest("To verify and validate the Contact Us form submission with all valid details but without file upload");
```

```
driver.get("https://automationexercise.com");
```

```
TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);  
contactus.clickContactUs();
```

```
// Enter valid details
```

```
contactus.enterName("Sneha");
```

```
contactus.enterEmail("sneha@example.com");
```

```
contactus.enterSubject("Capstone project");
```

```
contactus.enterMessage("Testing automation exercise");
```

```
contactus.clickSubmit();
```

```
//to handle alert after submission
```

```
try {
```

```
driver.switchTo().alert().accept();
```

```
}catch (NoAlertPresentException e) {
```

```
}
```

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
```

```
if(contactus.isSuccessMessageDisplayed()) {
```

```
test.pass("Form submitted successfully");
```

```
}
```

```
else {
```

```
test.fail("Form not submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form submission failed"));
```

```
}
```

```
//TC05. // TC04. Verify form submission with valid details without name
```

```
@Test
```

```

public void verifyContactFormWithoutName() throws IOException {

    ExtentTest test = extent.createTest("Verify Contact Us form cannot be submitted with
empty name");

    driver.get("https://automationexercise.com");

    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
    contactus.clickContactUs();

    // Leave Name empty
    contactus.enterName("");
    contactus.enterEmail("sneha@example.com");
    contactus.enterSubject("Capstone project");
    contactus.enterMessage("Testing automation exercise");

    contactus.clickSubmit();

    //to handle alert after submission
    try {

        driver.switchTo().alert().accept();

    } catch (NoAlertPresentException e) {

        // continue

    }

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

    if(contactus.isSuccessMessageDisplayed()) {

        test.pass("Form submitted successfully");

    }

    else {

        test.fail("Form not
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form
submission failed"));

    }

}

```

```

// TC06. Verify form submission with empty Email field

@Test

public void verifyContactFormWithoutEmail() throws IOException {

    ExtentTest test = extent.createTest("Verify Contact Us form cannot be submitted with
empty Email address");

    driver.get("https://automationexercise.com");

    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
    contactus.clickContactUs();

    // Enter valid details
    contactus.enterName("Sneha");
    contactus.enterEmail();    //leave email blank
    contactus.enterSubject("Capstone project");
    contactus.enterMessage("Testing automation exercise");
    contactus.clickSubmit();

    //to handle alert after submission
    try {

        driver.switchTo().alert().accept();

    } catch (NoAlertPresentException e) {

        // continue

    }

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

    String emailValidationMessage =
contactus.enterEmail().getAttribute("validationMessage");

    if (emailValidationMessage != null && !emailValidationMessage.isEmpty()) {

        System.out.println("Validation message is displayed");

    } else {

        System.out.println("Validation message is NOT displayed");

```

```
        test.fail("Message not  
displayed").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Navigation to  
contact us page fail"));
```

```
    }
```

```
}
```

```
// TC07. To verify and validate that the form cannot be submitted with an invalid email  
format.
```

```
@Test
```

```
public void verifyContactFormWithInvalidEmail() throws IOException {
```

```
    ExtentTest test = extent.createTest("Verify Contact Us form cannot be submitted with  
invalid email format");
```

```
    driver.get("https://automationexercise.com");
```

```
    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
```

```
    contactus.clickContactUs();
```

```
    // Enter invalid email format
```

```
    contactus.enterName("Sneha");
```

```
    contactus.enterEmail("snehacom12");
```

```
    contactus.enterSubject("Test Invalid Email");
```

```
    contactus.enterMessage("Testing automation exercise");
```

```
    // Click submit
```

```
    contactus.clickSubmit();
```

```
    try {
```

```
        driver.switchTo().alert().accept();
```

```
    } catch (NoAlertPresentException e) {
```

```
        // if no alert continue
```

```
    }
```

```
    String emailValidationMessage =
```

```
    contactus.enterEmail().getAttribute("validationMessage");
```

```
        if (emailValidationMessage != null && !emailValidationMessage.isEmpty()) {
```

```

        System.out.println("Validation message is displayed");
        test.pass("Validation message is displayed");
    } else {
        System.out.println("Validation message is NOT displayed");
        test.fail("Message not
displayed").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Navigation to
contact us page fail"));
    }
}

