Springboard

Software Engineering

CAREER TRACK SYLLABUS

2023

Content

Overview

The demand for software engineers is at an all-time high. Companies are constantly seeking developers to build new products and applications that users love from the ground up or improve existing ones. As a result, competition is growing among employers for developers who can build products that are both powerful and easy to use.

The **Springboard Software Engineering Career Track** will teach you job-ready web developer skills, including core programming languages, tools, and technologies.

The program covers **full stack web development**, which includes both the "front end" of websites (the part that users see and interact with) and the "back end" (the part that stores and manipulates data). Full stack developers build web-based pages and applications to achieve user, business, and product goals.

By the end of the bootcamp, you'll have a complete programming skill set to succeed in a web development role.

Overview

What You'll Learn

Over the course of 9 months, you'll:

- Gain intermediate proficiency in two of the most widely used programming languages in the world — JavaScript and Python.
- Learn the ins and outs of software development theory, tools, and skills.
- Demonstrate your knowledge of software development through course projects.
- Create an interview-worthy portfolio to show off to potential employers.

How You'll Learn

- **An online curated curriculum** - developed in collaboration with San Francisco-based Rithm School and top coding instructor Colt Steele — will help you build software engineering skills and understand best practices through learning resources, exercises, and assessments.
- **Project-based learning** - including four portfolio projects covering front end, back end, and full stack development— helps you attain proficiency in software engineering.

What You'll Gain

- **1-on-1 mentor support**: You'll be matched with a mentor who will help you tackle the curriculum, provide regular feedback, and answer your questions. Your mentor will keep you accountable and give you an insider's perspective.
- **Career coaching**: You'll work through career-specific units with a career coach guiding you — from defining your strategy, and developing your resume and LinkedIn profile to networking, mock interviews, and salary negotiation.
- **A certificate of completion and Job Guarantee**: You'll graduate with a certificate from Springboard backed by our Job Guarantee — if you don't land a job within six months of graduating, we'll give you a full refund. **Terms apply**.

Key Program Details

9 months
Program length

20-25
Hours per week

24 total projects
20 mini-projects

4 portfolio pieces, including 2 capstones

Prerequisites
- Basic skills in HTML, CSS, and JavaScript

Tools & skills learned

HTML 5 CSS 3 JS 5 Flask web development, one drop at a time

SQL jQuery

node JS Redux

React PostgreSQL

Express JS python™

Syllabus

Units Include:

Each unit covers key aspects of **front end, back end, or full stack web development**, including database management, working with APIs, and data structures and algorithms.

The program also includes optional units if you want to learn libraries like Redux, TypeScript, or additional interview preparation techniques.

Syllabus

Front End Development with JavaScript and HTML

1. Web Development Fundamentals

In this unit, you'll learn about the languages and technologies most commonly used in the industry and why you would use one language over another. You'll also start with a refresher of some JavaScript fundamentals before moving on to more intermediate content.

Topics Covered:

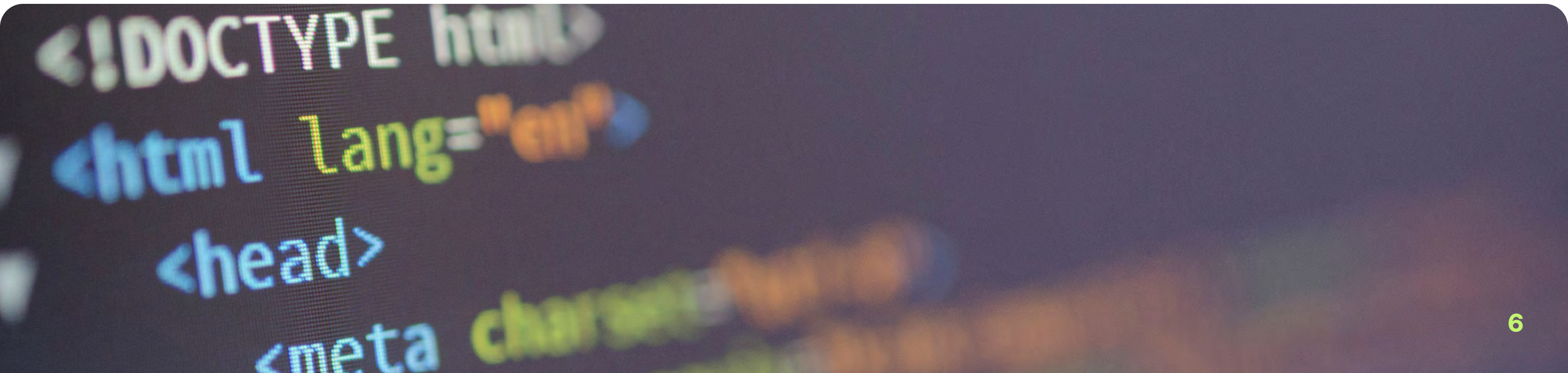
- Web development languages
- JavaScript fundamentals refresher
- JavaScript timers
- Chrome's developer tools for debugging
- Higher order functions
  - › Callback functions
  - › Writing your own callback functions

