Fakulti Sains Komputer dan Teknologi Maklumat
Universiti Tun Hussein Onn Malaysia

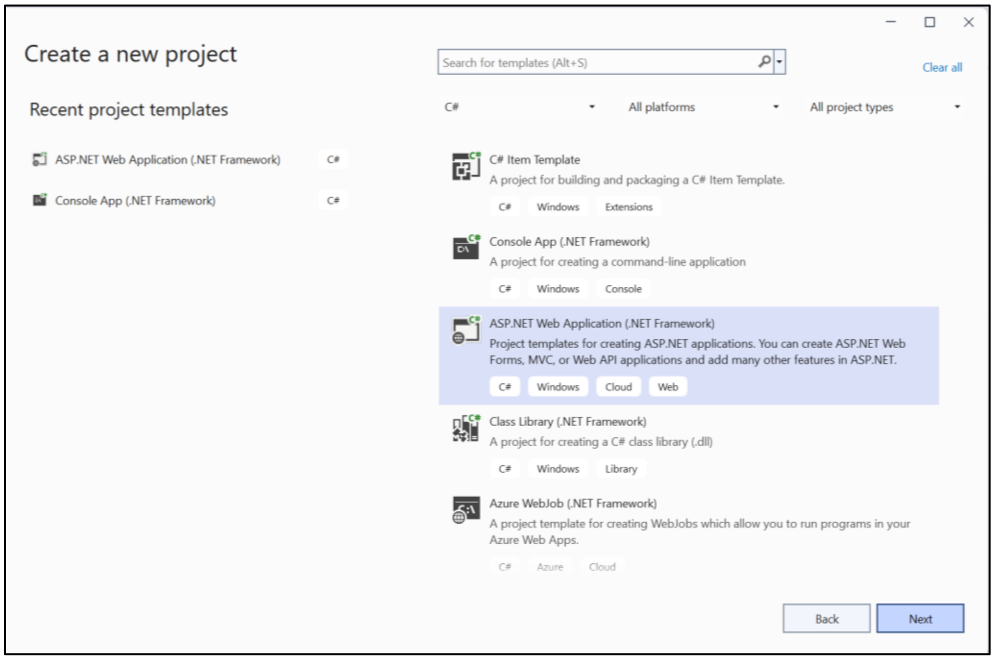
LABORATORY 3

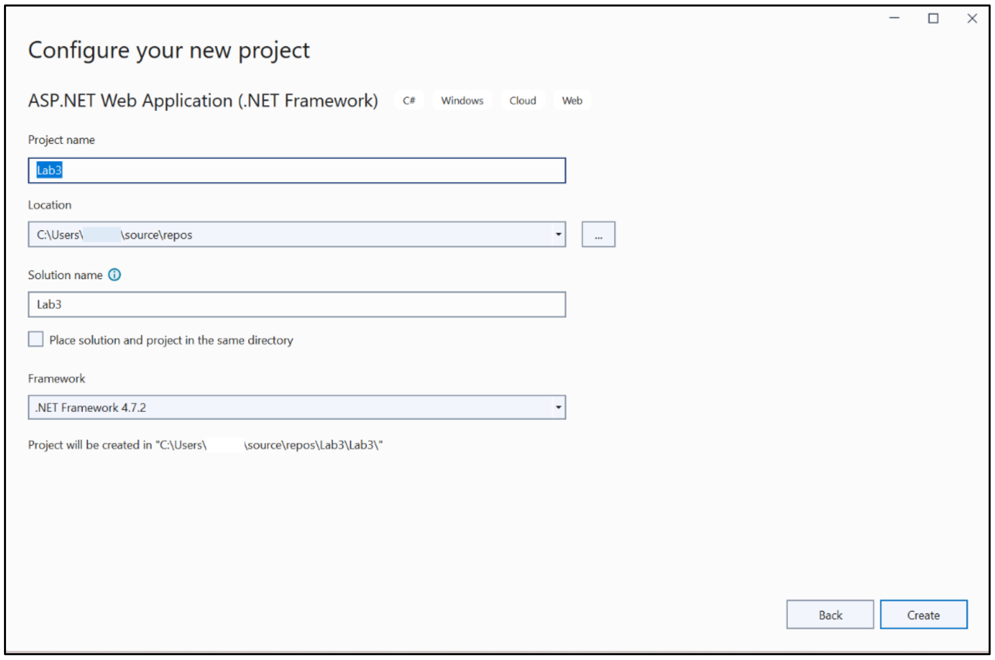
This laboratory exercise is about using Master Page and Content Page; and using validator.

A. Creating Master Page

Using Master Page is get the feel and look of consistent pages across our website, i.e., having the same template for all our pages except for the content.

1. Create new empty web sites


Create a new project
Search for templates (Alt+S)
Clear all
Recent project templates
ASP.NET Web Application (.NET Framework) C#
Console App (.NET Framework) C#
C#
All platforms
All project types
C# Item Template
A project for building and packaging a C# Item Template.
C# Windows Extensions
Console App (.NET Framework)
A project for creating a command-line application
C# Windows Console
ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
C# Windows Cloud Web
Class Library (.NET Framework)
A project for creating a C# class library (.dll)
C# Windows Library
Azure WebJob (.NET Framework)
A project template for creating WebJobs which allow you to run programs in your Azure Web Apps.
C# Azure Cloud
Back
Next



