

WEB 3700 – ASP.NET Web Development (Fall 2019 Version)

Last Updated 8/19/2019

General Information



Section Taught By: [Dr. Richard Fry, Professor of Computer Science](#)

I have been a faculty member of Weber State University's School of Computing for more than 19 years, and although I am very friendly and approachable, I want to keep our relationship in this course professional and **I prefer to be addressed by my title of Professor or Dr. Fry rather than my first or my last name only.**

Course Website: <http://canvas.weber.edu>

Live Classroom Instruction: Mondays and Wednesdays, 11:30AM – 1:20 PM (TE 109C)

Instructor Office Hours

Tuesdays and Thursdays 8:30AM -11AM (EH 383) walk-in welcome, but appointments have priority. Virtual online appointments also available upon request.

Direct Email (Quickest Response): rfry@weber.edu

PLEASE DO NOT USE THE CANVAS MESSAGING SYSTEM TO SEND E-MAIL. IT'S DELAYED.

Accommodations

Any student requiring accommodations or services due to a disability must contact Services for Students with Disabilities (SSD) in room 181 of the Student Services Center. SSD can also arrange to provide course materials (including this syllabus) in alternative formats if necessary.

Course Description

This course provides an introduction to Razor – a relatively new but very popular framework for building dynamic, data-driven web applications with clean separation of concerns. Built on Microsoft's newest platform called ASP.NET Core, we will create Razor pages and web applications in this course that provide full control over rendered HTML as you program server side code. Razor Pages are suitable for all kinds of developers from beginners to the enterprise level. Razor is based on a page-centric development model, offering a familiarity to web developers with experience of other page-centric frameworks such as PHP. It is also relatively easy for the beginner to learn. It makes use of the popular C# programming language for server-side programming, and the easy-to-learn Razor templating syntax for embedding C# in HTML markup to generate content for browsers dynamically.

Course Fees

Course fees for all School of Computing majors are designed to cover the costs of lab equipment maintenance and replacement including desktop and server computer systems and software; consumable materials and supplies; and support for lab aides, student tutors, and online instructional resources.

Course Objectives

- Learn to create Razor Web Page Applications and write code that implements business logic with database models, methods, properties, and events.
- Learn to add and to manage user interaction, update models, and select and return Views.
- Learn to implement a consistent look and feel, including corporate branding, across an entire web application.
- Learn to use partial page updates and caching to reduce the network bandwidth used by an application and accelerate responses to user requests.
- Learn to integrate client side programming within the Razor pages to enhance the User Experience.
- Learn to package and deploy an ASP.NET CORE 2.2 web application from a development computer to a web server for staging or production.

Course Delivery

This is a 4 credit hour course, which means all students are expected to engage in learning new material at least 3 hours per week, with 1 extra hour reserved for open lab time and help. This can easily be accomplished by either participating in class the entire 4 hours each week. Beyond learning new material each week, students are also expected to spend approximately 4 to 6 hours more each week doing homework. **So, the typical weekly commitment for this 4 credit course is approximately 10 hours per week** (that's why 16 credit hours is considered a full academic load – equivalent to 40 hours per week). **Note:** This is a BRAND NEW CLASS, and as such it is subject to changes as we go. Please be patient and flexible, as we're all in this together.

Course Materials

There is not an official textbook required for this course. You will be expected to utilize (and share) with others resources from class and the web.

The following software is required. [Visual Studio 2019* Community Edition](#) or the full Professional or Enterprise Editions - available free through the [Microsoft Academic Alliance](#). Note there is also a special version of [Visual Studio for Mac](#) (there are minor differences and small workarounds, but in the past students have been able to use this version with very few issues). The PC edition is the preference. During setup, you'll be asked to select the workloads you want to install. Make sure you select ASP.NET under the Web and Cloud section as well as Data Storage and Processing in the data section. If you already installed Visual Studio prior to reading this, you can go back and run the Visual Studio Installer (separate program) to go back and modify your installation.

**Important: Please install the Git Hub extension for Visual Studio <https://visualstudio.github.com/>

*VS 2019 version is preferred (as that's what I use for demos), but if you use 2017, make sure you update to [.NET Core 2.2 SDK](#)

Discussion Forum

Due to the large number of students participating outside of the classroom, we utilize a 3rd party discussion tool called Piazza (accessible via Canvas). Students are strongly encouraged to use the Piazza discussion forum FIRST to ask questions and/or provide assistance to each other on assignments or any material related to the course. When posting a question, please mark the subject with a #hashtag to facilitate searching. Posting guidance and/or hints on this public class discussion forum is NOT considered a violation of the cheating policy, unless you blatantly post full solutions to an assignment or the public URL address to your website. The discussion forums are for students to help each other, as I cannot always be on-line 24/7.