2. DOM Manipulation

JavaScript, known as "the programming language of the web," will provide the backbone of the web development stack. You'll begin to learn more intermediate skills, such as leveraging JavaScript to build sophisticated, event-driven applications using the DOM.

Topics Covered:

- Selecting elements
  - › What is the DOM?
  - › querySelector / getElementById
- Manipulating the DOM
  - › Changing text and styles
  - › DOM traversal
  - › Working with multiple elements
- JavaScript events
  - › Different ways to add event listeners
  - › Event object
  - › Event delegation



```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset
```

Front End Development with JavaScript and HTML

3. Developer Fundamentals (Git/Terminal/Github)

Before starting with any web development, you must build a sound foundation in how to work as a professional developer. In this unit, you'll learn essential tools that will become second nature to you as you delve into trickier units. Get a feel for using your Terminal's command line and version control through Git and GitHub.

Topics Covered:

- Terminal fundamentals
  - › Navigating in the terminal
  - › Creating files and folders
- Git and GitHub fundamentals
  - › What is Git
  - › Creating repositories, local workflow
  - › Branching
  - › Merge conflicts
  - › What is GitHub + signing up for an account
  - › Cloning/pushing to GitHub

4. Testing with Jasmine

It's time to dive deeper into JavaScript. You'll start by learning one of the most fundamental skills any developer needs to know: testing. As strange as it might sound now, you'll learn to write code that tests your code!

Topics Covered:

- Terminal fundamentals
  - › Testing with Jasmine
  - › Unit testing
  - › Jasmine with HTML

5. Advanced Array Methods

In this unit, you'll ensure your knowledge of JavaScript is at a professional level by learning its latest features and some of its complexities, too.

You'll focus on practical manipulations of the array data structure. These trickier parts will take a bit more time to master, but you'll see them everywhere as you learn more advanced libraries, including React. You'll use these skills in later units to refactor the code on some of your earlier projects.

Topics Covered:

- forEach, map, filter
- reduce
- some, every
- find, findIndex

Front End Development with JavaScript and HTML

6. JavaScript ES2015

Programming languages have to be updated to keep them modern. In this unit, you'll learn the version of JavaScript used by most workplaces, ECMAScript 2015 or ES2015 for short. The changes let developers write less code to achieve the same functionality.

Topics Covered:

- Arrow functions
- Rest / spread
- Object enhancements
- Destructuring

7. Fruit Search Project

In this project, you'll create a small predictive text search app for fruit using DOM Manipulation, advanced array methods, and the features of the ES2015 update in the previous unit.

Topics Covered:

- Plan and build an autofill search bar for various types of fruit
- Prepare for your code review
- Live code review

8. Object-Oriented Programming

In object-oriented programming, data is structured into classes, and coded procedures manipulate the data into objects. In this unit, you'll learn the ins and outs of object-oriented programming and work on projects that involve refactoring and creating an object-oriented class to model a car.

Topics Covered:

- ES2015 classes
- Inheritance
- `this`
- `bind`

Front End Development with JavaScript and HTML

9. jQuery and Twitter Bootstrap

Libraries provide new functionality for you to use without having to implement the functions and methods yourself. The two most popular libraries in web development are jQuery and Bootstrap. In this unit, you'll learn jQuery and then have the option to work through a Twitter Bootstrap sub-unit. The unit will also provide exercises on each to reinforce your learning.