Configure your new project

ASP.NET Web Application (.NET Framework) C# Windows Cloud Web

Project name

Lab3

Location

C:\Users\ \source\repos

Solution name

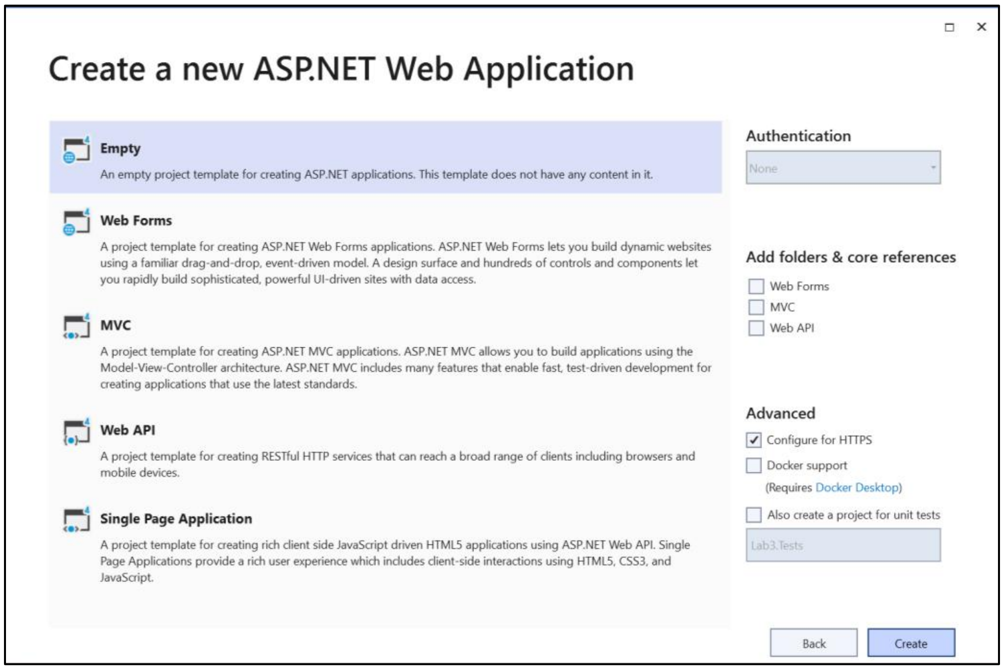
Lab3

☐ Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Project will be created in "C:\Users\ \source\repos\Lab3\Lab3\"



Create a new ASP.NET Web Application

Empty

An empty project template for creating ASP.NET applications. This template does not have any content in it.

Web Forms

A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

MVC

A project template for creating ASP.NET MVC applications. ASP.NET MVC allows you to build applications using the Model-View-Controller architecture. ASP.NET MVC includes many features that enable fast, test-driven development for creating applications that use the latest standards.

Web API

A project template for creating RESTful HTTP services that can reach a broad range of clients including browsers and mobile devices.

Single Page Application

A project template for creating rich client side JavaScript driven HTML5 applications using ASP.NET Web API. Single Page Applications provide a rich user experience which includes client-side interactions using HTML5, CSS3, and JavaScript.

