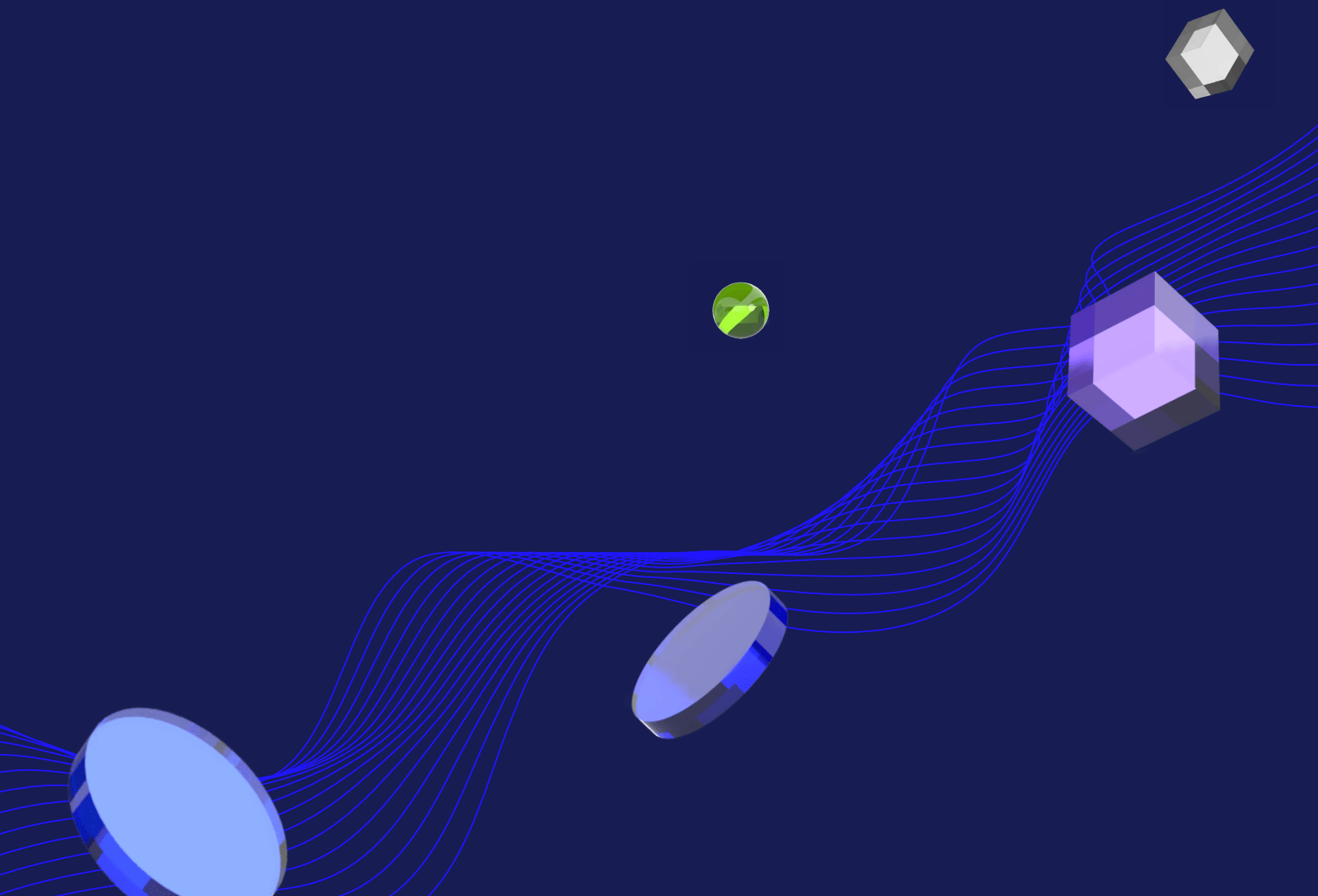




SCHOOL OF PROGRAMMING & DEVELOPMENT

Front End Web Developer

Nanodegree Program Syllabus



Overview

The goal of the Front End Web Developer Nanodegree program is to equip learners with the unique skills they need to build and develop a variety of websites and applications.