```

//TC08. To verify and validate that the form cannot be submitted with an empty 'Subject' field.

```

@Test
public void verifyContactFormWithoutSubject() throws IOException {
    ExtentTest test = extent.createTest("Verify Contact Us form cannot be submitted with
invalid email format");

```

```

    driver.get("https://automationexercise.com");

```

```

    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
    contactus.clickContactUs();

```

```

    // Enter invalid email format
    contactus.enterName("Sneha");
    contactus.enterEmail("03snehan@gmail.com");
    contactus.enterSubject("");
    contactus.enterMessage("Testing automation exercise");

```

```

    // Click submit
    contactus.clickSubmit();
    try {
        driver.switchTo().alert().accept();
    } catch (NoAlertPresentException e) {

```



```

        // if no alert continue
    }

    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

    if(contactus.isSuccessMessageDisplayed()) {

        test.pass("Form submitted successfully");

    }

    else {

        test.fail("Form not
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form cannot
be submitted without subject"));

    }

}

```

//TC09. To verify and validate that the form cannot be submitted with an empty 'Message' field.

```

@Test
public void verifyContactFormWithoutMessage() throws IOException {

    ExtentTest test = extent.createTest("Verify Contact Us form cannot be submitted with
invalid email format");

```

```

    driver.get("https://automationexercise.com");

```

```

    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
    contactus.clickContactUs();

```

```

    // Enter invalid email format
    contactus.enterName("Sneha");
    contactus.enterEmail("03snehan@gmail.com");
    contactus.enterSubject("Capstone project");
    contactus.enterMessage("");

```

```

    // Click submit

```

```

contactus.clickSubmit();

try {
    driver.switchTo().alert().accept();
} catch (NoAlertPresentException e) {
    // if no alert continue
}

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));

if(contactus.isSuccessMessageDisplayed()) {
    test.pass("Form submitted successfully");
}
else {
    test.fail("Form not
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form cannot
be submitted without meassage"));
}
}

```

//TC10. To verify and validate that the form submission is cancelled when the user clicks 'Cancel' on the confirmation alert.

```

@Test
public void verifyContactFormCancelSubmission() throws IOException {
    ExtentTest test = extent.createTest("Verify form submission is cancelled when user clicks
'Cancel' on alert");

```

```

driver.get("https://automationexercise.com");

```

```

TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
contactus.clickContactUs();
contactus.enterName("Sneha");
contactus.enterEmail("sneha@example.com");
contactus.enterSubject("Capstone Project");
contactus.enterMessage("Testing automation exercise");

```

```

contactus.clickSubmit();

try {
    driver.switchTo().alert().dismiss();
    System.out.println("Alert dismissed (Cancel clicked)");
    test.pass("User clicked Cancel on confirmation alert and submission cancelled");
} catch (NoAlertPresentException e) {
    System.out.println("No alert appeared!");
    test.fail("No confirmation alert
appeared").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver,
"NoAlertOnCancel"));
}

if (contactus.isSuccessMessageDisplayed()) {
    test.fail("Form submitted even after clicking cancel");

} else {
    test.pass("Form submission cancelled as
expected").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Submission
not cancelled"));
}
}

```

//TC11. To verify and validate the functionality of the 'Home' button on the success page.

@Test

```

public void verifyHomeButtonFunctionality() throws IOException {
    ExtentTest test = extent.createTest("Verify functionality of home button on success page");

```

```

    driver.get("https://automationexercise.com");

```

```

    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
    contactus.clickContactUs();

```

```

        contactus.enterName("Sneha");
        contactus.enterEmail("sneha@example.com");
        contactus.enterSubject("Capstone Project");
        contactus.enterMessage("Testing automation exercise");

        contactus.clickSubmit();

        try {
            driver.switchTo().alert().accept();
        } catch (NoAlertPresentException e) {
            // if no alerts continue
        }

        contactus.clickHomeButton();

        String expectedURL = "https://automationexercise.com/";

        String actualURL = driver.getCurrentUrl();

        if(actualURL.equals(expectedURL)) {
            test.pass("Home button navigated to home page");
        }

        else {
            test.fail("Home button is not navigated to home
page").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Homepage fail"));
        }
    }

    //TC12. To verify and validate form submission with numeric characters in the 'Name' field.
    @Test
    public void verifyContactFormWithNumericName() throws IOException {
        ExtentTest test = extent.createTest("Verify form submission with numeric characters in the
name");

        driver.get("https://automationexercise.com");

