Module 3 Day 6

MVC Controllers - GET

What makes an application?

- Program Data
  - ✓ Variables & .NET Data Types
  - ✓ Arrays
  - ✓ More Collections (list, dictionary, stack, queue)
  - ✓ Classes and objects (OOP)
- Program Logic
  - ✓ Statements and expressions
  - ✓ Conditional logic (if)
  - ✓ Repeating logic (for, foreach, do, while)
  - ✓ Methods (functions / procedures)
  - ✓ Classes and objects (OOP)
  - ❖ Frameworks (MVC)
- Input / Output
  - User
    - ✓ Console read / write
    - ✓ HTML / CSS
    - ❑ Front-end frameworks (HTML / CSS / JavaScript)
  - Storage
    - ✓ File I/O
    - ✓ Relational database
    - ❑ APIs

Notes

- Bad news first…
  - Pairs every day
  - Individual M, Tu & Th
  - Pathway at least 4 days
- But the good news…
  - Review Day Friday
  - Happy Hour Friday

Dependency Injection

- Inversion of Control (IoC) pattern
  - We write the logic, but instead of creating the things we need, some dependencies are handed to us
  - Removes the task of creating DAO's in the Controllers
  - Tells the framework to create and pass DAO's into the controller
- Real-world example:
  - Playing music: music box vs. CD
- ASP.Net Core is called the *DI Container* (Dependency Injection Container)
- **Decouples** Controllers from specific DAOs, making unit testing possible
- https://docs.microsoft.com/en-us/aspnet/core/fundamentals/dependency-injection?view=aspnetcore-2.2

HTML Forms Tutorial

Element	Attribute	Value	Notes
Form	Action	A url	Submitted form is “sent” to this URL
	Method	GET	Http GET
		POST	Http POST
Label	For	Id of input field	Associates label text with an input field
Input	Name	someName	Name of the field. If omitted, data for this field will not be sent
	Value	Some value	Initial value for the field
	Type	Text	A text box
		Radio	A radio button (select one of a group). All buttons in the group have the same name
		Hidden	Not shown to the user, but comes back to the server on submit
		Submit	A button which submits the form when pressed
Select			Dropdown list. Filled with Option sub-elements
Option	Selected		Dropdown values
TextArea	Name, rows, cols		A multi-line text box
Button	onClick		A button to run client logic (more later)

- https://www.w3schools.com/html/html_forms.asp
- https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Your_first_HTML_form

GET vs. POST (table – when to use)

GET	POST
Form data sent in QueryString (URL)	Form data sent in Request Body
Can be bookmarked	Cannot be bookmarked
Visible as clear text in address line and history	Not visible in address line or history
Length of URL limited ~3000 characters	Unlimited Body size
Use when the user may want to execute it again (idempotent). Should not modify data on the server.	When same form data should not be submitted twice (data changes on the server)

Model Binding

- Maps incoming Request data to Action parameters
- Can be simple (int, string) or complex (object) types
- Where data comes from:
  - Query string
  - Form fields
  - Route values (i.e., the URL, such as id)
- Where data goes
  - Action parameters which have the same name as the QS or Form field
  - Public settable properties of Action parameters types, which have the same name as the QS or Form field
  - Case-insensitive
- The complex parameter type must have a default constructor to work

ASP Tag Helpers

- asp-controller (<Form>)
- asp-action (<Form>)
- asp-for (<Input>)
  - Associates an input field with a Model property
- asp-for (<Label>)
  - Adds label text and associates it to the input field
  - Works with [Display(Name=“”)] attribute on Model
- asp-for (<Select>), asp-items (<Select>)
  - Generates a dropdown list from the model