**WEB FORM**

1. **Create a web page for login process.**

**Description:**   
Create a Web application following the requirements given below

* Create a web page in the web application for Login process.
* Create a Heading in the web page whose name should be ‘LOGIN’
* Create Label & textbox sets, one for UserName & one for Password.
* Create Label & checkbox for ‘Remember password’ option.
* Create a button ‘Login’
* Validate if UserName and Password fields contain data or not. If not, throw message next to the control. Use red color style for the validation message

**Technical details:**

* Use Table based design for the web page
* Use H2 tag for page header
* Use Web server controls for textbox, checkbox and button
* Use Required field validator to validate the data

# **2. Display the list of departments in a college**

**Description:**

* Create a web page that has a dropdown that displays the list of departments in a college
* Print the selected value of the dropdown onto a label

**Technical details:**

* Use string array to hardcode the list of departments
* Set the dropdown items to be added in Page\_Load event

Use IsPostback event to check and DON’T re-bind the data

**DataBound controls**

1. **Display the list of Student names with their Department in a Grid.**

Description:

In continuation of requirement on Login page, the Create a Web application following the requirements given below

On ‘Login’ button click in the Login page, redirect user to Home page

On page load, display list of Student names and their Department names on a Grid

Create an XML file in the application code that contains the student detail. This has to be set as the source for the grid

There should be a column in the grid to have a link ‘Update’. On clicking the link, textboxes should appear in all of the cells where the student data is displayed for user to update the data

Technical details:

Use Response.Redirect to navigate to Home page

Create XML file with Student detail. Sample XML provided can be utilized

Fill & display data using Gridview

Use XML datasource to bind data to Gridview

Sample XML:

<Students>

<Student>

<Name>Alok</Name>

<Department>ECE</Department>

</Student>

<Student>

<Name>Nadeem</Name>

<Department>IT</Department>

</Student>

<Student>

<Name>Sooraj</Name>

<Department>EEE</Department>

</Student>

<Student>

<Name>Shiva</Name>

<Department>CSE</Department>

</Student>

<Student>

<Name>Nirmala</Name>

<Department>CSE</Department>

</Student>

<Student>

<Name>SaiKumar</Name>

<Department>CSE</Department>

</Student>

</Students>

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**4.Web page to update the displayed Student details**

Display the list of Student names with their Department in a Grid. There should be a column with Update link to update the data

Description:

In continuation of requirement on Gridview with XMLDatasource, the Create a Web application following the requirements given below

There should be a column in the grid to have a link ‘Update’. On clicking the link, textboxes should appear in all of the cells where the student data is displayed for user to update the data

On clicking the Update link, the data in the textboxes has to be read and concatenated to be displayed on top of the page

Create a link ‘Cancel’ next to ‘Update’ link. On clicking the Cancel link, the textboxes created for Update, should be removed and the grid cells should be displayed

Technical details:

Use Gridview events, editItemIndex to handle Update and Cancel events

Use Response.Write to write the data on the web page

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**4. Server and client side session Management**

**5.Logged in user name to be displayed across pages of web application**

Description:

In continuation of requirement Login page, the Create a Web application following the requirements given below

On ‘Login’ button click, add the Username provided in the relevant textbox to Session

Redirect the user to Home page

On loading Home page, access the username saved in Session. Display it on the top left corner of the page in the following format. ‘Welcome [user name]’

Technical Details:

Use Session variable to set and get the User name

Use Response.Redirect to navigate to Home page

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**6.User logged in time to be captured on all screens of web application**

Description:

In continuation of requirement on Login page, the Create a Web application following the requirements given below

On ‘Login’ button click, create a Hidden field and set the current datetime to it

Redirect the user to Home page

On loading Home page, access the logged datetime captured in hidden field. Display it on the top right corner of the page in the following format. Logged in at []

Technical Details:

Use Input type hidden and display the date in dd/mm/yyyy hh:mm:ss am/pm

Use Response.Redirect to navigate to Home page

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**5. ASP.Net feature for web pages**

**7.Create a reusable control to be used in a Web application**

Description:

In continuation of requirement on Login page, create a Web application following the requirements given below

Create a reusable web user control with the content of ‘Login’ control

Create a new web page

Add the web user control, created in the step mentioned above, to the newly created web page

Remove the content from existing login page and add the web user control to it

Verify if the existing login & newly created page load the same login controls or not

Login page should load when the application starts

Technical details:

Use the Web user control

Set the master page of the web user control to new web page

Set the Login page as Startup page thru the Solution explorer

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**8. Web page to display Student details**

Display the list of Student names with their Department in a Grid.