Authentication

None

Add folders & core references

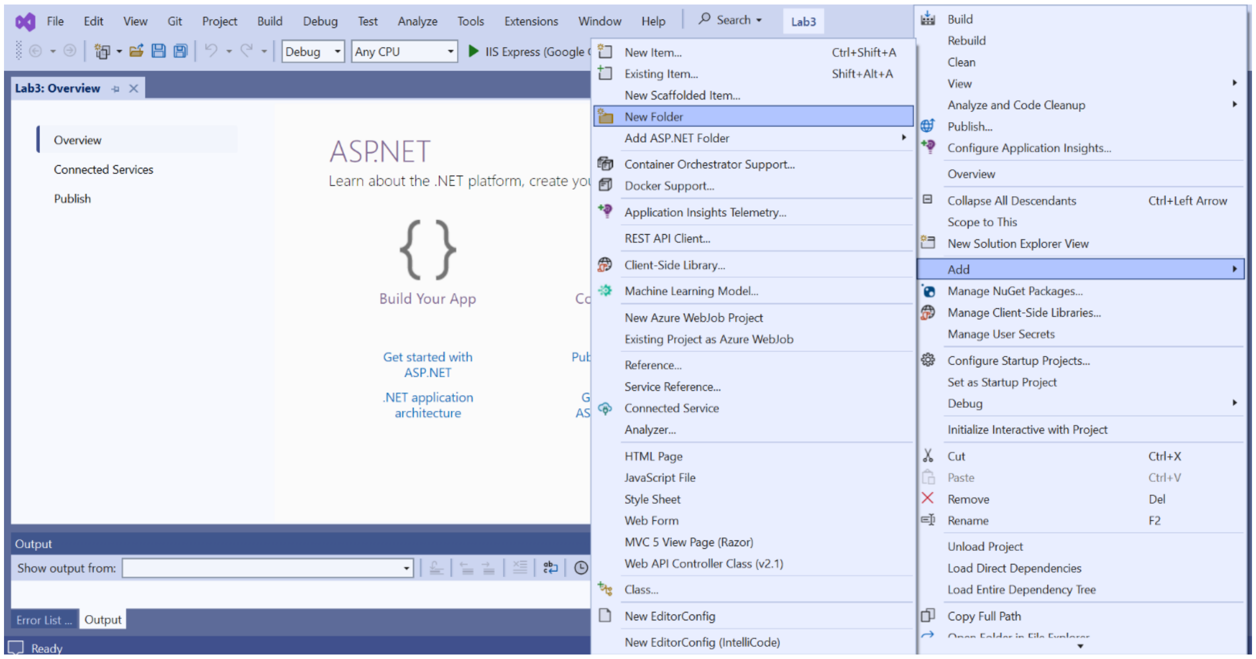
☐ Web Forms
☐ MVC
☐ Web API

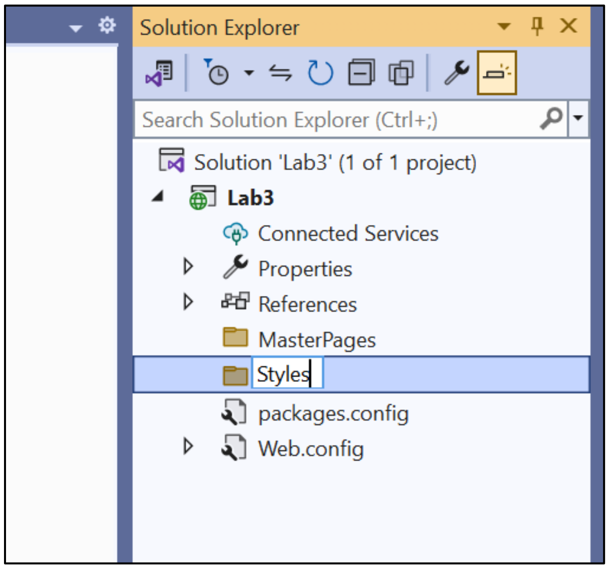
Advanced

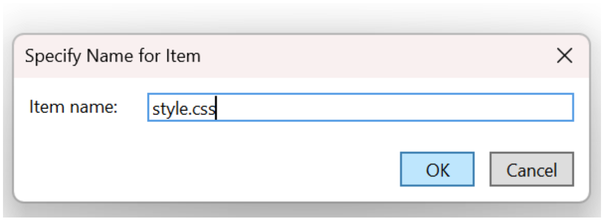
☑ Configure for HTTPS
☐ Docker support
(Requires Docker Desktop)
☐ Also create a project for unit tests

Lab3.Tests

2. Add New folders and name it as *MasterPages* and *Styles* (*MasterPages* for the master page, while *Styles* is for the CSS file)

