

# INTRODUCTION TO: WEB DEVELOPMENT

This outline is a work-in-progress, and may change in the future – medatech@medasf.org

MONDAY THROUGH FRIDAY 9AM-1PM ONLINE @ GOOGLE MEET

MEDA Cohort Spring 2020; Instructor Eduardo Garcia

## DESCRIPTION

This is an eight week course that introduces a student the student to back-end technologies and build basic web applications using back-end. This is an introductory course that will cover a couple of technologies involved with creating web applications and will break each technology individually. **If you miss a total of four days of class, you are automatically dropped from the class!**

## PREREQUISITES

While this is an introductory class to *computer programming*, there are a few things that you should have a foundational understanding of in order to have the best experience in this class. Keep in mind I do not cover any of these topics during class. You should have a basic understanding of math including addition, subtraction, multiplication, and division. Geometry-level math knowledge is *highly* recommended! The focus of this class is learning how to use code to react to users and execute functions based on the current situation, which means you should feel comfortable using either a Mac or Windows computer and include: writing syntactically-correct computer code, fixing code, and testing code. Please note that this is an online course, and you should have reliable access to a computer and the Internet.

## PENALTIES

- ➔ A total of four days missed from class is grounds for being dropped from the class. Attendance is crucial to your learning to write code is similar to learning Math, you miss one day and you will be completely lost the next day. I frown upon tardiness and if you arrive to class after 9:15AM you are marked absent. It is better to be thirty minutes early than thirty minutes late! If you are aware of future absences, please speak to me so we can work out the days missed.
- ➔ A total of three incomplete projects by their respective deadlines is grounds for being dropped from the class. There is no “late work” for this class and the student is responsible to schedule their work on projects outside of class time.

- ➔ Any type of disrespect toward other people or MEDA property may be grounds for being dropped from the class. Any offensive comments or arguments based on racial or religious background toward another person is grounds for being dropped from the class and the student will be asked to leave the class immediately.

## (OPTIONAL) RECOMMENDED BOOKS

The following two books focus on JavaScript and recommended as additional resources. Essential JavaScript has a higher learning curve but is invaluable to learning JavaScript for back-end.

- ➔ JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett
- ➔ Eloquent JavaScript by Marijn Haverbeke

## CLASS GOALS

- Understand how the web is built.
- The ability to creates web pages that can receive, send, and modify data.
- Have a clear understanding how computer code is written and handled.
- Learn the tools needed to a website back-end, including Git, Terminal, SSH, and HTTP Methods.
- Develop visual design for website and forms.
- Have a basic skill in writing HTML, CSS, and JavaScript.

## WEEK 1

Introduction to the MERN Stack

- ➔ What the MERN stack is and what each letter represents.
- ➔ Give an overview of each technology that goes into a MERN stack.
- ➔ Comparing the MERN stack to the LAMP stack.
- ➔ What we will expect to learn and do in the final 7 weeks.
- ➔ Review HTML, CSS, and JavaScript. Have a solid foundation on these as we will depend on these to use the upcoming technologies.
- ➔ Catch up and polish on previous Projects.

At the end of this week, we will have a quiz that will cover HTML, CSS, JavaScript, jQuery, and Terminal.

## WEEK 2

Creating, Reading, Updating, and Destroying files with Node.js

- ➔ What Node.js is and how it compares to the older LAMP stack.
- ➔ Setting up and using Node.js on your computer and other computers.
- ➔ Using Node.js to debug your JavaScript, and an introduction to nodemon and repl.
- ➔ Learn what npm is and how to use others people code instead of writing your own.
- ➔ Exploring built-in modules like fs and exporting your own.
- ➔ Using Node.js to create files using your JavaScript Code!

The project that we will get started on this week will be a To-Do list application that the user interacts with through the Terminal. This application will allow us to save and load data, which can be modified by the user.

## WEEK 3

Introduction to Express.JS, a server-side JavaScript Framework.

- ➔ Installing the Express.js NPM module.
- ➔ Setting up your Node.js projects to use Express.js.
- ➔ Understanding Express.js Routes.
- ➔ How to use SSH to use the Terminal on another computer!
- ➔ Project 6: A single web page application for your hobby guide

We will be updating our To-Do list application so that it can be loaded into a web browser and have the same functionality through a web page.

## WEEK 4

And introduction to MongoDB, the JavaScript Database solution.

- ➔ Why use a database over generic files.
- ➔ Learn how to create and use a MongoDB database.
- ➔ How MongoDB compares to older database standards (You don't need to learn another computer language!).

- ➔ Installing and using the Mongoose Node.js module.
- ➔ Learn how MongoDB stores data using flexible JSON-like documents.
- ➔ Learn how to use CRUD commands to modify your data on a MongoDB database.
- ➔ An introduction on how to structure your database for future use.
- ➔ ~~Create an official Entity Relationship Diagram for your database.~~

Prepare and plan a project to create a website that solves a common problem using MongoDB. We will be “collecting” data, and display facts based on existing data.

## WEEK 5

For this week we will be looking further into manipulating data to get information out of it. We will be looking at a public data set, setting up questions for this data set, and then creating code to answer these questions.

- ➔ Practice “cleaning” up data sets for our code and remove data anomalies.
- ➔ Create code that read but does not write to the dataset.
- ➔ Look into programming exception handling and dummy-proofing our code.
- ➔ Creating efficient code that will run through thousands of lines of data and converting them to objects for easy traversal.
- ➔ Testing if Input and Output for code reliability.

This weeks project will be selecting, questioning and executing a data set of our choice.

## WEEK 6

During week six we will be taking a look at React.js and Socket.io to create simple applications that will connect to a database.

- ➔ Introduction to Socket.io by creating a simple chatroom application which will allow us to understand two-way traffic with AJAX
- ➔ Introduction to React.js, a framework to quickly develop a web application using front-end and back-end.
- ➔ Understanding other frameworks available to us and why we use them.

## **WEEK 7**

For the final week of the online back-end class, we will think of an idea, plan and structure this project, and code the project. This project should include most technologies we have learned throughout front-end and back-end class.

## **NOTES**

---