

Final Project #3– Creator and Viewer for Clickable Code Viewing

Version 1.1

Due Last day of Class

Purpose:

For this project you will build a website and web service that support building and viewing clickable source code views for C++ or C# in which the view initially is presented showing only the namespace(s), class names, and, for C++ the names of global functions. When a class name is clicked it shows all its declared data and functions with no bodies. If a function is clicked its body becomes visible. We can also provide controls that, when clicked, show the initial commenting and test stubs. All clicking show toggle the view, e.g., show then hide then show ...

The way we do that is to wrap the class bodies and function bodies in div elements that have styling to toggle their display view between none and normal. The entire code file should be wrapped in `<pre>...</pre>` tags to preserve whitespace structure.

Requirements:

The requirements for this project are to provide:

1. A website that supports:
 - a. Modifying a C++ or C# source code file to become clickable as described in the Purpose section. Since code analysis is not an essential part of CSE686 – Internet Programming content your creation view may allow the user to select an area of text, e.g., class body or function body, and make it clickable. And repeat that process until the entire source file is appropriately clickable. This view should also allow the user to edit clickable areas by removing or changing the clickable regions.
 - b. Uploading files to be published and deleting files no longer of interest.
 - c. A list view showing all of the published file folders and a list view showing all the files in a single folder. This view should support the selection of files to view and/or edit.
 - d. The ability to add and delete code folders.
 - e. The ability to select a sequence of files from the lists in item c., above, for subsequent operations. These operations are:
 - i. Defining dependencies by clicking on the parent and the child in a dependency relationship.
 - ii. The ability to view in dependency order, defined by a depth first search of the dependency tree, or in the order of selection.
 - f. Downloading an XML file that defines the folders and files selected in item c., above.
 - g. Three roles:
 - i. User who may only select and view the files in dependency order or selection order.
 - ii. Developer who may create folders and create, edit, and view files and create dependencies, as described above. One developer cannot edit another developer's files.
 - iii. Administrator who may delete files and folders and assign users to roles.
2. An Asp.Net WebApi web service and WPF client that supports:
 - a. Uploading C++ and C# files and folders, and downloading clickable source code files.
 - b. Downloading files defined by an XML file generated by a user of the website.
 - c. Support for viewing the text of files but is not required to support clickable web viewing.
3. The website will provide:
 - a. Home view that shows a user what the site does and provides a login facility.
 - b. Views that support selection and editing as defined in 1.a and 1.c above.
 - c. A help page with instructions for the various site activities.
 - d. An administration view that supports the deletion and archiving of published code web pages.
 - e. I expect that you will need a few more additional views.
4. You are required to implement the website in Asp.Net MVC and the web service as and Asp.Net WebApi.
5. Login access that forces a link to any page to verify that you are a valid user, by checking the current session to see if you are currently logged in, otherwise redirecting you to a login page. You may use the Asp.Net provided login code, Asp.Net MVC login code, or develop a user control for this that you will make part of every page.
6. Your site **shall** use:
 - a. Both SQL server database(s) and XML file(s), of your design, to store state of the site, as described above.
 - b. Use LINQ to access state data.
 - c. Features unique to HTML5 for some parts of your site.

- d. Pages using each of the models: HTML with JavaScript and CSS, Asp.Net, and Asp.Net MVC. Note that you may use more than one project for your site.

You are expressly requested to refrain from using site editors like Dreamweaver, FrontPage, or other Web designers. You may use any of the facilities of Visual Studio 2013 or any other text editor of your choosing.

For the final demo, you will be asked to host your site in IIS on your own machine and do all navigation via links provided on the pages of your site¹.

Please Note:

Part of your grade is based on designing a pleasing website to meet these requirements. I will not specify the details of that design. **You** must decide the visual aspects and details for each page.

¹ That means you're not supposed to navigate through a directory structure to find a page.