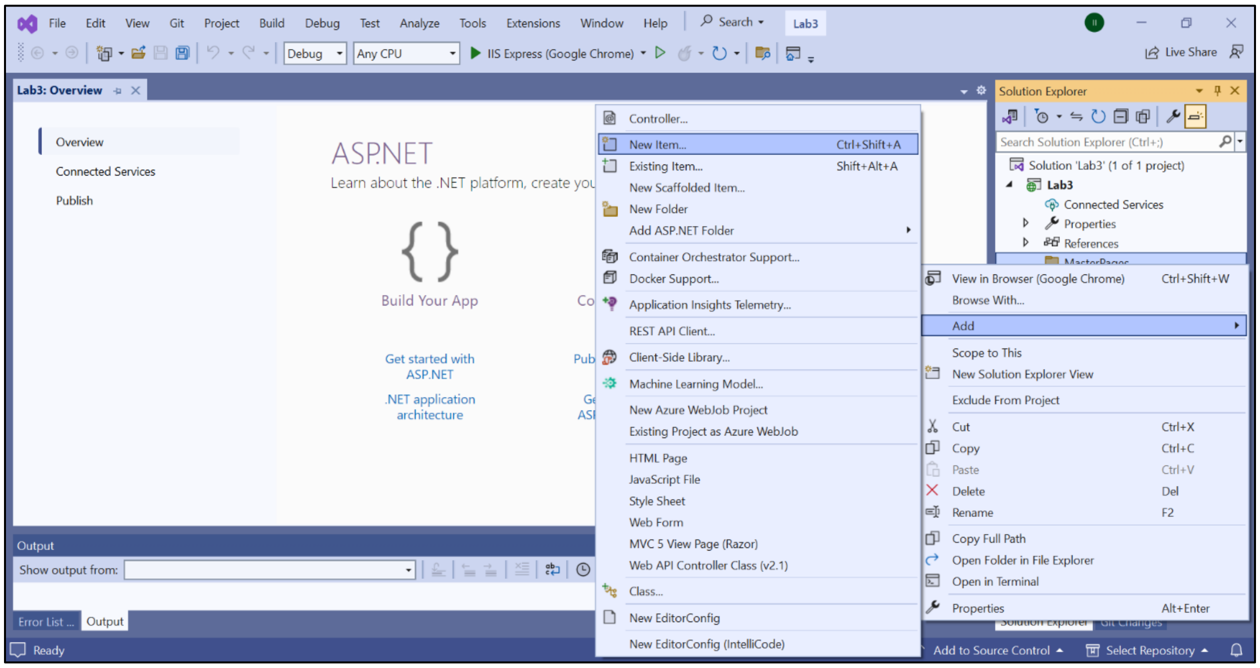


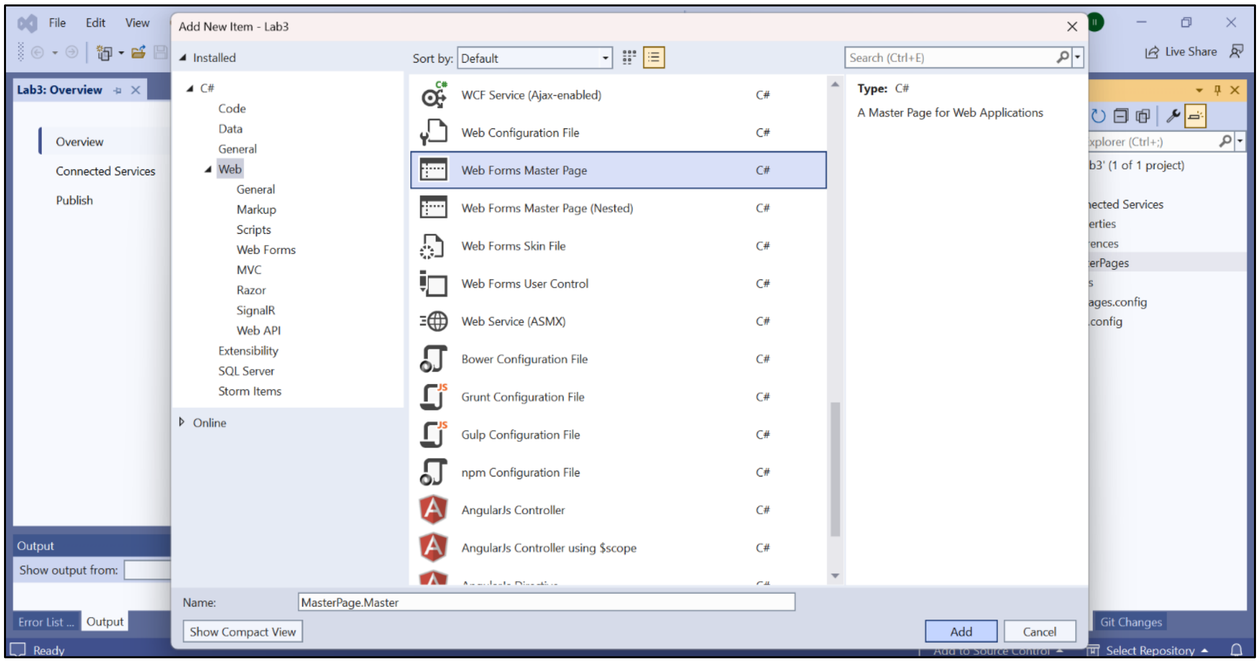




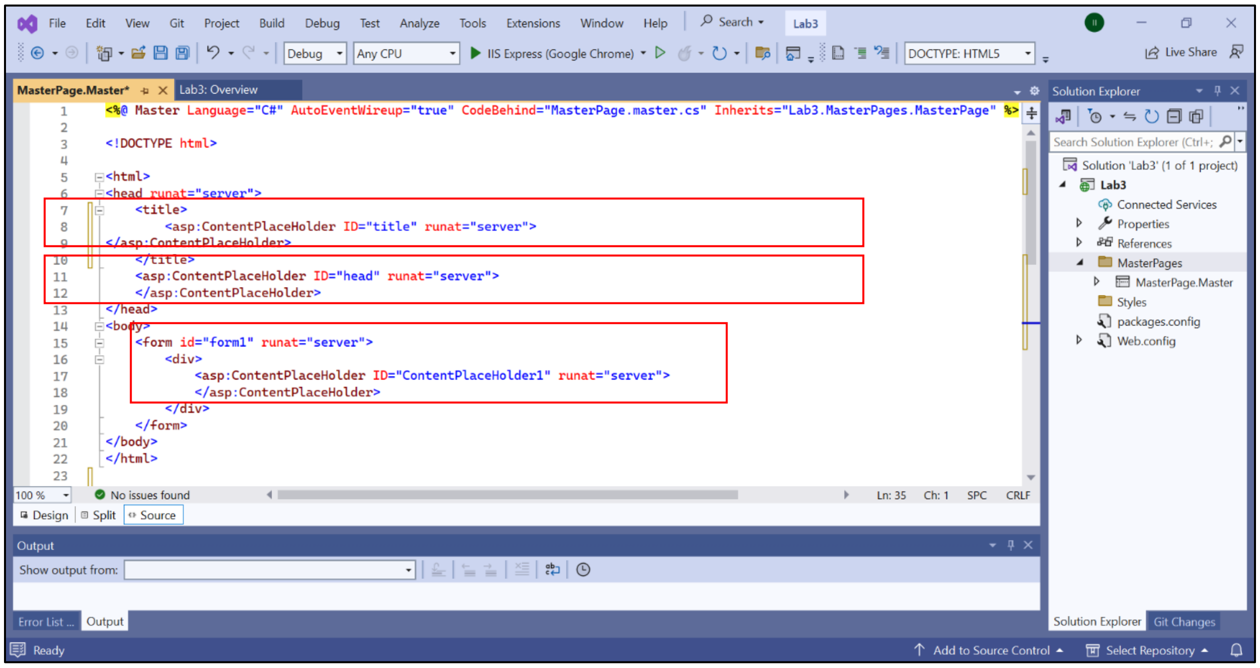
Item name: style.css

3. Right click on MasterPages folder and add Master Page



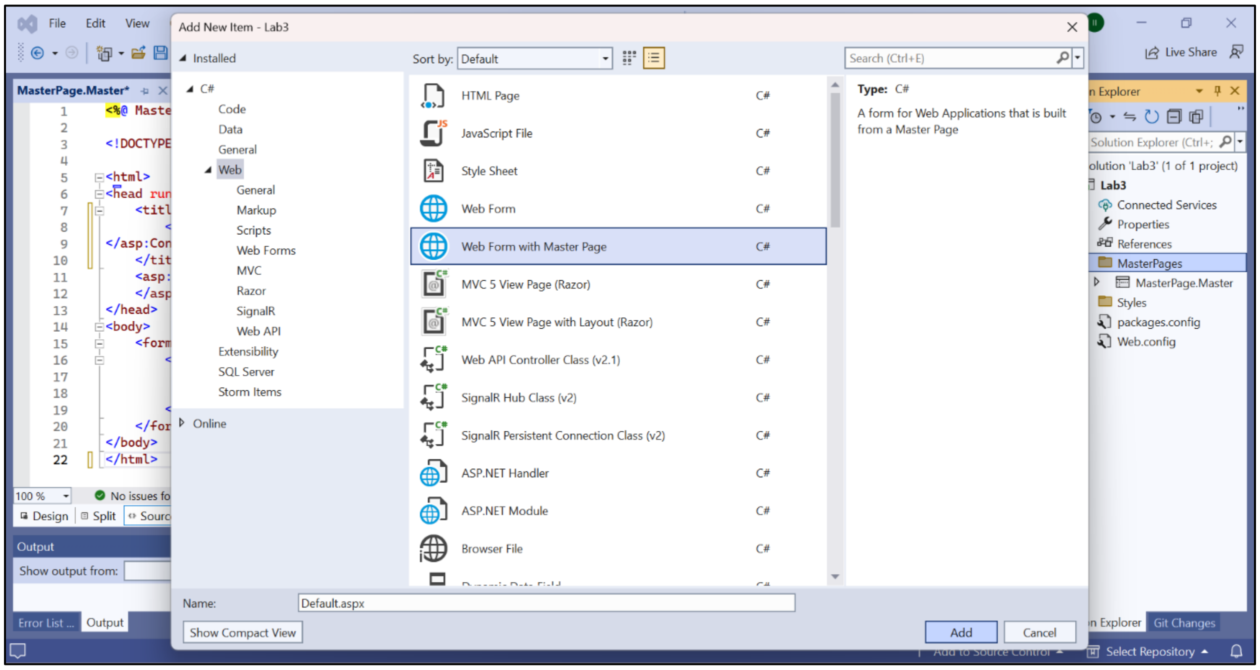


4. Create ContentPlaceHolder (where you want to put all the items/contents that are changeable according to different page). In this example, I use 3: (i) for the title, (ii) for the head, (iii) for body. ALL with different ContentPlaceHolder ID!



```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="MasterPage.master.cs" Inherits="Lab3.MasterPages.MasterPage" %>

<!DOCTYPE html>

<html>
<head runat="server">
    <title>
