

CS 546 – Web Programming I

Course Introduction and Policies





STEVENS
INSTITUTE *of* TECHNOLOGY

**Schaefer School of
Engineering & Science**

stevens.edu

Patrick Hill
Adjunct Professor
Computer Science Department
Patrick.Hill@stevens.edu



About Me

Education:

- Associate in Applied Science in Computer Programming and Systems from LaGuardia Community College.
- Bachelor of Business Administration w/ concentration in Computer Information Systems (minor in Psychology) from Baruch College.
- Master of Science in Computer Science from Stevens Institute of Technology.

Professional:

- Professional programmer since 1998 (worked at various companies throughout the years, from small/mid-sized startups to large law firms and corporations).
- Current CTO of Startup.

Teaching:

- 2017-2018: Started as a TA in CS 546, CS 223, and CS 810.
- 2017-2018: Stevens Pre-College Program. Taught Intro to Computer Science and was a TA for the Cybersecurity program.
- Fall 2018-Present: Became an Adjunct Professor at Stevens teaching CS 546, CS 554 and more recently, CS 146.

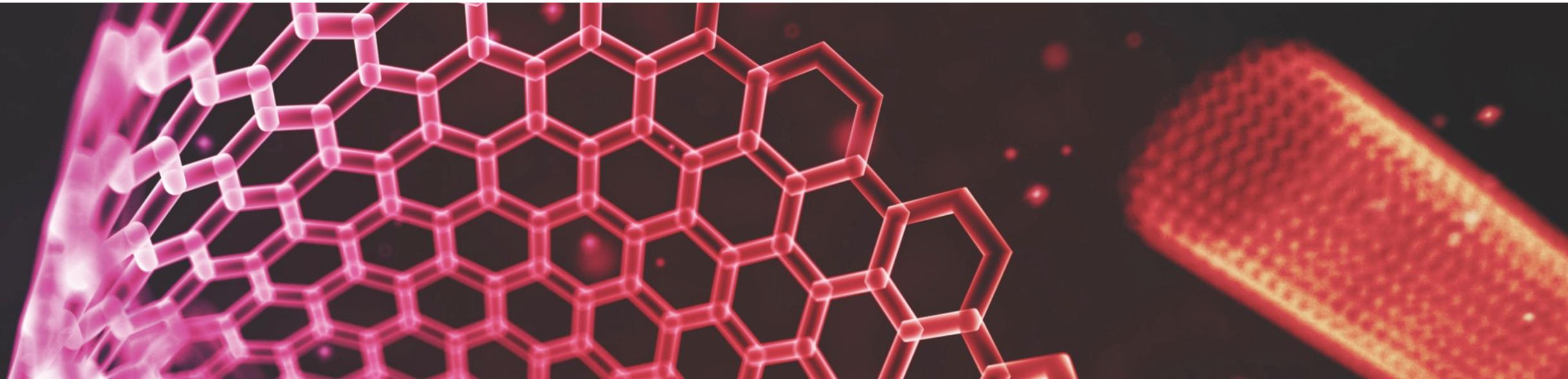


STEVENS
INSTITUTE *of* TECHNOLOGY

Schaefer School of
Engineering & Science



Course Logistics and Policies





Course Communication - Slack

- We will be using Slack for most communication in the course. You will find an invitation link in the course modules on canvas.
- Every student should be in the slack workspace in the #cs-546 channel for course related questions. #general is used for general chat related to CS, #random is for any off-topic chat.
Do not ask any course related questions in any other channel other than #cs-546
- Please do not direct message TA's or myself unless invited to, keep all communication in the chat channel.
- Even if you do not chat much in the channel, it is advised that you read the channel frequently for clarifications on assignments that other students may have asked.
- You can join the slack by visiting this URL. (it is also posted on canvas) <https://tinyurl.com/uosh623>



Course Codebase

The lecture code as well as all lecture slides for the course can be found on GitHub:

- <https://github.com/stevens-cs546-cs554/CS-546>



Grade Breakdown

Labs: 30%

- Labs will be weighted evenly.
- Labs will be given most weeks and will cover content covered that week.