Topics Covered:

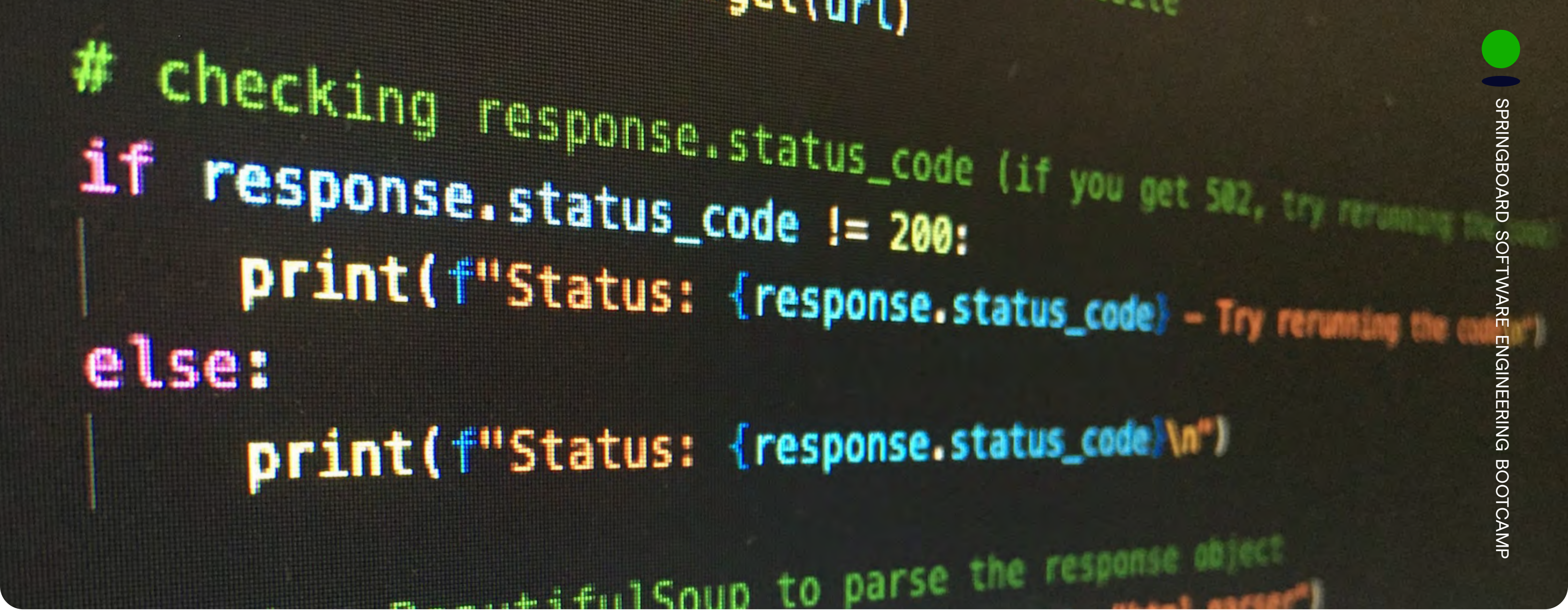
- jQuery
  - DOM manipulation
  - Selector caching
  - Event delegation
- Bootstrap **(optional)**
  - Colors, buttons, and badges
  - Containers and alerts
  - Jumbotrons
  - Images and cards
  - The grid system
  - Responsive grid
  - Auto columns
  - Nested grids
  - Forms
  - Navbars
  - Font awesome
  - JavaScript components

10. How the Web Works and AJAX

Now that you've gotten past some of the more challenging parts of JavaScript, it's time to learn about how it fits in the full stack web development ecosystem. So far, you've been using JavaScript to manipulate data on a web page, but JavaScript can also be used to fetch external data with a series of technologies known as AJAX. Before you dive deep into AJAX, you'll get comfortable with how the web works and how to make HTTP requests, as well as one of the tougher topics in JavaScript, asynchronous code. Finally you'll begin learning about data structures and algorithms through Big O Notation.