Description:

In continuation of requirement on Login page, the Create a Web application following the requirements given below

On ‘Login’ button click in the Login page, redirect user to Home page

On page load, display list of Student names and their Department names on a Grid

Create an XML file in the application code that contains the student detail. This has to be set as the source for the grid

There should be a column in the grid to have a link ‘Update’. On clicking the link, textboxes should appear in all of the cells where the student data is displayed for user to update the data

Technical details:

Use Response.Redirect to navigate to Home page

Create XML file with Student detail. Sample XML provided can be utilized

Fill & display data using Gridview

Use XML datasource to bind data to Gridview

Sample XML:

<Students>

<Student>

<Name>Alok</Name>

<Department>ECE</Department>

</Student>

<Student>

<Name>Nadeem</Name>

<Department>IT</Department>

</Student>

<Student>

<Name>Sooraj</Name>

<Department>EEE</Department>

</Student>

<Student>

<Name>Shiva</Name>

<Department>CSE</Department>

</Student>

<Student>

<Name>Nirmala</Name>

<Department>CSE</Department>

</Student>

<Student>

<Name>SaiKumar</Name>

<Department>CSE</Department>

</Student>

</Students>

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**5. Configuration Settings**

**9.Use Web application configuration settings for displaying application name in web page**

Description:

Create a Web application following the requirements given below

Create a key in Web application configuration file to store application name

Access the key in configuration file to fetch the application name

Set the key value in a variable

Use the variable to set the Application name in the web page as Heading

Technical details:

Refer System.Configuration DLL

Set key value in AppSettings section of Web.config

Access the key using ConfigurationManager.AppSettings

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**10.Use application configuration settings to show custom error page**

Description:

Create a Web application following the requirements given below

Create a web page that has a button

On clicking the button, an error has to occur

Use the Web application configuration setting to set the page to be redirected on error

Technical details:

Use customErrors attribute in Web.config under System.Web node

**7. Web Services - Features and usage**

**11. Web page to update the displayed Student details**

Display the list of Student names with their Department in a Grid. There should be a column with Update link to update the data

Description:

In continuation of requirement on Gridview with XMLDatasource, the Create a Web application following the requirements given below

There should be a column in the grid to have a link ‘Update’. On clicking the link, textboxes should appear in all of the cells where the student data is displayed for user to update the data

On clicking the Update link, the data in the textboxes has to be read and concatenated to be displayed on top of the page

Create a link ‘Cancel’ next to ‘Update’ link. On clicking the Cancel link, the textboxes created for Update, should be removed and the grid cells should be displayed

Technical details:

Use Gridview events, editItemIndex to handle Update and Cancel events

Use Response.Write to write the data on the web page

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**12. Logged in user name to be displayed across pages of web application**

Description:

In continuation of requirement Login page, the Create a Web application following the requirements given below

On ‘Login’ button click, add the Username provided in the relevant textbox to Session

Redirect the user to Home page

On loading Home page, access the username saved in Session. Display it on the top left corner of the page in the following format. ‘Welcome [user name]’

Technical Details:

Use Session variable to set and get the User name

Use Response.Redirect to navigate to Home page

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**13. User logged in time to be captured on all screens of web application**

Description:

In continuation of requirement on Login page, the Create a Web application following the requirements given below

On ‘Login’ button click, create a Hidden field and set the current datetime to it

Redirect the user to Home page

On loading Home page, access the logged datetime captured in hidden field. Display it on the top right corner of the page in the following format. Logged in at []

Technical Details:

Use Input type hidden and display the date in dd/mm/yyyy hh:mm:ss am/pm

Use Response.Redirect to navigate to Home page

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**14. Create a reusable control to be used in a Web application**

Description:

In continuation of requirement on Login page, create a Web application following the requirements given below

Create a reusable web user control with the content of ‘Login’ control

Create a new web page

Add the web user control, created in the step mentioned above, to the newly created web page

Remove the content from existing login page and add the web user control to it

Verify if the existing login & newly created page load the same login controls or not

Login page should load when the application starts

Technical details:

Use the Web user control

Set the master page of the web user control to new web page

Set the Login page as Startup page thru the Solution explorer

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**15. Set the mode to ‘On’ or ‘RemoteOnly’**

Create a HTML page to load static content with message ‘An error has occurred. Please try after later!!’

Set this page url relative to the root path in defaultRedirect attribute of customErrors node

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**16 .Create Virtual directory for a web application and view a web page**

Description/Technical details:

Create a Virtual directory in IIS with the name ‘StudentDetail’ and map the folder where the student details application created in previous steps

Verify if the web pages are listed in IIS or not

Check if anonymous authentication is set for the Virtual directory or not

Browse the login page and verify if the page is loading fine or not

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**17. Create a Web Service to return string data**

Description:

Create a web service with ‘Hello world’ to be returned

Reference the web service in a Web application

Create a web page with only a Button in it. Button text should be ‘Get Web service data’

Create a Label, below the Button, in the Web page to display the data obtained thru the Web service

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.

**18. Create a Web Service to return string data and consume the data returned by it in a web application**

Description:

In continuation with requirement on Web service creation, please create a web application based on the following requirements

On clicking the Button in the web application ‘Get Web service data’, as created in the Business case 8, Web service should be hit and the data obtained should be set to the Label’s text

Technical details:

Add web service reference to the web service in the web application

On button click event, hit the web service

Set the web service data to Label text

NOTE: eBox code submission not required. To be implemented using Visual studio and manually verified by the SME.