In-Class Exercises/Quizzes: 5%

- Some weeks we will have an in-class exercise or quiz.
- **YOU MUST BE PRESENT IN CLASS TO GET ACCESS TO AND GET CREDIT FOR THE EXERCISE/QUIZ (NO EXCEPTIONS)**

Final Project Proposal: 5%

- Students will be assigned groups randomly.
- Groups will Propose a final project to work on throughout the semester.

Final Project Database Proposal: 5%

- Each group will submit a database proposal with their collections and schema.



Grade Breakdown

Final Project Pitch Presentation: 10%

- Each group will give a non-technical in-class presentation about their project, explaining what use case it solves, who the audience is, and why it's a worthwhile project.
- **Attendance is MANDATORY for all students on days that there are presentations.**

Final Project Walkthrough Presentation: 5%

- You will give a technical presentation showcasing your project's features and technological decisions.

Final Project Result and Code: 40%

- Each group will submit their project code and a database seed file to be reviewed and graded based on what was promised in their proposal and delivered.
- The grade for this will be split. 20% will be graded based on the project as a whole and the other 20% will be based on an individual's contribution to the project.



Attendance

- While I do not require attendance except on Project Pitch presentation days, you are **HIGHLY** advised to attend class.
- Even though I do not require attendance, you are responsible for all information that is communicated in the lectures.
- I may clarify or communicate important information about the coursework during the lectures. If you miss a class, you will miss that information, but you will still be responsible for it.
- You will need to be present on days we have in-class exercises or quizzes to get credit for them. No make ups will be granted for missed in-class assignments.



Course Responsibility & Assignment Extensions

- I do not grant assignment extensions.
- When you register for a class you are taking responsibility to follow the course schedule, complete the work, and complete it by the deadlines set.
- We live in the age of smartphones and free WIFI on every corner, so “But I had no internet” excuses are not valid and will not be accepted.
- “I will be traveling” is also not a valid excuse for late assignments or missing presentation days as you should not be traveling when you have a responsibility to your classes, a responsibility you took on when you registered for the course.



Late Assignments

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

- **Late labs will receive a 15-point penalty PER DAY. A lab is considered late 5 minutes after the due date/time (You have a 5-minute grace period).**
- **For final project components, the penalty is 25-points per day and a final project component is considered late 5 minutes after the due date/time (You have a 5-minute grace period).**

There are NO exceptions to these policies unless a verifiable doctor's note is provided, or you are officially excused from the office of Graduate Affairs.



Plagiarism and Cheating

- **I have a ZERO tolerance policy when it comes to sharing code with each other and cheating. Anyone caught cheating will receive an automatic 0 and be reported to the college.**
- Moss is run against every student's lab submissions to detect plagiarism in code and it is VERY VERY good at what it does.
- Moss is not only run against all current student submissions but also against ALL student submissions from previous semesters.
- If Moss detects a 40% or higher code similarity between your work and another student's or previous student's, you will receive an automatic 0 for that lab and the incident will be reported to the Honor Board and the Graduate Academics department without hesitation.
- You CANNOT use any code from a previous student found online on GitHub, any other online code repository etc..

Again, ZERO TOLERANCE.



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.
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.



What Will We Be Covering in This Course?

6. You will learn how to use JavaScript to make your applications respond to your users' input and experience.
7. 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.
8. You will learn advanced client-side programming techniques and how to leverage frontend tools that allow you to create incredibly dynamic web experiences.
9. You will learn about security issues in the web and how to minimize their risks.
10. 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 several 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.
- You will be placed into groups and your group will need to produce 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 the 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 are focused on small, approachable goals.
- Most of your labs will be incremental: they will build on the solutions from the previous week.



How Are the Labs Graded?

- The TA's will run various test cases against your code, if it fails any of those tests, points will be deducted. You will be responsible to make sure you cover any edge cases and make your code as bug-free as possible.
- It is VERY important that you follow EVERY detail to a T in the lab spec requirements.
- A programmer's job is not just coding but also following spec sheets and paying CLOSE attention to the details. **Any deviation** from what the assignment states will result in a penalty no matter how small of a deviation.
- Your lab grades are **NON-NEGOTIABLE**. The only time lab grades will change after they have been graded is if there is a grading error.