Topics Covered:

- How the Web Works
  - HTTP
  - DNS
  - GET vs POST
- async/await
  - Asynchronous code
  - Async functions
- APIs
  - XML
  - JSON
  - Curl requests
- AJAX with axios
  - AJAX
  - Axios
- Big O Notation
  - Problem with timers
  - Counting operations
  - Space complexity



```
get(url)
# checking response.status_code (if you get 502, try rerunning the code)
if response.status_code != 200:
    print(f"Status: {response.status_code} - Try rerunning the code\n")
else:
    print(f"Status: {response.status_code}\n")

BeautifulSoup to parse the response object
```

Syllabus

Back End Development with Python

11. Python Fundamentals

Now that you're comfortable writing front end code, let's move to the back end. You'll start with an introduction to the second language in this course, Python. You'll get comfortable with the language just like you did with JavaScript and also recognize some of the key differences and similarities between Python and JavaScript.

Topics Covered:

- Python introduction
- Data structures in Python
- Intermediate Python
- Object orientation in Python

12. Flask Fundamentals

Once you have a good grasp of Python, you'll move on to building web applications using the highly popular web framework Flask. You'll build full stack applications and learn about essential back end concepts like server-side templates, rendering, redirecting, cookies, sessions, and more. Finally, you'll starting using the divide and conquer problem solving methodology to enhance your data structures and algorithms skill set.

Topics Covered:

- Flask fundamentals
- Server side templates with Jinja
- Flask testing
- Cookies and sessions
- Divide and conquer method of problem-solving

Back End Development with Python

13. SQL

SQL is foundational for building any relational database-backed application and has been the standard for over 40 years. In this unit, you'll get started working with databases and SQL. You'll master the fundamental commands and get comfortable with aggregates, joins, and data definition language.

Topics Covered:

- What is SQL
  - Relational databases
  - Installing postgres
- SQL querying
- DDL + schema design

14. SQLAlchemy

Now that you have a solid understanding of full stack development and databases, you'll move onto building more complex web applications. You'll be introduced to an object relational mapper (ORM) called SQLAlchemy, which allows you to use your knowledge of SQL, but handle database operations in Python.

Topics Covered:

- Installing and configuring SQLAlchemy
- SQLAlchemy associations
- SQLAlchemy many-to-many relationships

