



# THE CODING BOOT CAMP AT UNC-CHAPEL HILL

---

## OVERVIEW:

### Full Stack Flex Program

The digital revolution has transformed virtually every area of human activity—and you can be part of it as web development professional. [The Coding Boot Camp at UNC-Chapel Hill](#) is a part-time, 24-week Full Stack Flex course that gives you the knowledge and skills to build dynamic end-to-end web applications and become a full stack web developer.

Courses are scheduled to fit into your life, whether you're employed or attending college full-time, with convenient weekend and evening sessions.

The program is rigorous and fast-paced and covers both the theory and application of web development. As you gain proficiency, you'll use what you learn on real projects under the guidance of area employers. Plus, you'll have an impressive Professional Portfolio and the confidence to succeed as a web development professional.

# IS THE PROGRAM RIGHT FOR YOU?

Are you creative, curious and looking to reinvent yourself professionally? If so—or if any of the following describes your situation—enrolling in our coding boot camp could be a smart career move:

- You're considering a career change but not sure how to take the first step.
- You're happy in your current field, but want to move to another company—or stay put but shift from a non-technical into a technical position.
- You want to engage more deeply with your current job—or boost your earnings and broaden your experience with freelance work.
- You have an entrepreneurial idea and need to acquire the skills to go “all in” on it and launch your business.
- You're a full-time student but hungry to learn more and expand your skill set.

## THE SKILLS YOU'LL GAIN

You will graduate with full stack web development skills\*, including:

### Computer Science applied to JavaScript

- Algorithms (Searches, Sorts)
- Efficiency
- Time Complexity
- Big O Notation
- Data Structures

### Browser Based Technologies

- HTML
- CSS
- JavaScript
- jQuery
- Responsive Design
- Bootstrap
- Handlebars
- Local Storage, Session Storage, IndexedDB
- React.js

### API Interaction

- API
- JSON
- AJAX

### Deployment/Command-Line Fundamentals

- Heroku
- Git
- GitHub Pages

### C#

- ASP.NET

### Databases

- MySQL
- MongoDB

### Server Side Development

- Node.js
- Express
- User Authentication
- Progressive Web Applications (PWAs)
- MERN Stack (MongoDB, Express.js, React.js, Node.js)

### Quality Assurance

- Unit Testing
- Functional Testing
- Linting
- Continuous Integration

\*The material covered is subject to change. Our academic team adjusts to the market demand.

# BUILDING ON THE BASICS

In web development, you can't succeed without a solid grounding in the fundamentals. That's why our curriculum begins with a deep dive into the basics of coding and data structure.

That said, we recognize that the surest way to impress prospective employers and get job offers is to demonstrate your skills on real-world projects. You'll have ample opportunity for hands-on involvement in outside projects, which will make up your Professional Portfolio.

---

## REAL PROJECTS, REAL JOBS

Our graduates will be qualified for many different roles, including:

- Full Stack Developer
- Frontend Web Developer
- Backend Web Developer
- Product Manager
- Technical Project Manager
- QA and Test Engineer
- Technical Business Analyst
- Software Developer
- Application Development Manager
- Computer Programmer
- Web Designer
- Email Developer
- Web Producer

---

## WHAT YOU WILL LEARN

By the time you graduate, you can expect to be able to:

- Apply “social coding” accepted and best practices (including source control, issue tracking, functional feedback, etc.)
- Build a frontend website either from scratch or by utilizing a frontend framework (such as Bootstrap)
- Deploy static and dynamic websites to the cloud
- Implement complex logical conditions to meet an objective
- Write SQL commands to perform Create, Read, Update and Delete commands
- Create a full stack Single Page Application with AJAX communication
- Develop your vision for a website—and then build it!
- Expertly navigate the file system and terminal basics
- Work independently or in a group on complex projects throughout the entire development lifecycle
- Understand the basics of troubleshooting and enhancing legacy code
- Communicate the basics of serving a web page and how the browser renders code
- Create RESTful APIs utilizing JSON as a data format
- Consume RESTful APIs properly utilizing REST verbs
- Create web applications and services in C# using ASP.NET
- Create session-based applications utilizing user authentication schemes that are well-known and widely used

# COURSE STRUCTURE

Over the course of 24 weeks, you'll attend informative lectures and take part in a variety of individual and team exercises, working independently and in groups, in the classroom and at home. Homework assignments provide an opportunity to apply what you've learned and build on it. The goal is to give you a comprehensive learning experience and true insight into a "day in the life" of a full stack developer.

## Discussion

Instructor-led discussions cover the background, history and use of a new technology or concept.

## Lab Work

You'll put classroom teaching into practice individually and with a team to work on timed in-class exercises and projects.

## Portfolio Projects

Your portfolio signals to employers that you are ready for primetime! You'll build a substantial portfolio of projects that demonstrate your abilities across a wide variety of technologies.

---

## WE'RE HERE TO HELP

As you move up the learning curve, you're likely to have questions around some of the concepts covered in class. We're here to help—through in-person and virtual office hours, as well as a dedicated #slack channel where you can get assistance from instructors, support staff and your fellow students. All work is done via Github, so you can create issues directly on your own projects for instructors to assist you in a truly asynchronous fashion. In addition to learning to code, you will have access to career services that will help you prepare for technical roles after graduation such as:

- Career Content and Practice Sessions
- Database of Customizable Tools and Templates
  - Multiple Technical Resume Templates
  - Github Best Practices
  - Guidelines To Building A Portfolio
  - Creating an Elevator Pitch
  - Developing a Bio
- Online Career Events With Industry Professionals
- Soft Skills Training
- One-on-One Career Coaching



# PORTFOLIO

## BUILDING YOUR PORTFOLIO

It's a fact: Companies care about what you can **do**, not what you **say** you can do.

