Lecture 1: Course Introduction

Course Introduction

Grade Breakdown (1/2)

Labs: 40%
- Labs will be weighted equally
- Labs will be given most weeks and cover the content learned that week.

Final Project Proposal: 5%
- Students will form groups and propose a final project to work on throughout the semester.

Project Pitch: 10%
- Each group will give a non-technical presentation of their project, explaining what use case it solves, who the audience is, and why it's a worthwhile project.

Grade Breakdown (2/2)

Database Proposal: 5%
- ◦ Each group will submit a proposal for their database collections and schema

Final Project Presentation: 10%
- ◦ You will give a technical presentation showcasing your project's features and technological decisions.

Final Project: 30%
- ◦ Each group will submit their project code and a database dump to be reviewed and graded based on what was promised in their proposal and delivered.

What will we be covering in this course?

In this course, we will be going through many fundamental web concepts and learning technologies related to them.

1. You will learn how to install and configure a modern web programming environment, from server to database. In our case, we will be using Node.js, MongoDB, and Express as our programming environment, database, and server respectively. You will also learn about many tools that you will be using as a web developer, such as Git.

2. You will learn how to do server-side programming. In our case, that means you will learn the JavaScript language, as well as good coding patterns in order to structure a web application. You will learn how to separate your code in logical ways that make sense and follow modern conventions.

3. You will learn how to use a modern database; in our case, this is MongoDB. You will learn what this database's strengths are, what its weaknesses are, and how to utilize it effectively.

What will we be covering in this course?

4. You will learn how to code for the client. Learning HTML and CSS will enable you to create a document that makes sense, both semantically and meaningfully to the human eye. You will learn how to use JavaScript to make your applications respond to your users' input and experience.

5. You will learn about web accessibility and the major hurdles that many people face using the web as it is today, how to identify issues that exist in a web page, as well as how to correct them.

6. You will learn advanced client-side programming techniques and how to leverage frontend tools that allow you to create incredibly dynamic web experiences.

7. You will learn about security issues in the web and how to minimize their risks.

8. You will create a market-ready database driven web-application, from start to finish, involving technical presentations

What will we *do* in this course?

In this course, you will:

◦ Complete a number of labs that will assess your understanding of the topics covered in class. These will be simple programming assignments that will serve to help you on your assignments.

◦ Form a group and come up with an idea for a full web application to use as your final project, which you will submit in the form of a final project proposal.

◦ Setup your database schema and submit a database proposal.

◦ Give a non technical presentation detailing your project idea.

◦ Give a technical presentation demonstrating your project result.

◦ Submit your final project code.

What are labs like?

There are ten labs, designed to make you practice the material that we will go over in class that week. The labs will give you a good foundation for your assignments and final projects.

Labs will begin at the end of lecture, and are designed to only take an hour; myself and the CAs will be available until the end of lecture. You will, however, be given 24 hours to complete the lab.

Labs are focused on small, approachable goals.

The majority of your labs will be incremental: they will build on the solutions from the previous week.

What is the final project like?

For your final project, you will create a market-ready, database-driven application as part of a team. Your application will incorporate aspects from the entirety of the course.

The final project has several components to it, both technical and non-technical.

- You will form a group and submit a project proposal
- You will submit a proposed format for your database
- You will give a non-technical presentation detailing what your product is, who the user would be, and why the project is worthwhile to take on.
- You will give a technical presentation detailing your features and the technical decisions you made
- You will submit your codebase and a database dump and deliver the actual product.

General Policies and Notes

You will have ample time to complete each assignment, so lateness will be harshly penalized:

◦ Late labs will receive a 10 point penalty per day

◦ There is no lateness tolerated on final project components; if you miss a deadline, you will receive a 0 for that component.

◦ **If some extenuating circumstance occurs that will cause you to be late, and you know in advance that it will cause you to be late, reach out to me as soon as possible.**

In lieu of a textbook, assignments will require you to research the topics in order to complete them. I will point you to resources for each assignment.

If you are having a group issue during the final project period, you must reach out to me **as soon as possible**; it is much easier for everyone to resolve issues early and amicably than let them destroy a group as the deadline approaches.

Readings

Most weeks, I will give recommended readings to read up on content that will be covered in the following week.

It is **highly beneficial** to read those readings before class as a form of preparation.

For many labs and parts of your final project, you will be expected to read some form of documentation in order to learn how to use a particular technology or package.