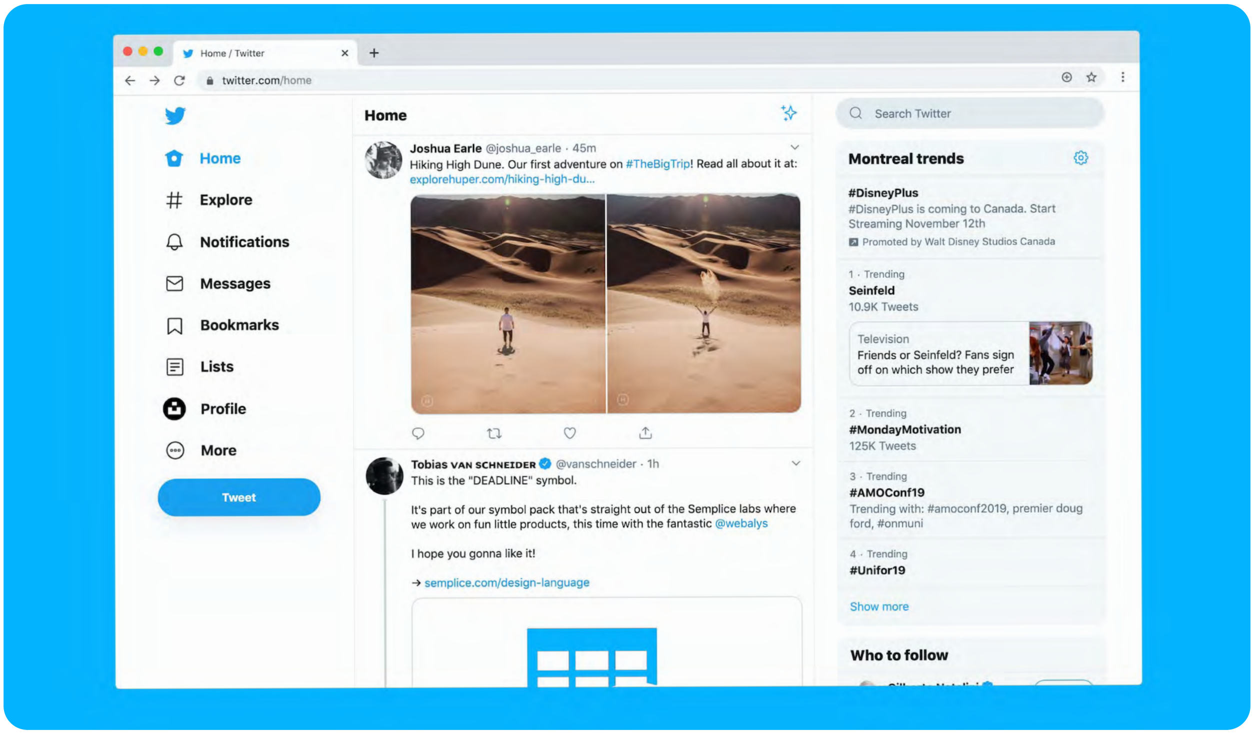
15. Intermediate Flask

In this unit, you'll start building JSON APIs and secure applications with hashed passwords, authentication, and authorization. You'll also learn how to make HTTP requests from the back end, allowing you to interact with most APIs to fetch and send data to and from external data sources. You'll then solidify your knowledge of data structures and algorithms through arrays and linked lists, before brainstorming ideas for your first capstone.

Topics Covered:

- Building JSON APIs
- RESTful JSON APIs
- Making API requests with Python and Flask
- Hashing and login
- Deploy applications to production with Heroku
- Arrays and linked lists

Back End Development with Python



Joshua Earle @joshua_earle · 45m
Hiking High Dune. Our first adventure on #TheBigTrip! Read all about it at: explorehuper.com/hiking-high-du...

Tobias VAN SCHNEIDER @vanschneider · 1h
This is the "DEADLINE" symbol.

It's part of our symbol pack that's straight out of the Semplice labs where we work on fun little products, this time with the fantastic @webalys

I hope you gonna like it!

→ semplice.com/design-language

Montreal trends

#DisneyPlus
#DisneyPlus is coming to Canada. Start Streaming November 12th
Promoted by Walt Disney Studios Canada

1 · Trending
Seinfeld
10.9K Tweets

Television
Friends or Seinfeld? Fans sign off on which show they prefer

2 · Trending
#MondayMotivation
125K Tweets

3 · Trending
#AMOConf19
Trending with: #amoconf2019, premier doug ford, #onmuni

4 · Trending
#Unifor19

Who to follow

16. Twitter Clone

This unit encompasses a project where you will create a Twitter clone called Warbler. This code base is larger than any previous codebases, and you'll have the additional task of adding functionality. You'll also get practice encountering unfamiliar elements in the codebase and fix bugs.

Topics Covered:

- Create a clone of Twitter and add functionality to it
- Practice reading a large codebase
- Perform some debugging

17. Intermediate Terminal (Optional)

In this unit, you'll learn advanced command line keywords like find and grep.

Topics Covered:

- Github workflows
- Hard links
- Symbolic links
- Symlinks
- The Find command
- Grep intro
- Recursive Grep

Syllabus

Back End Development with NodeJS

18. Node Fundamentals

Now that you're comfortable building back end applications in Python, let's revisit JavaScript, but on the back end. You'll learn about Node.js, one of the most popular technologies on the web and how to use its asynchronous model to build performant applications.

Topics Covered:

- Intermediate Github
  - Pull requests
- Command line scripts with Node and NPM
  - What is Node + installing Node
  - What is NPM
  - Command line scripts with Node
- Async in detail (promises / callbacks)
  - Async review
  - Callbacks
  - Promises
  - Async/await
- Testing with Jest and Node
  - Installing Jest
  - Matchers
- Data structures: Stacks and queues

Back End Development with NodeJS

19. Express Fundamentals

The most popular Node framework is Express.js. Many major companies like PayPal, Uber, and IBM use Express in their tech stack. Almost every company where you'd be using Node, you'll also use Express. You'll learn some of the intricate parts of this framework and use it to build an API before gaining experience with the programming technique called recursion.