```

```
TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);  
contactus.clickContactUs();
```

```
contactus.enterName("12345");  
contactus.enterEmail("sneha@example.com");  
contactus.enterSubject("Capstone project");  
contactus.enterMessage("Testing automation exercise");
```

```
contactus.clickSubmit();  
  
try {  
    driver.switchTo().alert().accept();  
} catch (NoAlertPresentException e) {  
    // continue  
}
```

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));  
  
if(contactus.isSuccessMessageDisplayed()) {  
    test.pass("Form submitted successfully");  
}  
  
else {  
    test.fail("Form not  
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form cannot  
be submitted with numeric in name"));  
}  
}
```

//TC13. To verify and validate form submission with special characters in the 'Name' field.

@Test

```
public void verifyContactFormWithSpecialCharName() throws IOException {  
    ExtentTest test = extent.createTest("Verify form submission with numeric characters in the  
name");
```

```
driver.get("https://automationexercise.com");
```

```
TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
```

```
contactus.clickContactUs();
```

```
contactus.enterName("sn@ha#");
```

```
contactus.enterEmail("sneha@example.com");
```

```
contactus.enterSubject("TCapstone Project");
```

```
contactus.enterMessage("Testing automation exercise");
```

```
contactus.clickSubmit();
```

```
try {
```

```
    driver.switchTo().alert().accept();
```

```
} catch (NoAlertPresentException e) {
```

```
    // continue
```

```
}
```

```
if(contactus.isSuccessMessageDisplayed()) {
```

```
    test.pass("Form submitted successfully");
```

```
}
```

```
else {
```

```
    test.fail("Form not  
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form cannot  
be submitted with special character in name"));
```

```
}
```

```
}
```

//TC14. To verify and validate form submission with special characters in the 'Subject' and 'Message' fields.

```
@Test
```

```
public void verifyContactFormWithSpecialCharSubMsg() throws IOException {
```

```
ExtentTest test = extent.createTest("Verify form submission with numeric characters in the name");
```

```
driver.get("https://automationexercise.com");
```

```
TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);  
contactus.clickContactUs();
```

```
contactus.enterName("Sneha");  
contactus.enterEmail("sneha@example.com");  
contactus.enterSubject("Aut@mati@n T@est @#$%^&*()_+{}|:~\\\"'");  
contactus.enterMessage("Testing@ for#$%^&* C@pstone project");
```

```
contactus.clickSubmit();  
try {  
    driver.switchTo().alert().accept();  
} catch (NoAlertPresentException e) {  
    // continue  
}
```

```
if(contactus.isSuccessMessageDisplayed()) {  
    test.pass("Form submitted successfully");  
}  
else {  
    test.fail("Form not  
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form cannot  
be submitted with special character in mesaage and subject field"));  
}  
}
```

```
// TC15. To verify and validate form submission with a very long string in the 'Name' field.
```

```
@Test
```

```

    public void verifyFormSubmissionwithLongStr() throws IOException {

        ExtentTest test = extent.createTest("To verify and validate the Contact Us form
        submission with valid details and file upload");

        driver.get("https://automationexercise.com");

        TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
        contactus.clickContactUs();

        contactus.enterName("Snehaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa")
;

        contactus.enterEmail("03snehan@gmail.com");
        contactus.enterSubject("Capstone project");
        contactus.enterMessage("Testing automation exercise");

        String filePath = projectpath +
        "\\src\\test\\resources\\automationexercise_TestData\\CapstoneProject.txt";
        contactus.uploadFile(filePath);
        contactus.clickSubmit();

        //to handle alert after submission
        try {
            driver.switchTo().alert().accept();
        } catch (NoAlertPresentException e) {
        }

        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        if(contactus.isSuccessMessageDisplayed()) {
            test.pass("Form submitted successfully");
        }
        else {
            test.fail("Form not
            submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form
            submission failed"));
        }
    }

```



