Milestone 5: Home Controller and View

CPS 353: Internet Programming
Web Development Project
Due Sunday, November 1, 2015 by 9pm

Introduction

The goal of this milestone is to deliver the home page mock content completed in a previous milestone up to a database driven website. To accomplish this, the database developed milestone 4 will be used. The details are below.

Specifications

Create asp.net we application project

Create a new asp.net web application project using the MVC type. This will generate the folder structure as well as provide a basic template for the home controller and the views.

Update the layout and view files

Update the layout file so that it contains the framework that you created for your home page mock up. This should contain the html tag, the head and body tags as well as links to the CSS and js files you will need. Put your CSS and image files in the content folder and your js files in the scripts folder.

The main content from your mock home page should be placed in the views/home/index.cshtml file.

Create a partial view in the shared folder called _Product.cshtml. This should have the product details shown on the home page. Include this partial view in the index.cshtml using @Html.RenderPartial or some other similar method.

Create the entity models, repository and service

Add an edmx file to your solution that is generated from your database. Add in the IRepository and GenericRepository from the Controllers and Views zip file linked on our site. You may have to change the namespace at the top of the file from FF to the namespace of your project.

Add a product service called ProductService that creates the repositories that you will need to get product data. This service should have a public method called GetFeaturedProduct which calls the repository to get the product where the featured product flag is true. Your service should implement an interface for all the public methods it has.

Update the Index method in the home controller

Update the Home controller to have a private member variable of type IProductService. Add a parameterless constructor to the home controller if it is not there. In here, create a new instance of the ProductService, passing in the db context. In class this was called FFEntities. Yours will be something different. Set your private member variable equal to this new object. Also add a constructor that takes an IProductService and set the private member variable equal to what was passed in. We did this in class as well.

Update the Home controller's Index method so that it gets the featured product from the service, maps the product to a new object you will create called ProductViewModel. You can send this to the view, or you can make this a property of another viewModel if you have additional data to send to the View (like we did in class with the Team and Owner models). Send your viewmodel to the Index view.

Update the views to use the model

Update the index view to require the same type of viewmodel that you are sending from the controller. Update the RenderPartial call to take a ProductViewModel. Update any static product content with content from the viewmodel so that if we change the featured product flag in the database, a different product will show.

As you do this milestone, you may find that you need to update the database schema you made in milestone 4. If you do, please update the sql files as well, since these will be part of what you include when you submit your milestone. I have videos on the course site showing how you an adjust your schema. The videos show a database project as part of your solution. You can do it this way, or make changes directly to sql server like I did in this last class. In either case you will need to:

- Capture the changes in a sql file
- Update your entity data model to get the changes made in the database.

Required files

Zip up and turn in your entire solution along with the sql files. Also, include sql for inserting product and category data to make testing easier.

This assignment is due November 1st by 9pm.