Topics Covered:

- What is Express
  - The request / response cycle with Express
  - Testing with Supertest
  - Error handling with Express
- Routing and middleware
  - Express Router
  - Using middleware
  - Testing middleware
- Recursion
  - The callstack
  - Counting recursively
  - Base case
  - Sum array recursively
  - Doubler Iterative and Doubler Recursive
  - Recursive Big O

20. Express and Postgress

Take your knowledge of SQL and connect it with Node and Express using the Node-pg module. You'll build a full stack application with Postgress and Express. You'll learn how to work with asynchronous code, handle common security exploits, write basic CRUD (Create, Read, Update, Delete) queries and make your own simple ORM.

Topics Covered:

- Node-pg introduction
  - Getting started with Node-pg
  - The Node / SQL Ecosystem
- Object Oriented Node-pg
  - Strategies to implement an ORM:
  - Simple OO Approach
  - Smart OO Model

21. Intermediate Express

You'll continue to explore more of the advanced features of Express including authentication and authorization using JSON Web Tokens.

Topics Covered:

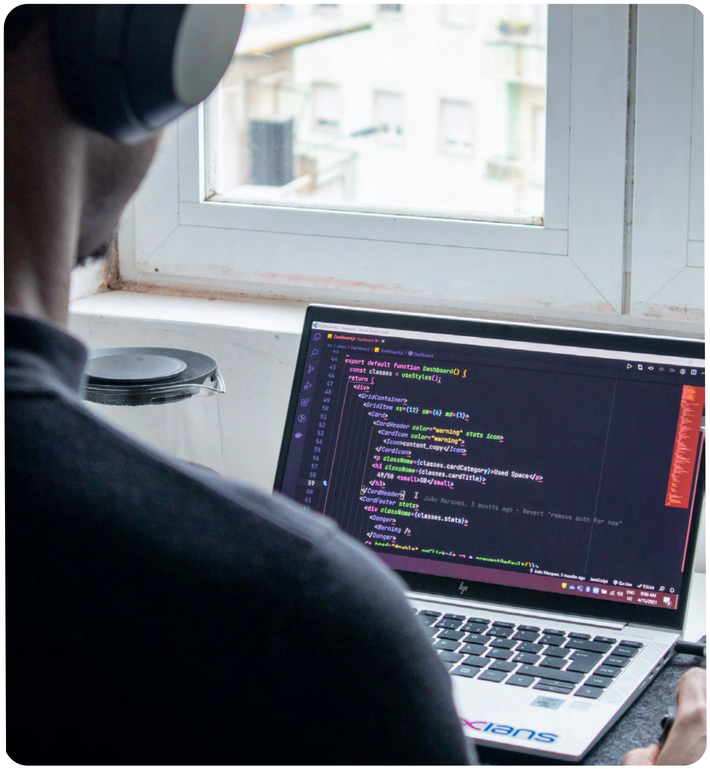
- Hashing and JSON Web Tokens (JWT)
  - API Validations
  - Express testing practices
  - Documenting APIs
- Authentication and authorization with bcrypt and JWTs
  - Storing passwords securely with bcrypt
  - Using JWTs for Auth
- Optional study: npm scripts, Passport JS, websockets, etc.

Syllabus

Front End Development with ReactJS

22. ReactJS Fundamentals

Now that you've built a few full stack applications, it's time to move back to the front end and learn a framework. You'll be focusing on one of the most popular and rapidly growing frameworks, React.js. Written by Facebook, this framework allows for building robust applications that can scale easily. Once you have a solid grasp on what React is and how to build components and simple applications, it's time to layer on more complexity with a few additional built-in hooks. You'll learn how to include side effects in your components with useEffect, manage state with useContext, and handle complex states with useReducer.