Learning Objectives

A graduate of this program will be able to:

- Construct responsive websites using CSS, Flexbox, and CSS Grid.
- Develop interactive websites and UI (User Interface) applications using JavaScript and HTML.
- Connect a web application to backend server data using JavaScript.
- Automate application build and deployment using Webpack.
- Improve offline performance of websites using Service Worker.

Program information



Estimated Time

4 months at 10hrs/week*



Skill Level

Intermediate



Prerequisites

Learners should be able to:

- Lay out a simple webpage using HTML.
- Style a website element using CSS.
- Write and test software with JavaScript.
- Inspect websites using Developer Tools on a modern web browser (Chrome, Firefox, or Edge).
- Debug and troubleshoot errors and failures in JavaScript programs.



Required Hardware/Software

Learners should have access to a computer with a broadband connection, on which you will install a professional code/text editor (e.g., Visual Studio Code, Atom, etc.).

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

CSS & Website Layout

For the first project, learners will create a multi-page blog website, using best practices for content and page styling with HTML and CSS. They'll practice using responsive layouts, Flexbox, and CSS Grid to create the structure and design for their own blog.



Course Project

Styled Blog Website

In this project learners will code a fixed-wing aircraft, and then implement solutions to a significantly more challenging control problem.

Lesson 1

Introduction to HTML

- Create a programming project with a code editor.
- Construct nested websites with HTML tags and elements.
- Troubleshoot and debug HTML errors and bugs.

Lesson 2

Introduction to CSS

- Style website components by ID, class, and type.
- Connect CSS to a website.
- Position and display website elements
- Modify and control website typography.
- Troubleshoot and debug issues with stylesheets.

Lesson 3

Introducing Flexbox

- Control web elements orientation and layout with Flexbox.
 - Control ordering of web elements with Flexbox.
 - Align and justify web elements with Flexbox.
 - Transform and resize web elements with Flexbox.
-

Lesson 4

CSS Grid

- Compare and contrast the use cases for CSS Grid and Flexbox.
 - Structure the layout of a web page using grid columns and rows.
-

Lesson 5

Creating Responsive Layouts

- Define custom styles for different screen sizes using media queries.
- Observe and create breakpoints in a website to change layout and styling as a page is resized.

JavaScript & the DOM

Use JavaScript to control a webpage. Learn what the Document Object Model (DOM) is, and use JavaScript and the DOM to dictate page content and interactions. Gain experience working with Browser Events and managing website performance by controlling content creation efficiently.



Course Project

Dynamic Landing Page for Marketing Content

In this project, learners will build a multi-section landing page. Often times, developers won't know how much content will be added to a page through a Content Management System (CSM) or an API. To handle this problem, learners will dynamically add content to a web page. They'll be building a landing page that combines their skills with JavaScript, HTML, and CSS to update and control the page and create a dynamic user experience.

Lesson 1

Syntax

- Declare block-scoped variables using `let` and `const`.
- Format JavaScript strings using template literals.
- Manage arrays and objects using JavaScript destructuring syntax.
- Iterate over arrays and objects using JavaScript `for` syntax.

Lesson 2

The Document Object Model

- Describe and explain the Document Object Model for web browsers.
- Access page elements by ID, class, and type using JavaScript.

Lesson 3

Creating Content with JavaScript

- Modify HTML content with JavaScript.
 - Create HTML content and elements with JavaScript.
 - Remove HTML content with JavaScript.
 - Style HTML content with JavaScript and CSS.
-

Lesson 4

Working with Browser Events

- Describe and explain the phases of browser events.
 - Create event listeners that handle browser events by writing code that runs when an event is triggered.
 - Describe and explain the events that are fired as a web page loads.
-

Lesson 5

Performance

- Manage website performance by controlling content creation efficiently.
- Describe what happens when a webpage has to be redrawn.
- Describe and explain the JavaScript call stack.
- Describe and explain the JavaScript event loop.
- Write efficient code by analyzing the call stack and event loop.
- Delay code execution with `setTimeout`.

Web APIs & Asynchronous Applications

Set up a Node and Express environment so you can develop web applications on your local machine, and learn to handle HTTP requests and routes. Update and modify website elements dynamically using asynchronously retrieved data.