        <asp:ContentPlaceHolder ID="title" runat="server">
</asp:ContentPlaceHolder>
    </title>
    <asp:ContentPlaceHolder ID="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
            </asp:ContentPlaceHolder>
        </div>
    </form>
</body>
</html>
```

5. Right click on your folder on the Solution Explorer and create Default.aspx to use the master page created. Make sure to click on *web form with master page*. And then, select our created Master Page in Step 3 above.



Add New Item - Lab3

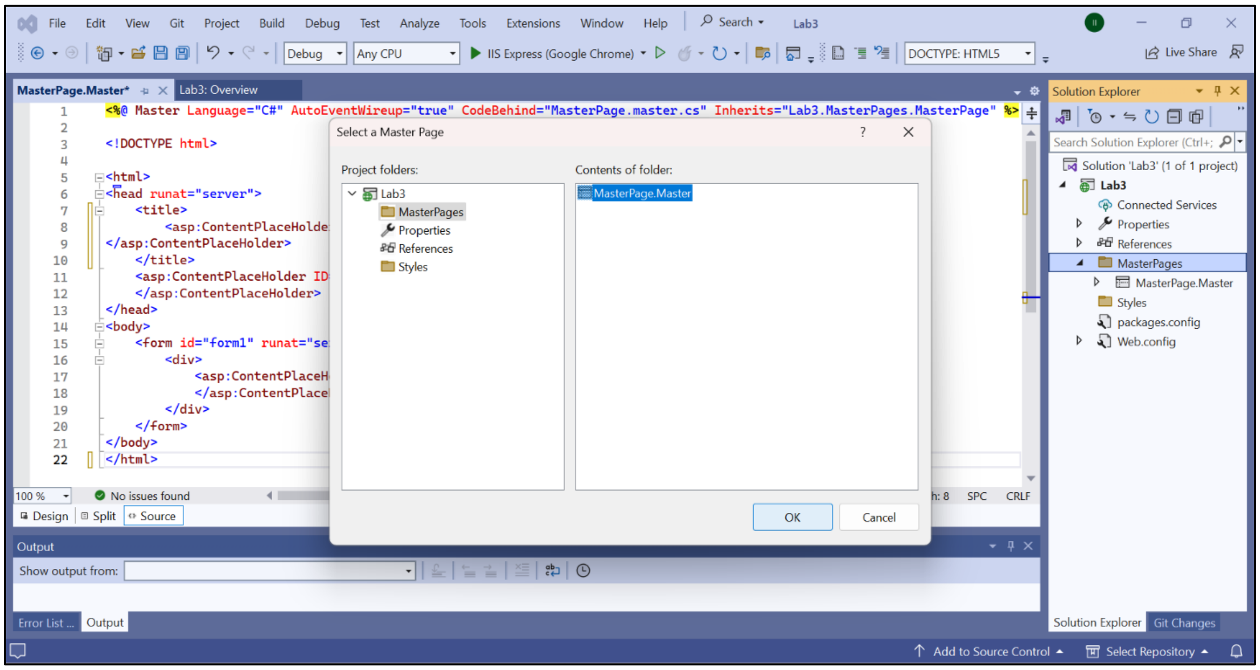
Installed: C# – Code, Data, General, Web (General, Markup, Scripts, Web Forms, MVC, Razor, SignalR, Web API), Extensibility, SQL Server, Storm Items; Online