Course Format (Online & In Person)

This course runs as both an in-person and online course, and both courses are graded the same way. Online students and in-person students may work together on their final projects.

Each class follows an identical timeline:
- In person-lectures are held on Friday
- Labs are due the following Thursday at 11:55pm EST, unless otherwise stated

Online students are expected to read the lecture slides; optional, bi-weekly office hours will be held at varying times to accommodate questions in a real-time manner.

For both courses, the combination of lecture slides + weekly code is your primary source of material for this course.

What is web development?

What is web development?

Simply put, web development is the the very broad field of creating and building websites and web applications.

There are many opinions out there on what the best technology is and what the best practices are: this course is not about opinions and proclaiming what the best technology, but rather arming you with the skillset needed to work in any web technology.

What's special about web development?

There are many aspects of web development that are not different than non-web development
- You will break complex problems down to smaller, approachable issues
- There are a slew of programming languages and technologies to choose from

There are some unique problems, as a web developer, that you have to care greatly about
- In some way or another, your product is about transmitting information; you have to worry about the actual delivery of your information.
- Technology on the web moves **fast!** Change is constant!

Why is web development important?

Web development has allowed the internet to bring forth a new era for technology
  ◦ Lower barrier of entry; you can start without a compiler, all you need is a text editor and a browser!

Web development has ushered in a new era of communication, where ideas and content can be transmitted in new and exciting ways.

Allows information to be spread much more rapidly through many mediums.

Technologies and Tools

Git: For Version Control

If you are unfamiliar with Git, it is a versioning control software.

Versioning control allows us to take periodic snapshots at code and save a reference to it at a certain point in time.

You can download Git here:
- https://git-scm.com/downloads

Node.js (Server-Side JavaScript)

Node.js is a JavaScript runtime environment that allows you to write JavaScript without a browser. It also exposes a number of system operations that allow you to manipulate files, make servers, etc.

Node.js has a huge community and large package repository, making it easy to build applications without having to re-engineer the wheel.

You can download Node.js here:

◦ https://nodejs.org/en/

◦ Make sure to download version **8.9.4**

◦ You may also opt to use nvm (https://github.com/creationix/nvm) to install that version

MongoDB

MongoDB is a document based database.

You can download MongoDB here:
- https://www.mongodb.org/downloads#production

tota11y

The *tota11y* tool is an accessibility testing tool created by Khan Academy for the sake of identifying accessibility issues.

You can install it via a bookmarklet from the tota11y website
- http://khan.github.io/tota11y/

You may wonder why it's called *tota11y*; the phrase *a11y* is a condensed version of the term *accessibility*; there are 11 letters between the *a* and *y* in that word.

Being able to call yourself an accessibility *ally* also makes it quite a good phrase.

HTML and CSS

HTML and CSS are the markup and styling languages of the web, respectively.

HTML describes the format of a document, while CSS is a set of specifications as to how a document is styled.

You will write HTML and CSS to make web pages and applications.

JavaScript (Browser)

JavaScript originated as a programming language that was only run in a web browser. You will not only be writing JavaScript to run on Node.js, but you will also be writing it to run in your web browser.

In a browser environment, you will not have access to a user's file system; instead, you will have access to a number of browser APIs, such as a limited use of a user's history, their screen size, and so on. This will allow you to create robust web applications.

Preparing for CS-546

Install Node.js, npm, and MongoDB

You will need Node.js now
- ◦ While installing, it may ask you if you want npm; **you do**.

You will not need MongoDB for a few weeks, but it does well to install it and test it now.

Check out some Text Editors and IDEs.

There are many options for text editors to program for this course. Some common text editors people like for frontend programming are:

◦ Sublime Text

◦ Notepad++

IDEs

◦ Webstorm

◦ Visual Studio Code

My personal recommendation is **Visual Studio Code**, as it allows us to use a debugger in Node very easily.

Download a developer-friendly browser

While you can use any browser you want on a day-to-day basis, you're going to want to develop a browser that has a great developer tool panel:

- Mozilla Firefox
  - https://www.mozilla.org/en-US/firefox/new/
  - http://getfirebug.com/
- Google Chrome
  - http://www.google.com/chrome/

Read up on the Fundamental JavaScript

Mozilla, the maintainers of the Firefox browser, are excellent resources for all things frontend; for now, you can start by looking at JavaScript basics.

- https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/JavaScript_basics