Last Updated October 2021

Syllabus - Web Development - Fall 2021

Lecturer: Hunter Jorgensen (ahjorgen167@gmail.com)
Wednesday 6:00pm - 9:00pm PST
Location: Room 402, 225 Terry
TAs:

Name	Email	Public Github	NEU Github	Responsibility	Office Hour
Shaun Ho	ho.s@northeastern.edu	shaunhoshenteck	shaunhoshenteck	Lead TA	Tuesday 9:00AM - 11:00AM Zoom
Areal Jones	jones.ar@northeastern.edu	arealjones	arealjones	Discussions	Sunday 2:00PM-4:00PM Zoom (No in-person OH)
Bo Niu	niu.bo@northeastern.edu	bo-niu	boniu	Grade submission handling	Zoom Thursday 1:00PM - 3:00PM
Lukai Lin	lin.luk@northeastern.edu	archliner	lukai	Late days	Monday 1:00 PM - 3:00 PM Zoom (In-person OH available if needed)
Nandish Murugeshi	murugeshi.n@northeastern.edu	nandish291	nandishmurugeshi	Extra Credit labs	Friday 11:00 AM- 1:00 PM Zoom (No in-person OH)

Office Hours are also by appointment (email any one of us!)

Online Students: Please contact Hunter directly for information about the online recordings. Recordings are available for all students after class.

Summary and Objective

The purpose of this course is to understand core technologies, theories and practices in modern web development. At the end of this course, students should be familiar with broad trends in web development and understand the fundamentals of major technologies that support development practices and also peek at some of the more advanced technologies that affect the most popular websites on the planet. All of this will be reinforced with a series of take home assignments, lectures, in class labs, quizzes, online discussion boards and a final capstone project. The goal of the assignments and projects are aimed to be work that you would be comfortable to show future employers.

The trajectory of this course follows modules that build upon each other. First, we will make simple, static web pages using HTML, CSS and with some considerations on design. Then will start to make websites more interactive by introducing JavaScript and React (a JavaScript library). Once we are comfortable with client facing applications, we will develop a simple backend service using Node.js and Express, exploring how the backend piece interacts with the frontend. Finally, we'll end the semester understanding both how security (with OAuth and bcrypt) and databases (MongoDB) affect how to build our service. Other topics covered will include MVC, DOM, Redux, HTTP, REST, Sass, and more, time permitting.

This class is mainly geared for students who have no previous experience with web development but have some understanding of programming. If you have lots of experience with things like React, JavaScript and HTML/CSS, you may find this class boring but I am always willing to modify the parameters of the projects to fit your personal interests.

Development Environment

As you begin preparing for this course, I recommend setup the following technologies:

- An internet browser (Chrome and Firefox are recommended)
- An IDE or Text Editor (VS Code is the most fully featured IDE for web development; Sublime is recommended for if you prefer something more lightweight; if you have a personal preference, that is fine as well).
- Access to a terminal or command prompt (Linux and Mac have these built in; for Windows, I recommend setting up Bash for Windows and choosing Ubuntu, Fedora or Debian as your Linux Distribution during installation)
- A Github account (you may use a public Github account or the one provided by NEU)
- A way to push code to Git (either with the CLI or the GUI)

Coursework and Grading

Type	**Percent**
Project 1	10
Project 2	20
Project 3	20
Quizzes	15
Discussions	10
Participation	5
Mini Assignments	20

Your final grade will be determined through several different factors: assignments, quizzes, discussions, participation and a final project.

Projects:

There will be 3 projects in this class that will allow students to practice and gain experience with different parts of web development technologies. Project 1 can be done solo; project 2 and 3 may be done with one partner, but can be completed alone. The goal of each project is to provide you the opportunity to build something with the technology of the course AND have something demonstrable that you can show to employers. Many of the projects are based off of interviews that I have seen in the wild. Each project should take you about a week of effort. Note: This class completely supports the opportunity for students to learn subjects that they find valuable to them. In this way, I am open to you swapping one technology for another (i.e. using Vue.js instead of React) or changing the result of the project (i.e., turning a game into a different game or into an app with a similar amount of complexity). However, you MUST discuss this with the teaching staff before project submission so that we can ensure that the work is providing a worthwhile and meaningful challenge.

Mini Assignments (MA):

There are many mini assignments throughout the semester. The purpose of these assignments is to provide a low stakes space to try out some of the technology of the course and to get familiar with this technology: you can think of it like a trial run for the projects. Each mini assignment should only take you a couple of hours (i.e., no more than a single day.)

Quizzes:

There will be 3 quizzes in this course. Quizzes will be online and open book (i.e. - open internet), and are always timed. These quizzes will reinforce learnings and allow students to ensure that they completely grasp the material. The goal of the quiz is to simulate a slightly higher pressure environment similar as to what you would face in a job interview.

Discussions and Responses

Schedule and Details

Over the course of the semester, students will be asked to write 2 in-depth posts on modern web development, and 9 responses (not including your introduction.) Links to duplicate sources will be given half credit and any late submission will also be given half credit.