Sort by: Default

HTML Page	C#
JavaScript File	C#
Style Sheet	C#
Web Form	C#
Web Form with Master Page	C#
MVC 5 View Page (Razor)	C#
MVC 5 View Page with Layout (Razor)	C#
Web API Controller Class (v2.1)	C#
SignalR Hub Class (v2)	C#
SignalR Persistent Connection Class (v2)	C#
ASP.NET Handler	C#
ASP.NET Module	C#
Browser File	C#

Type: C#
A form for Web Applications that is built from a Master Page

Name: Default.aspx



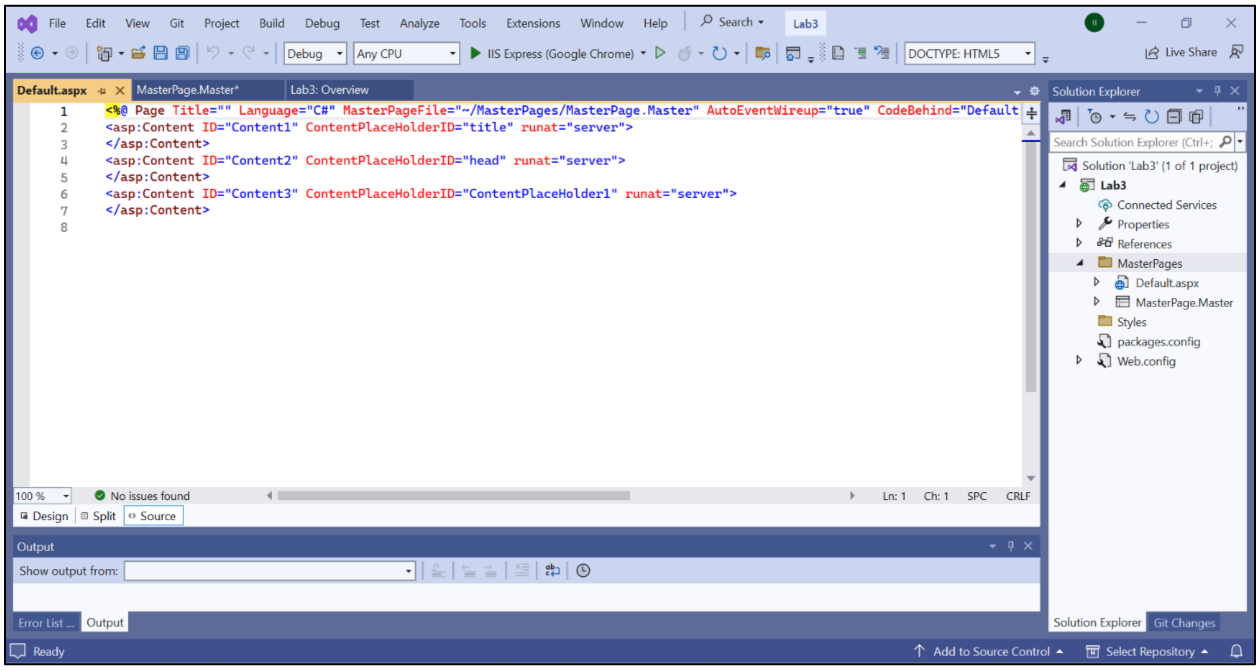
```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="MasterPage.master.cs" Inherits="Lab3.MasterPages.MasterPage" %>

<!DOCTYPE html>

<html>
<head runat="server">
    <title>
        <asp:ContentPlaceHolde
</asp:ContentPlaceHolder>
    </title>
    <asp:ContentPlaceHolder ID
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="se
        <div>
            <asp:ContentPlaceH
            </asp:ContentPlace
        </div>
    </form>
</body>
</html>
```

Select a Master Page

Project folders: Lab3 – MasterPages, Properties, References, Styles

Contents of folder: MasterPage.Master

6. Notice that, Default.aspx will contain three ContentPlaceHolder just like in Step 4.



```
<%@ Page Title="" Language="C#" MasterPageFile="~/MasterPages/MasterPage.Master" AutoEventWireup="true" CodeBehind="Default
<asp:Content ID="Content1" ContentPlaceHolderID="title" runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="head" runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="ContentPlaceHolder1" runat="server">
</asp:Content>
```

Pressing the Ctrl+F5 will bring you the output.

7. Say we want to have menu inside the template (master page), therefore, we create menu under *form1* (MasterPage.master)

```
<div class ="menu">
    <ul>
        <li><a href="Login.aspx">Login</a></li>
        <li><a href="#">About Us</a></li>
        <li><a href="#">Contact Us</a></li>
        <li><a href="#">Class</a></li>
        <li><a href="#">Gallery</a></li>
        <li><a href="Default.aspx">Home</a></li>
    </ul>
</div>
```

Pressing the Ctrl+F5 will bring you the output. Still not looking nice, right?