For that reason, our curriculum teaches you how to put what you've learned to work on actual portfolio projects, ranging from simple HTML and CSS code samples to sophisticated Single Page Applications with backend databases.

---

## YOUR FULL STACK PORTFOLIO PAGE

Once you complete our program, your portfolio page will help you showcase your work with links and descriptions to the projects you've created, code samples, and personal information that employers want to see. Think of your portfolio page as your new home on the web.

### Skills Needed:

- HTML5
- CSS
- JavaScript
- Bootstrap
- Git
- Heroku

### Objectives

- Create a home on the web to showcase your skills
- Build a complete site from concept
- Commit code to a shared repository

---

## BUSINESS-ORIENTED HOMEWORK PROJECTS

Our homework assignments are designed to emulate two real-world scenarios: 1) on-the-job tickets; and 2) job-seeking coding challenges. In both cases, the assignment is framed as a user story. In addition to user stories, ticket-based homework assignments follow the Agile project management conventions of framing the issue in terms of business context and acceptance criteria.

### Skills Needed:

#### All homework:

- HTML
- CSS
- JavaScript
- Git

#### Select homework:

- jQuery
- Bootstrap
- MySQL
- Node.js
- Express.js
- ORM
- API Consumption
- Heroku
- NoSQL
- React
- HTML5/CSS
- JavaScript/jQuery
- State Management

### Objectives

- Each assignment focuses on a specific layer of the tech stack; objectives will vary based on the tech stack focus.

## SELF-SELECTED FRONTEND PROJECT

This is a group project that forces you to think outside your comfort zone. You and your group will decide what to build and then build it—a frontend application that interacts with real-world services like Google Maps, Twitter or the OMDb API.

### Skills Needed:

- HTML5/CSS
- JavaScript/jQuery
- API Consumption
- JSON
- Cloud Database Usage

### Objectives

- Work in a group to build a project together
- Interact with third-party services
- Think in terms of mobile responsive design
- Read/write from/to a remote database

---

## FULL STACK PROJECT

In your first full stack web application you'll create an intuitive frontend/robust backend and scalable database.

### Skills Needed:

- HTML5/CSS
- JavaScript/jQuery
- State Management
- Sessions
- Bootstrap
- Interactivity (AJAX)
- MySQL
- Node.js
- Express
- ORM

### Objectives

- Track issue progress with industry standard tools
- Communicate with team members asynchronously
- Design a MySQL Database Schema
- Create a full stack application
- Write project documentation
- Understand database relationships

---

## ASP.NET APPLICATION

Exposure with Microsoft's versatile .NET ecosystem is highly sought after by employers. Familiarity with web development using C# and ASP.NET is sure to earn their attention.

### Skills Needed:

- HTML
- CSS
- SQL & NoSQL Databases
- Git
- C#
- ASP.NET

### Objectives

- Create an ASP.NET Project
- Use databases to build dynamic web applications
- Build cloud-based web apps and services
- Implement robust and flexible APIs

---

## FINAL PROJECT

You will work independently or break out into groups to collaborate on a final project. You will come up with your own project and actually build it. The skills you learn during this project will truly help you to prepare for your first interviews and jobs!

### Skills Needed:

- Everything you've learned!

### Objectives

- Define project scope
- Quality Assurance testing
- Responsive Design
- Deployment
- Code Organization

# COURSE CURRICULUM BY PHASE

Phase	Description	What You'll Learn
<b>Phase 1: Foundation</b> (Weeks 1-8)	The first phase, Foundation, equips you with the fundamental concepts of web development, covering HTML, CSS, and JavaScript, as well as command line fundamentals and API consumption.	<ul style="list-style-type: none"><li>• HTML, CSS, and JavaScript</li><li>• Creating a web page from scratch</li><li>• Mastering terminal commands</li><li>• DOM manipulation</li><li>• jQuery</li><li>• Consuming RESTful APIs</li><li>• Parsing JSON to extract meaningful data</li><li>• Using AJAX to update data on a website</li></ul>
<b>Phase 2: Technical</b> (Weeks 9-16)	In the second phase, Technical, you learn the skills necessary to engineer a full stack web application, working with servers, databases, and other back end technologies, and connecting them to the front end.	<ul style="list-style-type: none"><li>• Writing Node.js server code to serve static web pages</li><li>• Querying large amounts of data and answering questions from a MySQL database</li><li>• Understanding and using Joins, Wheres, and Counts strategically</li></ul>
<b>Phase 3: Performance</b> (Weeks 17-24)	The last phase, Performance, has a dual meaning in that you acquire skills to optimize your web applications for speed and efficiency as well as prepare yourself for the transition to a career in web development.	<ul style="list-style-type: none"><li>• Utilizing NoSQL databases, such as MongoDB, as an alternative to MySQL</li><li>• Improving the performance of applications</li><li>• Converting traditional applications into progressive web applications (PWAs)</li><li>• Creating single-page applications with React</li><li>• Computer Science applied to JavaScript (data structures, algorithms)</li></ul>
<b>Asynchronous Regional Content</b> (Week 25)	ASP.NET is a flexible C# web framework, designed to maximize programmer productivity. ASP.NET allows developers to build safe, scalable, and robust applications, quickly.	<ul style="list-style-type: none"><li>• Create web apps and APIs</li><li>• Take a deep dive into C# and ASP.NET</li><li>• Build safe applications for large numbers of users</li></ul>