

The Digital Upscale Manager for Ballgames Site (3)

Objectives:

- Develop a multi-page website
- Use basic server controls

Overview:

For Project 03, continue adding new functionality to our site.

What to do:

Step 1 – Revisit Project 02

Be sure to read the comments in the Gradebook for the exercises so far and be sure that your existing site is up to date in meeting these requirements. The biggest concern (this is a general statement for the class and may or may not apply to you specifically) is the formatting of the site.

Many of you have been using the generic style sheet that we use in our weekly chapter exercises. Please use a little style and try to develop a style sheet that's attractive ... and different from what we've been using.

Step 2 – New Features

Once you are comfortable that your Project 02 work is up to speed, then start concentrating on the new material.

Working with Incidents

Unfortunately there times when our customers will have issues with our products. For that reason, we'll want to develop a page (IncidentDisplay.aspx) to be used to list out the existing incidents. This page should (of course) use the same Master page as the others. In the content area, be sure to include:

- A DropDownList used to display the customers
- A DataList to display the incident(s) details

The DataList control must be set up as a table with the following data:

- A header with four titles:
 - Product/Incident
 - Technician Name
 - Date Opened
 - Date Closed

- Each incident record should be displayed as two rows such that:
 - The first row lists the Product name, the Technician name, the date opened, and the date closed (if any)
 - The second row will span all four columns and will list out the incident's description

Product / Incident	Technician Name	Date Opened	Date Closed
<Product Name>	<Technician Name>	<Date Opened>	<Date Closed>
<Incident Description>			
<Product Name>	<Technician Name>	<Date Opened>	<Date Closed>
<Incident Description>			

}

Incident #1

}

Incident #2

When the user selects a new customer, the DataList should update with that customer's incidents.

Be sure to apply an appropriate style so that the controls on this page are attractive with the other elements on the site.

Working with Products

From time to time, the products in our database need to be updated. Luckily we know an easy way to create a page to do just such a thing. Create a page called Products.aspx that will display the contents of the products page in a GridView control. Be sure to list all of the fields from the table and sort the records based on the Product ID.

Provide the user with the ability to update product information (everything except the Product ID, which cannot be edited) or delete the product entirely. Include data validation so that the user is required to enter a value for the name, version, and release date. Also be sure that the date entered is valid. Use validation controls, as well as a ValidationSummary control (so that you can just use a * next to the location of an error and can include the full details about the error elsewhere on the page).

Provide the ability to add a new product. Because this cannot be done in the GridView, you'll need to include a set of labeled TextBox controls, one for each product field, as well as an 'Add' button. Include data validation for each of these TextBoxes. Once the product is added to the database, be sure to clear out the TextBoxes so that the user can enter another new product, if desired.

Don't forget to include appropriate exception handling, just in case something goes wrong when updating an existing product or attempting to add a new one. If something does happen, do not let the page crash. Instead, handle the exception and display an appropriate error message on the page to let

the user know what happened. This error message should disappear once the user has either corrected the action, or tried something different.

Note: Because of referential integrity, you shouldn't be able to delete any of the products currently in the table (because these are referenced in other records in the database). However, I will be adding a new product and then attempting to delete it, which should work (because there won't be any other records in the database that use this product)

Working with Customers

Our customers are the most important part of our company, therefore it is critical that we keep accurate records of them in our database and provide a mechanism for viewing and modifying these records. For that reason, develop a page called CustomerMaintenance.aspx. This page will need to have:

- A GridView control to display the name, city, and state of all customers in the database, initially sorted by customer's name (yes, it will sort them by first name, that's OK). Be sure to display only 6 records at a time and allow the user to sort by any of the three columns. This GridView should also allow the user to select a particular customer.
- A DetailsView control that will display all of the details about the currently selected customer. This DetailsView control should also allow the user to edit the current customer, delete the current customer, or add a new customer.

Be sure to validate the user's input so that all fields are required. Also be sure to handle any exceptions that may arise in order to keep the page from crashing.

Again, because of referential integrity you'll only be able to delete a customer that doesn't have any records anywhere else in the database.

Updating the Navigation

Because we have added 3 new pages, you'll need to modify the site's navigation. Be sure to place these pages in a new section called "Maintenance" as shown below:

- Home
 - Customer Support
 - Complete Customer Feedback
 - Contact Us
 - Administration
 - Display Customers
 - Display Contacts
 - Maintenance
 - Maintain Customers
 - Maintain Products
 - Maintain Incidents
 - Site Map

IMPORTANT

We've made a big deal about style / formatting and repositories in this course (and previous courses) and I expect that you'll continue to follow those guidelines here as well. The grading guide will be based on functionality and style points will be 'taken off the top'. If you have questions about what is expected of you for this, please ask early!

When you are finished and have tested the site to be sure that everything works correctly, please close Visual Studio and use 7-Zip to zip -the entire- Solution folder. Please use 7-Zip to create a file with the extension .zip. Also be sure to include the URL of your online repository in the Comments section on your submission.

Note that Visual Studio is going to create a lot of files and folders and I need them all to run (and grade) your work. It's probably worth you taking a minute out of your day and downloading your Zip file to a new location and reopening it to be sure what you think is there, actually is there.

Upload that Zip file to Moodle.