Advice on How to Do Well on Labs

- Start them early! Do not wait until last minute to do them.
- Follow every detail, failure to do so will result in points being lost, even if it's a small detail that you do not think matters much.
- If you have ANY doubts about ANY of the assignment requirements, reach out to either a TA or myself for clarification.
- Double check your work before you submit it. Once you think you are done with an assignment print out the assignment specs and carefully go over it word for word to make sure you covered everything.
- Double check you submitted the correct file and a valid file (you can do this by downloading the submission and testing it). **If you upload an incorrect file, i.e. upload lab 1 when you should have uploaded lab 2 or upload a corrupt file, it will result in a 0 for the lab.**
- **Test, debug, test, debug and test some more! Step through your code using breakpoints when debugging!!!!**



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 be randomly placed in groups and submit a project proposal.
- You will submit a proposed format for your database.
- You will be required to use GitHub or a similar online repository for the project so contributions can be monitored.
- You will give an in-class 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 (this will be a screen cast where one group member goes over the various functions and features of the system and will be submitted with your final code).
- You will submit your codebase and a database seed file and deliver the actual product.



How Is the Final Project Graded?

You will be graded based on many factors for the final project.

- You will be graded on the functionality of your core features and will also lose points for any core feature not implemented or not fully working.
- You will be deducted points for any bugs, issues with routing, logging in or anything that makes the application not function as intended as well as not fulfilling any of the final project requirements.
- You will be deducted points if you do not contribute as much as your teammates.
- You will get extra points for any extra feature you complete that is fully functional.
- We will also have “checkpoints” throughout the semester where I will ask your group for a status report on the work that has been completed to date.



Advice on How to Do Well on the Final Project

- The final project and all its components are like your final exam and should be treated as such.
- Start Early! Most issues with final projects come from those groups who started working too late.
- Make sure you meet all your core requirements and they function as intended and noted in the project proposal.
- Make sure you fulfill all requirements set for the project.
- Double check you submitted the correct file and a valid file (you can do this by downloading the submission). **If you upload an incorrect file or upload a corrupt file, it will result in a 0 for the final project code portion.**
- **Test, debug, test, debug and test some more! Step through your code using breakpoints when debugging!!!!**
- 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.



How to Succeed in This Course

If your grade is important to you, you will follow these instructions.

- Attend the lectures, read the slides and read the recommended readings.
- Start your labs early and do not wait until last minute. If you wait until last minute, you cannot assume there will be a TA available to help if you get stuck.
- Start the group project early and do not wait until last minute
- Make sure you meet ALL your core features listed in your final project proposal.
- Follow every detail on the lab assignment specs fully. Once you think you are done with a lab, go over the requirements twice and make sure you have met them all.
- Test, debug, test, debug and test again. Step through your code using breakpoints when debugging!!!!
- Attend the TA's office hours.
- ASK QUESTIONS!



Teaching Assistants

- We have 6 teaching assistants for this course. They each hold a 2-hour office hours every week to answer your questions and they will also be in the Slack channel to answer any of your questions
- They will be making their introductions and posting their office hours soon.
- I am very protective over my TA's therefore rudeness, pushiness, hostility etc. will not be tolerated. If a TA tells me a student was rude, pushy or demeaning in any way then we are going to have issues.
- The TA's have no authorization to change your lab grades without discussing it with me first. They also do not set the amount of points you get deducted for an issue.
- If you have a question, please reach out to the TA's first, if they are unable to address your issue then reach out to me.



Readings

- 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.
- Most weeks, I will provide recommended readings and other resources regarding the content that will be covered.
- 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.