Participation and Bonus Points

It is important that students develop the confidence and knowledge to express their ideas and thoughts, though every student may do this differently. During class, there will be labs that allow students to experiment with technologies: if students participate during this section, this may help boost their assignment and quiz scores. Please know that scores can not go above 100%.

Schedule

Last Updated October 2021

Note that the schedule is flexible, but I'll keep this up to date as best as possible. Topics may shift or move depending on challenges or interest, but the list below is always the source of truth.

Last Updated October 2021

NEU Academic Calendar

Wk	Day	Topic	Work Due	Class Links
1	Sep 8	Mod 1 (Web Dev Basics) - Introduction, Basic Concepts, HTML		
2	Sep 15	Mod 1 - HTML Review, CSS	MA 1: Introductions Due Before Class	HTML Class Code
3	Sep 22	Mod 1 - CSS Review, Github Pages, Flex/Grid, Mobile Design	MA 2: HTML Due Before Class	
4	Sep 29	Mod 1 - Design, CSS/HTML Review	MA3: CSS Due Before Class; **Quiz 1**	
5	Oct 6	Mod 2 (Frontend Logic) - JavaScript, DOM/Web Components, Bootstrap	Project 1: Personal Website Due Before Class	
	Oct 13	**Class To Be Rescheduled**		
6	Oct 20	Mod 2 - JavaScript Review, Advanced JavaScript, Node Setup, React/Hooks	MA4: JavaScript Due Before Class	
7	Oct 27	Mod 2 - React Review, Redux	MA5: React Due Before Class	
8	Nov 3	Mod 2 - Redux Review, Class Components, Heroku Setup, React Router		
9	Nov 10	Mod 2 - JavaScript/React Review	MA6 Due Before Class; **Quiz 2**	
10	Nov 17	Mod 3 - API Concepts, Introduction to Node, Node Review, Introduction to Axios	Project 2: Battleship Due Before Class	
	Nov 24	**Thanksgiving Break - No Class**		
11	Dec 1	Mod 3 - Review API concepts, Fullstack App, MongoDB		
12	Dec 8	Mod 3 - Authorization, Encryption, Mongo Review, Browser Storage	MA7 Due Before Class; **Quiz 3**	

13	Dec 13	Finals Week - No Class	Project 3 Due By Midnight Dec 15	
	Dec 20	*Final Grades Due to NEU Before 6am*		

Late Policy

All the coursework in this class has a slightly different late policy, so please read this section carefully. Obviously, if there is some extreme emergency (accident, family issue, etc.) we can revisit these policies on a case-by-case basis but that will require contacting the lecturer or the TA's as soon as possible.

For *projects/mini assignments*, every student has 10 late days that may be used without question, each giving you another 24 hours. Late days need to be requested BEFORE the assignment is due. If you are working in a group for certain assignments, everyone will need to use one of their late days to be granted additional time. For assignments that are late and late days are not used, students will lose 5 points per day.

For *quizzes*, you will have to reschedule with the TA or lecturer the time to take the quiz before the start of the next class.

For *discussions*, if you are unable to write a post, then you need to contact another student and switch weeks with them. Late discussion posts and responses are worth half credit.

Special Lecture Topics:
Given time, we may dive into some additional and interesting ideas: Sockets, PHP, GraphQL, Angular, Vue.js, D3, Meter.js, Python/Flask, TypeScript, Server-side rendering, Web Worker, SASS, nginx, Web Assembly, etc.

Tips for Success

This course covers a wide variety of topics and ideas, so that even the strongest students may struggle. That said, I have noticed some trends that define the most successful students:

- Make sure to complete all discussions on time. This is often the cause of many lost points!
- Ensure that by the end of the semester the instructors know your name. We are happy to help debug, run test interviews, review resumes or just talk, but you need to initiate it. This is a big way we determine your participation grade.
- If you are stuck, spend only a bit of time (an hour or so) debugging and then come to the instructional staff. A day spent on a single bug, is a wasted day.
- Try to get in the habit of reading log and error outputs. Sometimes they don't help at all, but most of the time they can be useful. Look for filenames that come from your code

Resources
Reading on HTML/CSS/JavaScript: https://developer.mozilla.org/en-US/docs/Web
Documentation for React: https://reactjs.org/docs/getting-started.html
NodeJS: https://nodejs.org/en/docs/
Mongoose Documentation: https://mongoosejs.com/docs/guide.html
MongoDB: https://docs.mongodb.com/manual/introduction/

Academic Integrity
This course has no patience for any kind of cheating, fraudulence, etc. While collaboration is allowed, if you are copying or stealing another student's assignment (from this or previous semesters) you will be reported to OSCCR. If you suspect that another student is copying or stealing, you should report that to the lecturer immediately.

Please note that I will be using TurnItIn to check code for plagiarism, which compares your code not only against the code you write yourself but also against the code published elsewhere on the internet.