

# TRILOGY'S SKILLS BOOTCAMP IN FRONT-END WEB DEVELOPMENT

## CURRICULUM OVERVIEW

The digital revolution has transformed virtually every area of human activity, and learning to code can put you, too, at the heart of today's dynamic workforce.

The [Skills Bootcamp in Front-End Web Development](#) is an intensive programme that provides students with the hard and soft skills needed to pursue a career in front-end web development. It offers students a crash course in application programming interfaces (APIs), user experience design, and building and deploying modern web applications. Students learn in-demand skills through a market-driven curriculum that includes HTML5, CSS3, Javascript, jQuery, React.js, Node, and more.

Through a series of hands-on activities and authentic projects students also gain experience with industry-standard programming practices — including agile development, version control, and technical project management — and use popular technologies such as Git, Github, and Netlify. The course is recommended for those seeking placement or advancement within technical roles, such as junior web developer, front-end designer, or a technical project manager position.

# Is a Skills Bootcamp right for **you**?

Success in this Skills Bootcamp takes creativity and curiosity. We seek students who are looking to push themselves to the next stage in their careers, whether they are starting fresh or looking to move up at their current role.

You should consider a Skills Bootcamp if:

You want to switch careers and find new challenges and opportunities in an in-demand industry

You would like to move into a more technical position but need to bridge the gap between your current role and the next step up

You need more flexibility in your career, or you want to work remotely, make your own hours, and strike out on your own as a freelancer

You hope to launch your own business and need new skills to take the leap

You recently graduated and need hands-on experience to enter an in-demand field

# What you'll learn

This is a fully online programme that pairs an immersive learning experience with a flexible, part-time schedule. Over the course of 16 weeks, you will progress through a blend of online challenges and live sessions led by instructors in the field who will teach you real-world skills.

Course topics include:\*

## Front-end fundamentals

- HTML
- JavaScript
- CSS
- UX
- Git
- Github
- Information architecture
- User experience design

## Modern front-end frameworks

- Structuring, positioning, and optimisation of web page content
- React.js
- Netlify

## Developing with APIs

- Javascript framework
- DOM manipulation
- Web-based applications
- Data extraction and retrieval

\* Note: These topics are subject to change based on local market demand and the input of hiring partners.

# Your **next** career move

Upon completion of the Skills Bootcamp you will be qualified to pursue a range of professional roles, such as:

Front-end web developer

Application developer

Technical project manager

Web developer

QA and test engineer



# Learning outcomes

Graduates of the Skills Bootcamp in Front-End Web Development are prepared to:

Apply fundamental knowledge of web development with HTML, CSS, and JavaScript

Organise code, having mastered foundations of object-oriented and functional programming paradigms

Create visually-compelling page layouts using UX design principles

Build modern web applications with industry-standard frameworks like node and React

Maintain code versioning and work across distributed teams using Git and GitHub

Deploy web-based applications using Netlify

Manipulate page information and create interactivity using jQuery and React.js

Prepare a portfolio of front-end development work

Retrieve data from servers using RESTful-compliant network requests



## Course structure

The Skills Bootcamp includes informative discussion as well as individual and team exercises. Homework assignments provide an opportunity to apply what you've learned, and to build on it. The goal is to give you a comprehensive learning experience and true insight into what it's like to be a front-end web developer.

### Group activities



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

### Practice labs



Work on timed in-class exercises and projects, and put classroom teaching into practice on your own and with a team.

### Project-based challenges



Build a substantial portfolio of projects that demonstrate your abilities across a wide variety of technologies.

# Support every step of the way

As you progress through your Skills Bootcamp, you will have access to a range of support and resources to help you succeed. You can work with instructors during office hours, access 1:1 tutoring and 24/7 learning assistants, and work with your classmates and support teams in a dedicated Slack channel.

You will also work closely with our dedicated Career Engagement Network, which offers the following comprehensive services to students:

CV and profile development coaching	Online career events with industry professionals
Interview practice sessions	Soft skills training
Customisable tools and templates	One-on-one career coaching

After completing the Skills Bootcamp, you'll have access to additional resources:

A Slack group for all Skills Bootcamp alumni	Curated open-source projects and activities
Technical and behavioral interview preparation materials	Lists of networking groups and nonprofits seeking web development volunteers

# Building **your** portfolio

You will build your portfolio throughout the programme, and will put in what you've learned as well as examples of your work on actual projects. Projects will range from simple HTML and CSS code samples, to sophisticated single page applications with back end databases.





