Course Code/Course Title:

FSW104-CS Back End Foundations

Course Description: The Back-End Foundations course is an introduction to creating server-side web applications and services. Students will learn how to create server-side APIs and render websites. Language-specific concepts will be covered, outlining the nuances associated with the elective framework.

Course Length: 40 hours	**Prerequisites:** FSW102	**Proficiency Exam** ☐Yes ☐ No	
Course Start Date: **Course End Date:**	**Meeting Days/Times**		

Required Resources:
Minimum: PC (Windows) or MacBook laptop. 4GB ram, 256GB HD, Core i5
Recommended: PC (Windows) or MacBook laptop. 8GB ram, 256GB SSD, Core i5

This will become your programming rig. Every student will need their own laptop. We will be downloading programming resources to your laptop, and it will also house your coding projects.

Additional Resources:
Students are expected to supply notebooks, pens, pencils, highlighters, folders, ring binders, calculators, USB storage devices and other general supplies as needed to aid in the collection and storage of information in their courses

A. **For Classes Delivered in an Online Format (for approved courses and campuses).** Online courses are delivered via https://wozu.exeterlms.com in an asynchronous format. Students enrolled in online courses/programs are expected to spend an equivalent amount of time on task, as campus-based students, in meeting course objectives. For Online Courses the total expected hours required for completion of course objectives are identified on the syllabus as **Total Contact Hours** and reflect the sum of theory, laboratory, and outside hours.

Educational Objectives:

Upon successful completion of this Program, students will be able to:

1. Learn the basics of the request/response cycle
2. Learn how to render HTML in response to a request
3. Learn how to render JSON or XML in response to a request
4. Understand how to build a REST endpoint
5. Learn how to interface with a database system from a web application

Course Outline

Lessons:

1. **Introduction to Backend Foundations with ASP.NET:** Includes server-side web frameworks, HTTP protocol, web framework capabilities, setup, Postman, VSCode, extensions, MVC application
2. **HTTP & HTTPS Protocols:** Includes web clients and servers, URL's, HTTP and HTTPS protocols, domain names, IP addresses, DNS, Port number, resource path, HTTP message body, HTTP verbs, status codes, query strings and fragment identifiers, HTTP query string parameters, HTTP cookies, HTTP request headers
3. **Model View Controller:** Includes MVC Controllers, walkthrough, MVC views, tag helpers, view layouts, partial views, adding views, models, create a form, view files, Razor syntax features
4. **Databases:** Includes First database connection, walkthrough, create data model, Nuget package, add the database, pass the query, bind data to model
5. **Entity Framework:** Includes Database migrations, Create an ASP.NET Core MVC Application, walkthrough, Adding Migration & Creating the Database, Create, Read, Update, Delete, Reverse Engineer an Existing Database, Scaffold the Database, Generate Controllers and Views
6. **RESTful Web APIs:** Includes Web API Template, Web API Architecture, Building Your First Web API with .NET Core, Accessing Web APIs with Postman, Modify the API, Optional, Constraint, and Default Value, Web API and jQuery Front End
7. **Authentication:** Includes Security from Scratch, Securing User Data, Hashing Passwords, Checking Passwords, Identity Framework Application, Chain of Events, Creating a User
8. **Front End Integration:** Includes connecting to React, and Connecting to Angular
9. **Error Handling and Logging:** Includes unhandled Exception, App Configuration, Custom Exception Handling Page, Logging providers, Log Levels, Using a Logger
10. **Final Project**

Outline:

- **L1 Practice Hands On:** Create a new directory and use the CLI to create a WEB template project.
- **L2 Hands On:** Create a new directory and use the CLI to create a WEB template project. Add a name and display it on the page, if no name is entered make the default name equal to yours.
- **L3 Practice Hands On:** Using the PartyRSVP form, include links for navigation to the form, modify the .cshtml naming, and add style to all pages using Bootstrap and CSS.
- **L4 Hands On:** Using Postman, post a new track to the database. Create a model that matches the customers table and run a GET method that will return 20 customers. Create another model that matches the employees table and run a GET method that will return Employees hired before 2003.
- **L5 Practice Hands On:** You will select a new table of your choosing and create a new migration, new controller, and check the route works with Postman.
- **L6 Hands On:** Create a Web API project that takes data from a front end (Postman), passes the information to the Web API, and then stores the data in an SQLite database. The object passed by the API should be a model called Car.

- **L7 Practice Hands On:** Create a movie application that allows anyone to add movies to a list. If a user logs in, though, they should be able to create a playlist that only they can update, or delete.
- **L8 Hands On:** Take a .NET Web API and use it as the backend for your application. You will enable CORS and then use the application to serve data to your choice of front-end applications.
- **L9 Practice Hands On:** Create a few projects which will demonstrate the use of logging, configuration, and Web APIs.
- **L10 Final Project:** Final Project

Final Project:

Using C# and Web API create a Record Store. The record store will allow a user to create a playlist and view information based off tracks, albums, or artists. The user can edit or delete tracks from their playlist or create more than one playlist.