Week 3 curriculum

* HTML Fundamentals (Elements, Attributes, Doctype, HTML 5)
  + HTML – Hypertext markup language Defines Webpages. Responsible for structure & meaning and bring together all the different pieces(CSS, JS)
    - HTTP Request – Get Post (just a url you’re sending to the server)
    - HTTP Response -
  + Elements – the root element or document element(the one containing all others) **<html lang=”en”></html>**
    - **<head></head>**Head element contains metadata about the page(how the page interpreted and adds some extra info)
      * <meta charset=”UTF-8”>Metatag has no closing tag
      * Title tag
    - Body
    - P – paragraph
    - H1- section header
    - Images
    - Links <a href=”google.com”>Text</a>
    - Lists <ol>
    - Unordered Lists <ul>
    - Definition list for key value pairs
    - Formatting elements <b>boldface</b>(strong) <i>italic</i>(emphasis) <u>underlined</u>
    - Div element(generic block element) short for division
    - Span element(generic inline element)
    - Some elements are inline and others are block
    - Line breaks <br>
    - Audio
    - Video
    - Canvas – supports graphics primitive
    - Header
    - Main
    - Select- drop down list (option value) selected is default one to show
    - Article
    - Aside
    - Footer
    - Form
    - Field Set
    - Nav – navbar with links to different parts of the page
    - Tables <table><thead><th></th></thead></table>
    - Block elements grab some vertical space to themselves
    - Inline don’t, they can be next to each other
  + Attributes
    - Always a key/value pair. Selected = “selected”
    - Attributes to put on a form(get, post)
      * <form method = “GET” action =”#”></form>
      * Get(puts form data in query string)
      * Post(puts form data in request body)
    - Action: what url to send the data to.
    - Inputs go into forms
      * put input tags into label tags(take for att.)
      * <input type=”checkbox” name = “checkbox1”>
      * <input type =”submit”>
    - “alt” (on img) is an accessibility attribute to show what image is about as placeholder text when img can’t be loaded
    - Global attributes are attributes that any element can have
      * Id: identifier that should be unique on the page
      * Class: identifier that can be shared with others, you can have many classes on one element, just separate them with spaces
        + We give elements IDs and classes mostly so that CSS and JS can target them to give them custom appearance and behavior.
  + Doctype – every HTML 5 document needs to begin with the proper HTML 5 document type declaration(**<!DOCTYPE html>**) Not an element
  + HTML 5 – WHATWG(living standard)
  + XHTML – similar to XML
  + SEO(search engine optimization)
* Multimedia (Audio, Canvas, Video)
  + Audio
  + Canvas
  + Video
* Forms (Input, Validation)
  + Input
  + Validation
* CSS Fundamentals (Rule, Property, Cascade, Inheritance, Box-Model, Media-Queries, Positioning, Responsive-Design, Values-Color, Values-Size, External, Internal, Inline)
  + CSS – Cascading Style Sheets. Responsible for the layout/appearance. All about the visuals. Java Script is responsible for the behavior.
  + Rule
  + Property
  + Cascade
  + Inheritance
  + Box-Model
  + Media-Queries
  + Positioning
  + Responsive-Design
  + Values-Color
  + Values-Size
  + External
  + Internal
  + Inline
* Selectors (Class, Id, Omni, Parent-Child, Parent-Descendant, Pseudo-Classes, Pseudo-Elements, Sibling, Tag)
* MVC Concepts (Model, View, Controller)
  + Model
  + View
  + Controller
* Request Lifecycle (DNS, Response, Request)
* Controller (Actions, HTTP Verbs)
  + Actions
  + HTTP Verbs
* Model (Data-Annotations, ViewModel)
  + Data-Annotaions
  + ViewModel
* View (Partial-Views, Strongly-Typed, Weakly-Typed, Layout, Razor, ViewData, ViewBag, TempData)
  + Partial-Views
  + Strongly-Typed
  + Weakly-Typed
  + Layout
  + Razor
  + ViewData
  + ViewBag
  + TempData
* Routing (Global, Controller, Action, Parameter-Route, Parameter-Query)
  + Global
  + Controller
  + Action
  + Parameter-Route
  + Parameter-Query
* Validation (Server, Client, CSRF, Anti-Forgery)
  + Server
  + Client
  + CSRF
  + Anti-Forgery
* Testing (Behavior Driven Development, Mock, Acceptance-Testing)
  + Behavior Driven Development
  + Mock
  + Acceptance-Testing
* Filters (Action, Authorization, Custom, Exception, Response)
  + Action
  + Authorization
  + Custom
  + Exception
  + Response
* Helpers (Custom, HTML, Tag)
  + Custom
  + HTML
  + Tag
* Model Binding
* Entity Framework (Code-First Approach)
* Dependency Injection (Singleton, Scoped, Transient, From-Services)