# Building Your Portfolio

## Your portfolio page

Your portfolio page will showcase your work, with links and descriptions to the projects you've created, code samples, and personal information that employers want to see.

### Skills needed:

- HTML5
- CSS
- JavaScript
- Git

### Objectives:

- Create a home on the web to showcase your skills.
- Design and build a complete site from concept.

## Business-oriented homework projects

Challenge assessments are designed to emulate two real-world scenarios: on-the-job tickets and job-seeking coding challenges. In both cases, the assignment is framed as a user story.

### Skills needed:

#### All homework:

- HTML
- CSS
- JavaScript
- Git

#### Select homework:

- jQuery
- Bootstrap
- Node.js
- APIs
- React
- Unit Testing
- Deployment

### Objectives:

- Assess learner proficiency.
- Participate in a code review with instructional staff.

# Portfolio continued...

## Project 1: API application

Students write an API-powered web application that pulls data from multiple sources and adheres to design specifications.

### Skills needed:

- HTML5/CSS
- JavaScript
- jQuery
- API Consumption
- Bootstrap
- Git

### Objectives:

- Apply agile development methodologies.
- Prepare a professional presentation and repository README.
- Explain and execute Git branching workflow.
- Resolve merge conflicts.
- Speak technically about a feature you implemented.
- Explain agile or iterative software development.

## Project 2: Final project

Work independently, or break out into groups to collaborate on a final project, to build and deploy a client-side single-page application. The skills you learn during this project will truly help you to prepare for your first interviews.

### Skills needed:

- Everything you've learned!

### Objectives:

- Build a larger group project utilising contemporary front-end frameworks.
- Deploy projects on a host using tools like Netlify.
- Present final project to peers and instructors.

# Course curriculum by module

Module	Description	Learning objectives
<b>HTML, CSS, Git</b> (Week 1)	The basis of all front-end layouts, HTML and CSS together form the backbone of what gives web pages their content and styles. Students practice command line fundamentals using Git — the world's most popular version control system — and build a solid foundation in front-end development.	<ul style="list-style-type: none"><li>• Scaffold web pages using HTML.</li><li>• Create repositories and define versions using Git from the command line.</li><li>• Style HTML elements using CSS.</li></ul>
<b>Working with CSS</b> (Week 2)	We cover a broad set of topics in responsive layout-building provided by CSS technologies, including Flexbox, Grid, and media queries. We also cover wireframing and modern CSS syntax, such as selectors and variables. The week culminates in a challenge during which students build a responsive web portfolio site deployed using GitHub Pages.	<ul style="list-style-type: none"><li>• Design visually-appealing layouts using CSS grid and flex-box.</li><li>• Select elements programmatically using CSS pseudo-selectors.</li><li>• Implement responsive CSS styles which react to changes on page.</li></ul>
<b>UX + Advanced CSS</b> (Week 3)	Students learn to leverage frameworks and layouts so as to quickly and seamlessly structure and position content on a page. Contemporary front-end development requires an understanding of both technical skills as well as design fundamentals — including accessibility, information architecture, and user experience — in order to create cohesive interfaces and applications.	<ul style="list-style-type: none"><li>• Identify common UX design elements and how they affect user behavior.</li><li>• Analyse user funnels on existing websites.</li><li>• Build web pages using Bootstrap as a style framework.</li></ul>
<b>JavaScript fundamentals</b> (Week 4)	Foundational JavaScript programming principles allow students to design engaging and interactive user experiences. This module covers the basics of the language syntax, conditional statements, arrays, and looping, and equips students with programming fundamentals.	<ul style="list-style-type: none"><li>• Create simple applications using the JavaScript language.</li><li>• Identify common JS syntax, such as variable types, conditional statements, and Arrays.</li><li>• Implement for-loops to iterate through values or over arrays.</li><li>• Create applications that accept user input through prompt.</li></ul>