```
}
```

```
//TC16 To verify and validate form submission with a very long string in the 'Subject' field.
```

```
@Test
```

```
public void verifyContactFormSubmission() throws IOException {
```

```
    ExtentTest test = extent.createTest("To verify and validate the Contact Us form  
submission with valid details and file upload");
```

```
    driver.get("https://automationexercise.com");
```

```
    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
```

```
    contactus.clickContactUs();
```

```
    contactus.enterName("Sneha");
```

```
    contactus.enterEmail("03snehan@gmail.com");
```

```
    contactus.enterSubject("In Capstone project we are automating test cases using  
automation exercise url ");
```

```
    contactus.enterMessage("Testing automation exercise");
```

```
    String filePath = projectpath +  
    "\\src\\test\\resources\\automationexercise_TestData\\CapstoneProject.txt";
```

```
    contactus.uploadFile(filePath);
```

```
    contactus.clickSubmit();
```

```
    //to handle alert after submission
```

```
    try {
```

```
        driver.switchTo().alert().accept();
```

```
    }catch (NoAlertPresentException e) {
```

```
    }
```

```
    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
```

```
    if(contactus.isSuccessMessageDisplayed()) {
```

```
        test.pass("Form submitted successfully");
```

```
    }
```

```
    else {
```

```
        test.fail("Form not  
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form  
submission failed"));
```

```
    }
```

```
}
```

```
//TC17 To verify and validate form submission with a very long string in the 'Message' field.
```

```
@Test
```

```
public void verifyContactFormSubmissionwithLongMsg() throws IOException {
```

```
    ExtentTest test = extent.createTest("To verify and validate the Contact Us form  
submission with valid details and file upload");
```

```
    driver.get("https://automationexercise.com");
```

```
    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
```

```
    contactus.clickContactUs();
```

```
    contactus.enterName("Sneha");
```

```
    contactus.enterEmail("03snehan@gmail.com");
```

```
    contactus.enterSubject("In Capstone project");
```

```
    contactus.enterMessage("On the Automation Exercise website, the Contact Us form  
includes built-in validation to ensure users provide correct information. If a user enters an invalid  
email format, the form displays a clear message prompting them to correct it. This feature helps  
prevent incorrect submissions and ensures that messages reach the intended recipients reliably.");
```

```
    String filePath = projectpath +  
    "\\src\\test\\resources\\automationexercise_TestData\\CapstoneProject.txt";
```

```
    contactus.uploadFile(filePath);
```

```
    contactus.clickSubmit();
```

```
//to handle alert after submission
```

```
try {
```

```
    driver.switchTo().alert().accept();
```

```
}catch (NoAlertPresentException e) {
```

```
}
```

```
WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
```

```

        if(contactus.isSuccessMessageDisplayed()) {
            test.pass("Form submitted successfully");
        }
        else {
            test.fail("Form not
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form
submission failed"));
        }
    }
}

```

//TC18 To verify and validate that the form cannot be submitted with all fields empty.

@Test

```

public void verifyContactFormWithAllFieldsEmpty() throws IOException {
    ExtentTest test = extent.createTest("Verify Contact Us form cannot be submitted with all
fields empty");

```

```

    driver.get("https://automationexercise.com");

```

```

    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);

```

```

    contactus.clickContactUs();

```

```

    contactus.enterName();

```

```

    contactus.enterEmail("");

```

```

    contactus.enterSubject("");

```

```

    contactus.enterMessage("");

```

```

    contactus.clickSubmit();

```

// Handle alert if appears

```

    try {

```

```

        driver.switchTo().alert().accept();

```

```

    } catch (NoAlertPresentException e) {

```

```

        // continue
    }
}

```

```

    }

    String nameValidation = contactus.enterName().getAttribute("validationMessage");
    String emailValidation = contactus.enterEmail().getAttribute("validationMessage");
    String subjectValidation = contactus.enterSubject().getAttribute("validationMessage");
    String messageValidation = contactus.enterMessage().getAttribute("validationMessage");
    if ((nameValidation != null && !nameValidation.isEmpty()) ||
        (emailValidation != null && !emailValidation.isEmpty()) ||
        (subjectValidation != null && !subjectValidation.isEmpty()) ||
        (messageValidation != null && !messageValidation.isEmpty())) {
        System.out.println("Form cannot be submitted with empty fails");
        test.pass("Form cannot be submitted with empty fields");
    }
    else {
        System.out.println("Form submitted with empty fields");
        test.fail("Form submitted with empty
fields").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form submission
failed"));
    }
}

```

//TC19. To verify and validate that another email format is considered invalid.

@Test