Course Project

Weather Journal

In this project, learners will apply their new skills to combine data from the OpenWeatherMap API and client side (browser) HTML forms to create a web app that records a weather journal for users. This project requires learners to create an asynchronous web app that uses Web API and user data to dynamically update the UI for a Weather Journal App.

Lesson 1

Node & Express Environment

- Set up a Node and Express environment to develop a web application on your local machine.
- Install JavaScript packages using npm (Node Package Manager).
- Setup and run a local development server.
- Manage web application file structure and website assets using Express.

Lesson 2

HTTP Requests & Routes

- Handle requests to an Express with routes.
- Describe and explain the differences between GET and POST requests.
- Build a web server and use it to serve data and responses to web requests.

Lesson 3

Asynchronous JavaScript

- Manage asynchronous JavaScript control flow with Promises.
- Request data from a server using JavaScript Fetch.
- Update and modify website elements dynamically using asynchronously retrieved data.

Course 4

Build Tools, Webpack & Service Worker

Develop an understanding of how to use build tools, such as Webpack, for automating build tasks. Create CSS variables with Sass and configure Webpack to use Sass controlled stylesheets. Learn how to cache server data and website functionality using Service Worker.



Course Project

Article Analysis Website

In this project, learners will get a taste of some common production environments and tools that front end developers often come across. Learners will be building a web tool that allows users to run Natural Language Processing (NLP) on articles or blogs found on other websites. Using an exciting new API called Aylien, learners can build a simple web interface to interact with their NLP system. This tool will give back pertinent information about the article: whether the content is subjective (opinion) or objective (fact-based) and whether it is positive, neutral, or negative in tone.

For this project, learners will use:

- Node
- Express
- Aylien API
- Webpack
- Service Worker



Course Project

Front End Web Developer Capstone Project

In the final project, learners will combine all of the skills they've developed throughout the Nanodegree program to build an online travel app. They'll work with data sources from multiple APIs to create a dynamic travel weather planning application that helps people plan trips by generating weather forecasts for the places they're visiting.

Learners will pull together all of the JavaScript, HTML, CSS, and build tool skills and knowledge you've gained to create this application. Learners will have the flexibility to include and combine other APIs to build this final project.

Lesson 1

Intro to Build Tools

- Describe and explain the problems solved by using automated build tools.
- Inspect and analyze the activity that occurs when loading a web page in a browser.
- Describe and explain what Webpack is and how it can be used.

Lesson 2

Basics of Webpack

- Install Webpack to a computer.
- Configure webpack for automating build tasks.
- Define an entry point for a webpack bundle.
- Install and configure middleware for building an application.
- Extend Webpack functionality by installing plugins.

Lesson 3

Sass and Webpack

- Describe and explain the benefits and use cases for Sass.
 - Create CSS variables with Sass.
 - Extend and nest CSS sheets and classes with Sass.
 - Configure Webpack to use Sass controlled stylesheets.
-

Lesson 4

Final Touches

- Control variable and function scope with JavaScript IIFEs (immediately invoked function expressions).
- Optimize an application build pipeline with Webpack.
- Cache server data and websites functionality using Service Worker.

Meet your instructors.



Daniel Silber-Baker

Programmer

Daniel Silber-Baker is a programmer, poet, and educational design expert. He has a master's degree from NYU's Interactive Telecommunications Program, and his professional career has stretched across the non-profit, corporate, and academic spheres.



Rachel Manning

Full Stack Freelance Developer

Rachel is a full stack freelance developer and educator where she spent 3 years as a bootcamp curriculum developer while developing full stack freelance projects. An advocate for continued learning, she is passionate about mentoring women and underserved community in technology.



Alyssa Hope

Full Stack Developer

Alyssa is a full stack developer who was previously the lead instructor at a coding bootcamp. With a degree in international communications, her passion is to express thoughts well, whether in code or writing.