# Course curriculum by module

Module	Description	Learning objectives
<b>Intermediate JavaScript</b> (Week 5)	We extend from our JavaScript fundamentals to work with objects and functions in order to organise code effectively. The combination of the new data structure and flow control enables students to write more reusable, modular code.	<ul style="list-style-type: none"><li>• Organise data using Objects in JS.</li><li>• Iterate over information in an object using object methods.</li><li>• Maximise reusability of code with functions.</li><li>• Substitute repetitive elements in functions using JavaScript's "this."</li></ul>
<b>Working with web APIs</b> (Week 6)	Contemporary web development relies heavily on modern web browser features, including cookies, localdb, and DOM manipulation.	<ul style="list-style-type: none"><li>• Extract data from a web page, and place data into it.</li><li>• Trigger JS functions on events to add interactivity to a page.</li><li>• Store data client-side, using local storage.</li></ul>
<b>Third party APIs</b> (Week 7)	In this module, we review ways to manipulate the DOM with JavaScript, while exploring the jQuery library and other frameworks as alternative methods to basic JavaScript functions.	<ul style="list-style-type: none"><li>• Implement DOM manipulation strategies using the jQuery library.</li><li>• Build small games using JavaScript and jQuery.</li><li>• Create time-based operations using Moment.js.</li></ul>
<b>Server APIs</b> (Week 8)	Front-end systems rely on servers sending information to the browser. Learn how to send, receive, and manipulate data from external sources using modern web-based protocols.	<ul style="list-style-type: none"><li>• Access data on RESTful web APIs using AJAX calls.</li><li>• Process return data to isolate key elements for JavaScript logic.</li><li>• Generate page elements based on API responses.</li></ul>
<b>Project 1: API application</b> (Week 9 + 10)	Students begin work on their first major summative project, collaborating in groups to execute against specific design specifications. They write an API-powered web application that pulls data from multiple sources in a creative way.	<ul style="list-style-type: none"><li>• Work as part of a collaborative project team to apply agile development methodologies.</li><li>• Prepare a professional presentation and repository README for your project.</li><li>• Explain and execute Git branching workflow in a collaborative project.</li><li>• Resolve merge conflicts.</li><li>• Speak technically about a feature you implemented in your project.</li><li>• Explain agile or iterative software development as it relates to the project.</li></ul>

# Course curriculum by module

Module	Description	Learning objectives
<b>Working with ES6 and Node</b> (Week 11)	This module introduces advanced JS keywords and operators used in ES6. Students also learn Node.js, an open-source environment that executes JS outside of a browser. Students interact with Node.js in preparation for implementing React.js later in the course.	<ul style="list-style-type: none"><li>• Differentiate between JavaScript in the browser and Node.js environments.</li><li>• Implement arrow functions for anonymous callback functions.</li><li>• Utilise ES6 core language improvements, including spread and rest operators.</li></ul>
<b>Object-oriented programming</b> (Week 12)	Students deepen their understanding of JavaScript by learning the principles in object-oriented programming — specifically defining how inheritance works in JavaScript — before moving into React. This module also introduces the foundations of test-driven development (TDD).	<ul style="list-style-type: none"><li>• Identify and implement Prototypical inheritance in JavaScript.</li><li>• Create constructor functions to instantiate multiple objects of similar type.</li><li>• Ensure code functionality using test-driven development (TDD)</li></ul>
<b>Modern development with React</b> (Week 13)	React.js is the de facto leader of front-end libraries. Students learn to decompose complex web pages into smaller, interactive components that re-render with high frequency. They also recreate their portfolio site using React, to improve real-time functionality.	<ul style="list-style-type: none"><li>• Build web pages using the most popular library on the web.</li><li>• Decompose page layouts into JSX components.</li><li>• Reimplement fundamental web page organisation using componentised layouts.</li><li>• Pass data between components using state and props.</li></ul>
<b>React state and deployment</b> (Week 14)	Students deepen their understanding of contemporary React by exploring Hooks. Students review and employ test- and behavior-driven development methodologies that are critical to building front-end systems that run stably and handle user input safely.	<ul style="list-style-type: none"><li>• Manage component state using modern React Hooks.</li><li>• Test front-end UX using behavior-driven development (BDD).</li><li>• Deploy front-end applications using Netlify.</li></ul>
<b>Project 2: Final project</b> (Weeks 15 + 16)	In the final project, students use their knowledge of React, and all of the other technologies covered in this course, to build and deploy a client-side single-page application.	<ul style="list-style-type: none"><li>• Synthesise all of the skills covered in this course.</li><li>• Build a larger group project utilising contemporary front-end frameworks.</li><li>• Deploy projects on a host using tools like Netlify.</li><li>• Present final project.</li></ul>