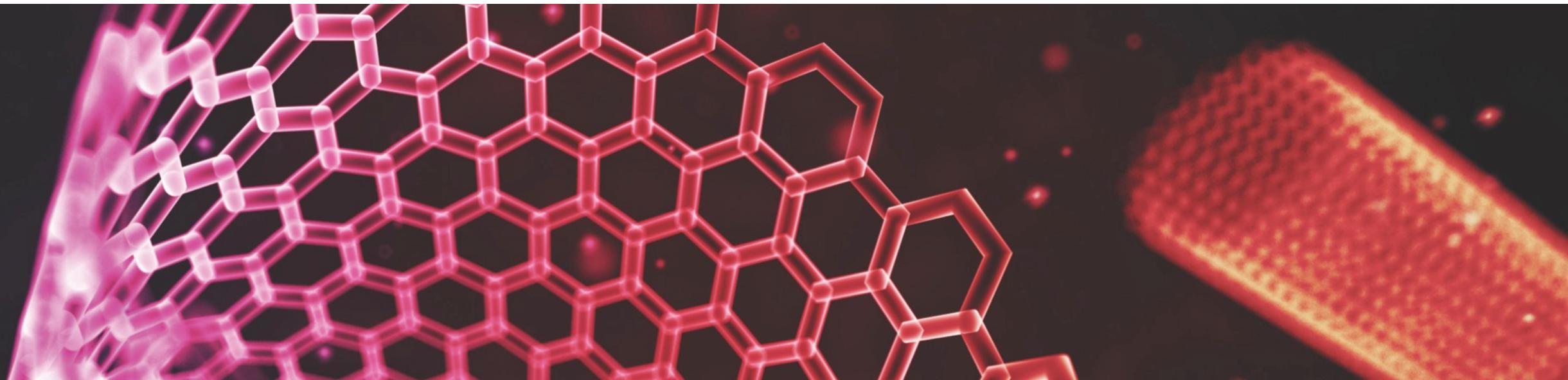


STEVENS
INSTITUTE *of* TECHNOLOGY

Schaefer School of
Engineering & Science



What is Web Development?





What Is Web Development?

Simply put, web development is 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 Is 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 must care greatly about

- In some way or another, your product is about transmitting information; you must worry about the actual delivery of that information.
- Technology on the web moves fast! Change is constant!



Why Is Web Development so 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.

Web Development allows information to be spread much more rapidly through many mediums.

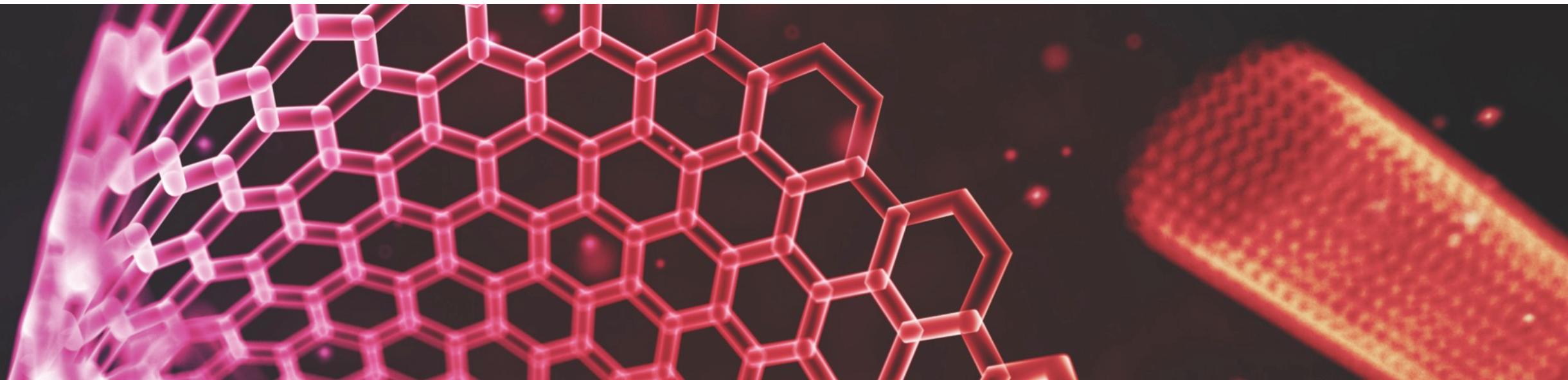


STEVENS
INSTITUTE *of TECHNOLOGY*

Schaefer School of
Engineering & Science



Technologies and Tools





Git for Version Control

If you are unfamiliar with Git, it is a versioning control software. We will cover it in lecture 8.

- Versioning control allows us to take periodic snapshots at code and save a reference to it at a certain point in time.
- Allows many developers to work on the same files and push their changes to an online repository.