8. Use CSS to improve the Master Page by getting the preferred layout and nature of the page.
   Right click on Styles folder, add Style Sheet. Write your CSS file! Mine is like below:

```css
body {
    background-color: rgb(237,237,237);
    font-family: Arial, Helvetica, Sans-Serif;
    font-size: 12px;
}

.wrapper {
    width: 1368px;
    margin: auto;
}

.content {
    width: 100%;
    background-color: rgb(254,254,254);
    border: 1px solid rgb(224,224,224);
    border-radius: 5px 5px 5px 5px;
    float: left;
    margin-top: 8px;
    margin-bottom: 8px;
    min-height: 400px;
}

.menu {
    background-color: #114B5F;
    width: 100%;
    margin: 0px 0px 10px;
    padding: 0px;
    height: 40px;
    color: #114B5F;
    border-radius: 5px 5px 5px 5px;
}

    .menu ul {
    }

        .menu ul li {
            float: right;
            display: block;
            list-style: none;
            border-right: 1px solid #B1D4E0;
            border-left: 1px solid #B1D4E0;
        }


            .menu ul li a {
                font-size: 13px;
                font-weight: bold;
                line-height: 40px;
                padding: 8px 20px;
                color: rgb(255,255,255);
                text-decoration: none;
            }
            .menu ul li:hover {
                background-color: #1e82a5;
                border-right: 1px solid #1e82a5;
            }

.clear {
    clear: both;
}

.footer {
    height: 50px;
    background-color: #114B5F;
    color: rgb(255,255,255);
    border-radius: 5px 5px 5px;
}

    .footer h2 {
        padding: 15px;
        text-align: center;
    }
```

9. Go to Design View of Master page file, drag and drop style.css file onto this page, so that, when you go back to Source view, you will get the following code inside <head></head>

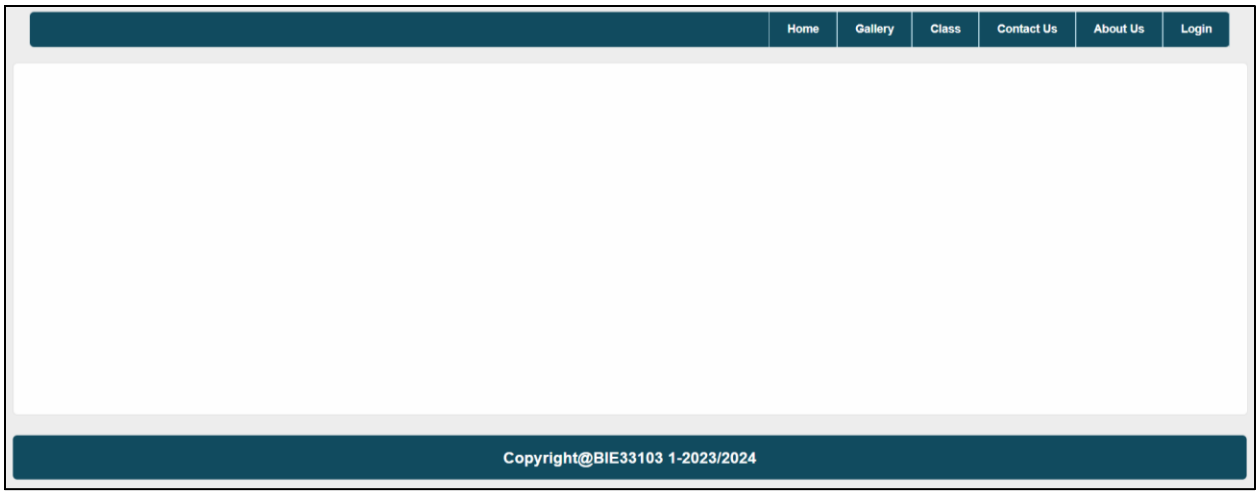
```
<link href="../Styles/style.css" rel="stylesheet" type="text/css" />
```

10. Match the css file with our Master page, therefore we need to create *class* for each of the component in css file, e.g.:

    We have footer component in CSS, then we create *<div class="footer"…* inside Master page file. The code becomes the following:

```
<form id="form1" runat="server">
    <div class="wrapper">

    <div class ="menu">
        <ul>
            <li><a href="Login.aspx">Login</a></li>
            <li><a href="#">About Us</a></li>
            <li><a href="ContactUs.aspx">Contact Us</a></li>
            <li><a href="#">Class</a></li>
            <li><a href="#">Gallery</a></li>
            <li><a href="Default.aspx">Home</a></li>
        </ul>
    </div>
</div>
<div class="clear"></div>
<div class="content">
    <div>
        <asp:ContentPlaceHolder ID="bodycontent" runat="server">
        </asp:ContentPlaceHolder>
    </div>
    </div>
<div class="clear"></div>
<div class="footer">
    <h2>Copyright@BIE33103 1-2023/2024</h2>
</div>
</form>
```

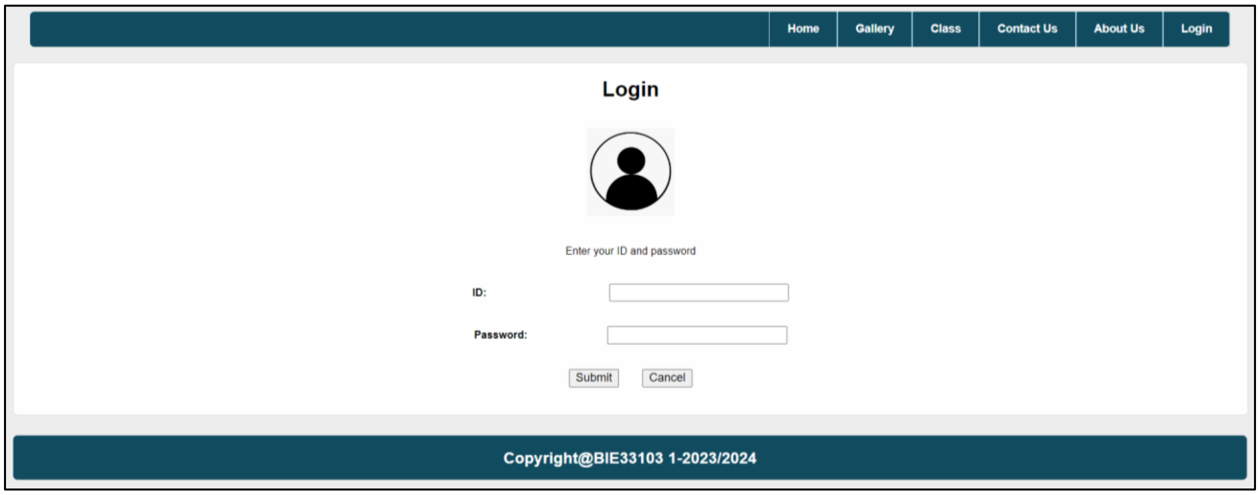
Pressing the Ctrl+F5 will bring you the output.