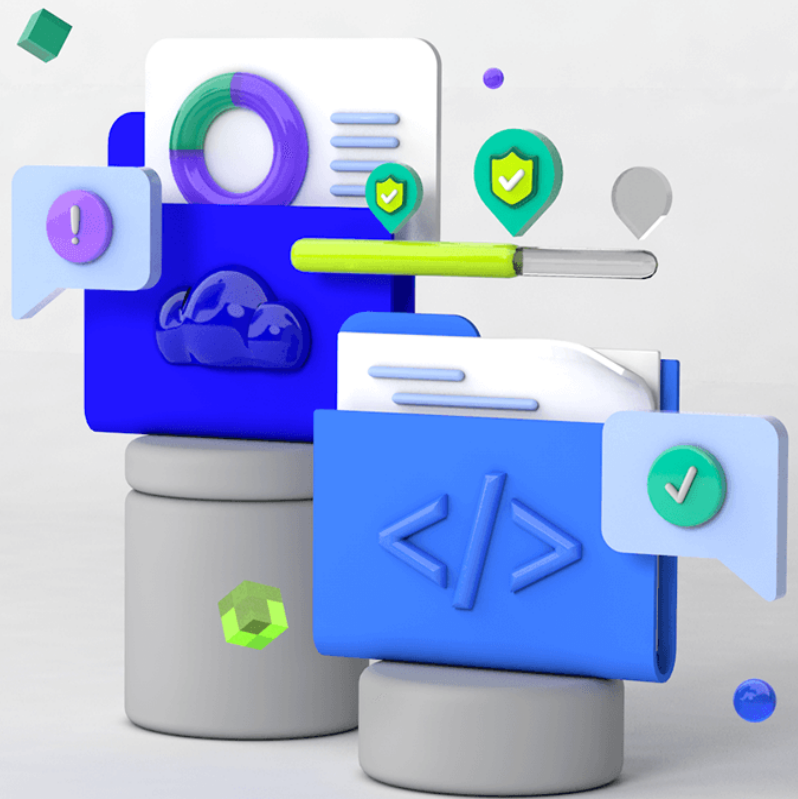


Richard Kalehoff

Course Developer

Richard is a course developer with a passion for teaching. He has a degree in computer science. He first worked for a nonprofit doing everything from front end web development, to backend programming, to database and server management.

Udacity's learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren't just multiple choice questions or step-by-step guides, but instead require critical thinking.



Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.



Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.



Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.



Custom Study Plans

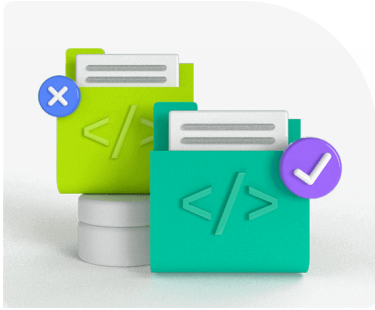
Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.



Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

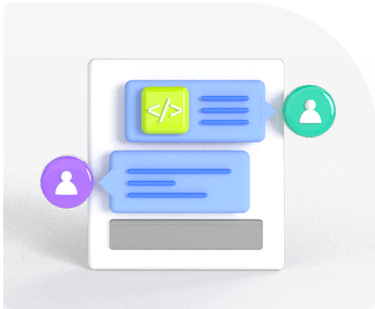
Our proven approach for building job-ready digital skills.



Experienced Project Reviewers

Verify skills mastery.

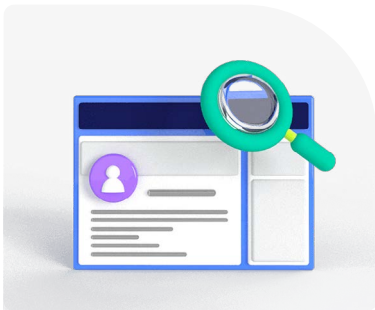
- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.



Technical Mentor Support

24/7 support unblocks learning.

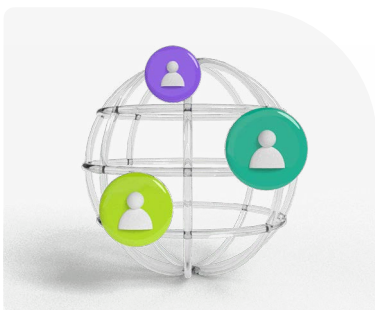
- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.



Personal Career Services

Empower job-readiness.

- Access to a Github portfolio review that can give you an edge by highlighting your strengths, and demonstrating your value to employers.*
- Get help optimizing your LinkedIn and establishing your personal brand so your profile ranks higher in searches by recruiters and hiring managers.



Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

*Applies to select Nanodegree programs only.

Learn more at

www.udacity.com/online-learning-for-individuals →