You can download Git here:

<https://git-scm.com/downloads>

You will be required to use Git for the final project so that contributions can be monitored



Node.js: Server-Side JavaScript

Node.js is a JavaScript runtime environment that allows you to write JavaScript without a browser. It also exposes several 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 the most current LTS version



MongoDB

MongoDB is a document-based database.

You can download MongoDB here: <https://www.mongodb.com/download-center/community>

We will not need this until lecture 4 but you are advised to set it up ASAP as we will have an in-class exercises during lecture 4 and you will need to have Mongo installed and operational by then.



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 web applications.



Client-Side JavaScript

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 on the client-side.

In a browser environment, you will not have access to a user's file system; instead, you will have access to several 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.

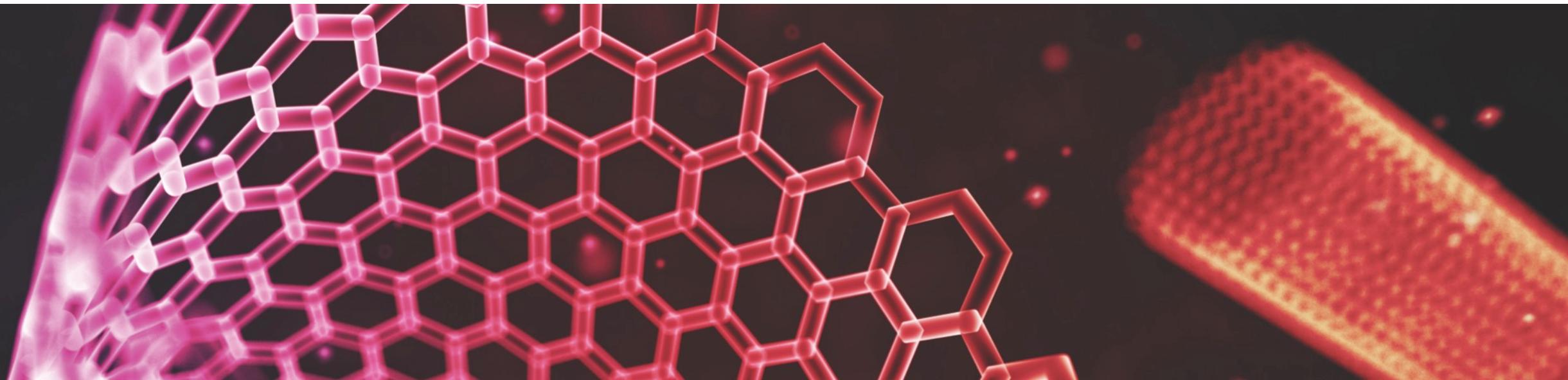


STEVENS
INSTITUTE *of* TECHNOLOGY

Schaefer School of
Engineering & Science



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.



IDE

The IDE we will be using for this class is Visual Studio Code.

- <https://code.visualstudio.com>

It runs on Windows, Mac and Linux and is a very lightweight yet powerful IDE that supports a ton of languages and technologies and has many installable extensions to make development easier.

I will show you some neat things about using VSCode for node development. You're free to use another IDE but I highly advise using VS code. It has great debugging features, git version control, integrated terminal and many other nice features.

You can also download an alternate version of VS code that was built from their source code but with their telemetry/tracking removed. It's called VSCode and can be found here: <https://vscodium.com>

It has all of the same features as the normal VSCode.



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 using a browser that has a great developer tool panel:

Mozilla Firefox Developer Edition

<https://www.mozilla.org/en-US/firefox/developer/>

Google Chrome

<http://www.google.com/chrome/>

It's always a good idea to have multiple browsers installed on your development machine so that you can test your application in different browsers to find any browser compatibility issues.



Read up on the Fundamentals of JavaScript

Mozilla, the maintainers of the Firefox browser, are excellent resources for all thing's 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](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/JavaScript_basics)

W3 Schools is also a great resource for all thing's web development related:

<https://www.w3schools.com>

And here is their JavaScript page. <https://www.w3schools.com/js/default.asp>



STEVENS
INSTITUTE *of* TECHNOLOGY

Schaefer School of
Engineering & Science



Questions?