B. Using Master Page on Content Page

In Section A, we have already use Master Page on Content Page, i.e. on Default.aspx (the Content Page).

We can use the template (Master page) on other page as well, e.g. by creating Login.aspx and you get the following page. (Make sure to follow Step 5 in Section A).



C. Using validator(s)

1. In Section B, you may continue the process by creating the C# methods upon clicking Submit and Cancel button.

   For Submit button, mine is like the following:

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    string id = txtbox_ID.Text;
    string pword = txtbox_password.Text;
    if (id.Equals("admin") && pword.Equals("admin"))
    {
        //use this simple command if want re-direct to another page.
        Response.Redirect("default.aspx");
    }
    else
    {
        //use this simple command if want to display the status
        lblStatus.Text = "Login unsuccessful!";
    }
}
```

2. ASP.NET offer at least five validation controls for our use.

RequiredFieldValidator	Checks that the user has entered or selected anything.
RegularExpressionValidator	Checks user input against a regular expression. This allows a wide variety of checks to be made and can be used for things like ZIP codes and phone numbers.
CompareValidator	Compares an input control to a fixed value or another input control. It can be used for password verification fields, for example. It is also possible to do typed date and number comparisons.
RangeValidator	Much like CompareValidator, but can check that the input is between two fixed values.
CustomValidator	This allows you to write your own code to take part in the validation framework.

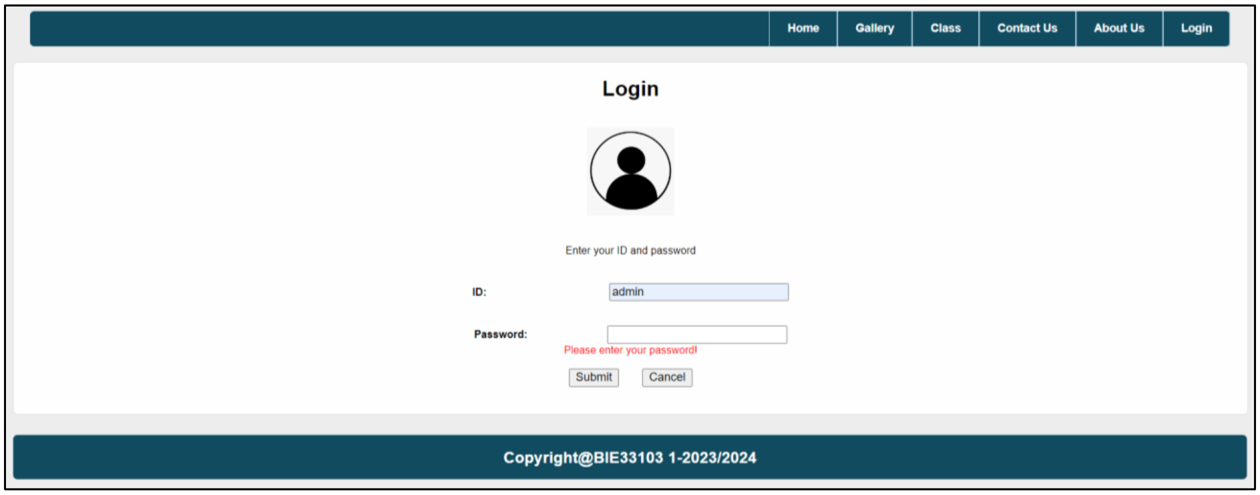
3. In this example, we may add RequiredFieldValidator checking on ID and password input to ensure that both data are inserted (inside Login.aspx). Following is sample of the code:

```
<asp:RequiredFieldValidator ID="rvfPassword" runat="server"
ControlToValidate="txtbox_password" ErrorMessage="Please enter your password!" ForeColor="Red">
</asp:RequiredFieldValidator>
<asp:Label ID="lblStatus" runat="server"></asp:Label>
<br /><br />
</div>
```

   To avoid the environment from requesting JQuery upon using the validator, a little bit of setting needs to be done inside the C# file as follows:

```
protected void Page_Load(object sender, EventArgs e)
{
    ValidationSettings.UnobtrusiveValidationMode =
    UnobtrusiveValidationMode.None;
}
```

Using the validator, it will check whether the input is already inserted. Otherwise, error message will be shown as in the sample below:



Login

Enter your ID and password

ID: admin

Password:

Please enter your password!

Copyright@BIE33103 1-2023/2024

Exercise:

1. Write the code for Cancel button, when user click on Cancel, text inside and ID and Password TextBox will be erased.
2. Use at least **ONE(1)** more validator wherever appropriate.
3. Create **Contact Us** page and its C# coding.

Instruction for submission:

- Your lab report must be in pdf.
- Copy your code program in asp.net, C# code in codebehind and screenshot the output displayed in the browser.
- Submission at **AUTHOR** (Tab Individual Activities).
- All work is to be done on an individual basis.
- Duration: 1 week only.