Grading Criteria

- **Attendance** (2 free absences allowed. Any other – excused or not is penalized) – **20%**
- **Weekly Tutorials** (Following along in class, and/or completing outside class) – **30%**
- **Challenge Activities** (On your own outside of class, with occasion open lab time) – **50%**

94-100 = A, 89.5-93.9 = A-,

86.5-89.4 = B+, 82.5-86.4 = B, 79.5-82.4 = B-,

76.5-79.4 = C+, 72-76.4 = C

***Minimum passing grade is 72%.**

Homework Due Dates

- Homework will typically be due 8-12 days after it is assigned.
- **I will not accept any late assignments under any circumstance, even an “emergency” (unless you use a token). Plan and work ahead!**
 - However, all students begin the semester with 5 free late “tokens” that provide up to five extra days to turn in any of your assignments late – no matter what the reason (or you can save them for extra credit at the end of the course). Tokens are AUTOMATICALLY APPLIED whenever an assignment is submitted late as follows:
 - 1 token is deducted for each (1 minute to 24 hour) late extension on any assignment.
 - **If you run out of tokens, all late assignments (regardless of reason) will receive an automatic zero. Tokens are intended for “emergency” extensions on late assignments, and you shouldn’t be used because you waited until the last minute to start an assignment. Life happens! Be prepared for the unexpected!**
 - **Unused tokens convert to extra credit at the end of the semester, so save them.**

Plagiarism and Cheating Policy

Although cheating has many forms, I generally consider cheating to be any attempt to claim someone else's work as your own. Any assistance provided and/or received on problem solving or programming assignments without being publically posted on our course discussion forum (so I know about it) is considered cheating...because I always assume you are doing your own original work. You are encouraged to assist other students whenever possible or cite internet website resources that provide help, but this help must be transparent, and posted via the discussion forum so everyone gets the same information. If you have any questions about this policy, please ask me.

WARNING: The School of Computing’s policy dictates that any verifiable evidence of student academic cheating, as defined and determined by the instructor above, will result in: 1) an automatic failing grade for the class and 2) a report to the Dean of Students that will include a detailed description of the student's dishonest conduct.

Tentative Topics Covered (SUBJECT TO CHANGE!! ALWAYS FOLLOW THE LIVE CALENDAR IN CANVAS)

<ul style="list-style-type: none">• Introduction to ASP.NET Core (and C#)• Razor Page Files<ul style="list-style-type: none">○ Razor Files○ Partial Pages○ Layout Files○ ViewImports File○ ViewStart File• Razor Syntax• Page Models<ul style="list-style-type: none">○ Handler Methods○ ViewData○ Action Results• Tag Helpers<ul style="list-style-type: none">○ Anchor Tag Helper○ Cache Tag Helper○ Distributed Cache Tag Helper○ Environment Tag Helper○ Form Action Tag Helper○ Form Tag Helper○ Image Tag Helper○ Input Tag Helper○ Label Tag Helper○ Link Tag Helper○ Option Tag Helper○ Partial Tag Helper○ Script Tag Helper○ Select Tag Helper○ TextArea Tag Helper○ Validation Message Tag Helper○ Validation Summary Tag Helper	<ul style="list-style-type: none">• View Components• Routing and URLs -Custom Route Conventions• Startup• Configuration<ul style="list-style-type: none">○ Global Error Handling And Logging○ Custom Error Pages○ Configuring HTTPS• Middleware• Working With Forms<ul style="list-style-type: none">○ Request Verification○ Uploading Files in Razor Pages○ Working With Checkboxes○ Select Lists in a Razor Pages Form○ Working With Radio Buttons• Validation• Model Binding• State Management• Caching• Managing Security With ASP.NET Identity<ul style="list-style-type: none">○ Customizing Identity• Using AJAX<ul style="list-style-type: none">○ Posting Forms○ Partial Page Updates○ Uploading Files○ Unobtrusive AJAX• Working with JSON• Publishing To IIS• Scaffolding• Advanced<ul style="list-style-type: none">○ Areas○ Custom Tag Helpers○ Dependency Injection○ Custom Route Constraints○ Parameter Transformers For Routes○ Filters○ Partial View To String○ Razor Class Libraries
---	---