```

public void verifyContactFormWithAnotherInvalidEmail() throws IOException {
    ExtentTest test = extent.createTest("Verify form cannot be submitted with another invalid
email format");

```

```

    driver.get("https://automationexercise.com");

```

```

    TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
    contactus.clickContactUs();

```

```

        contactus.enterName("Sneha");

        contactus.enterEmail("sneha@.com");

        contactus.enterSubject("Test Invalid Email Format 2");

        contactus.enterMessage("Testing contact form submission with another invalid email
format.");

        String filePath = projectpath +
"\src\test\resources\automationexercise_TestData\CapstoneProject.txt";

        contactus.uploadFile(filePath);

        contactus.clickSubmit();


        //to handle alert after submission
        try {

            driver.switchTo().alert().accept();

        } catch (NoAlertPresentException e) {

        }

        String emailValidationMessage =
contactus.enterEmail().getAttribute("validationMessage");

        if (emailValidationMessage != null && !emailValidationMessage.isEmpty()) {

            System.out.println("Validation message is displayed");

            test.pass("Validation message is displayed");

        } else {

            System.out.println("Validation message is NOT displayed");

            test.fail("Message not
displayed").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Navigation to
contact us page fail"));

        }

    }

}

//TC20 To verify and validate form submission with a different valid file type (e.g., .pdf).

@Test

public void verifyContactFormWithPdfUpload() throws IOException {

    ExtentTest test = extent.createTest("Verify form submission with a valid PDF file upload");

```

```

driver.get("https://automationexercise.com");

TC_Ecom_ContactUs_Func contactus = new TC_Ecom_ContactUs_Func(driver);
contactus.clickContactUs();

contactus.enterName("Sneha");
contactus.enterEmail("sneha@example.com");
contactus.enterSubject("Test PDF File Upload");
contactus.enterMessage("Testing form submission with a PDF file upload.");

String pdfFilePath = projectpath +
"\\src\\test\\resources\\automationexercise_TestData\\CapstoneProject.png";
contactus.uploadFile(pdfFilePath);
contactus.clickSubmit();

try {
    driver.switchTo().alert().accept();
} catch (NoAlertPresentException e) {
}

WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
if(contactus.isSuccessMessageDisplayed()) {
    test.pass("Form submitted successfully with png file format");
}
else {
    test.fail("Form not
submitted").addScreenCaptureFromPath(ScreenshotUtilities.capturescreenshot(driver, "Form
submission failed with png file format"));
}
} */

```

//TC21 To verify and validate the system's behavior when uploading a file that exceeds the size limit.

@Test

public void verifyContactFormWithLargeFileUpload() **throws** IOException {

ExtentTest test = extent.createTest("Verify system behavior when uploading a file exceeding size limit");

driver.get("https://automationexercise.com");

TC_Ecom_ContactUs_Func contactus = **new** TC_Ecom_ContactUs_Func(driver);

contactus.clickContactUs();

contactus.enterName("Sneha");

contactus.enterEmail("sneha@example.com");

contactus.enterSubject("Test Large File Upload");

contactus.enterMessage("Testing form submission with a file exceeding the allowed size limit.");

String pdfFilePath = projectpath +
"\\src\\test\\resources\\automationexercise_TestData\\CapstoneProject.png";

contactus.uploadFile(pdfFilePath);

contactus.clickSubmit();

try {

driver.switchTo().alert().accept();

catch (NoAlertPresentException e) {

}

if(contactus.isSuccessMessageDisplayed()) {

test.pass("Form submitted successfully with png file format");

}

else {

test.fail("Form not
submitted").addScreenCaptureFromPath(ScreenshotUtilities.captureScreenshot(driver, "Form
submission failed with large file size"));

}

}

}