Topics Covered:

- React introduction
  - What is React?
  - Webpack / Babel / JSX
- Props
  - Default props
  - Proptypes
  - props.children
- Create React App
- Events and State
  - useState
  - useState patterns
- Testing with RTL
  - Smoke tests
  - Snapshot tests
  - RTL Query methods
- Component Design
  - Component reuse
  - Dumb components
  - Dice design
- React forms
  - Forms with React
  - Testing Forms
- React Effect and Refs
  - useEffect callback
  - useEffect empty array
  - useEffect cleanup
  - useRefTress
- Trees
  - Implementing a Tree
  - Depth first search
  - Breadth first search
- Writing custom hooks
  - useToggle Custom Hooks
  - localStorage Custom Hook

Front End Development with ReactJS

23. React Router and React History

In this unit, you'll learn about client-side routing. If you have visited a website that has changing URLs but displays the same page, you know you're dealing with a React Router. It is used ubiquitously online and across the corporate world.

Topics Covered:

- Using React Router and React Router patterns
  - Link and navLink
  - Redirect/switch
- React History
  - Class components
  - Classes and binding
  - Component lifecycle
  - Render props
  - Hooks and closure
  - React API context

24. React Jobly

In this unit, you'll implement a React front end for the Jobly application you created in a cumulative project (see project description below). You'll download a starter code for the back end, design a component hierarchy, make an API helper class, create a routes file, and more. You'll also submit three potential website ideas for your second capstone.

Topics Covered:

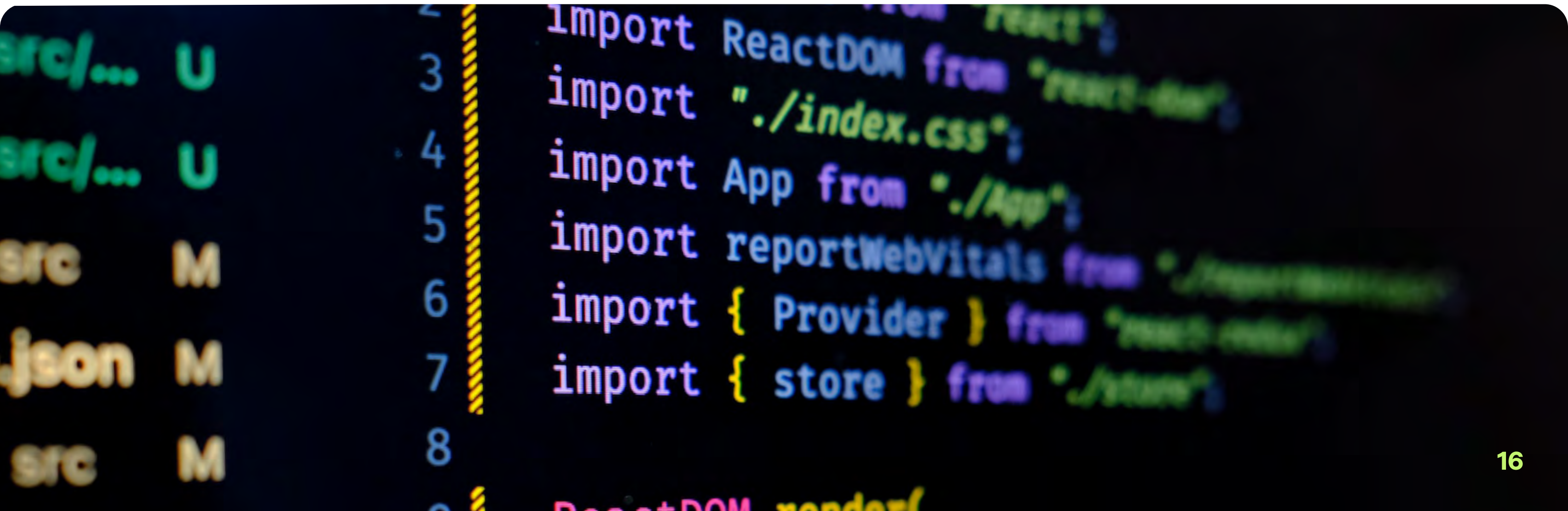
- Build a React front end for your Jobly app
- Preparing for your second capstone project

25. Data Structures and Algorithms

In this unit, learn how to architect applications and make the right tradeoffs regarding performance using data structures and algorithms.

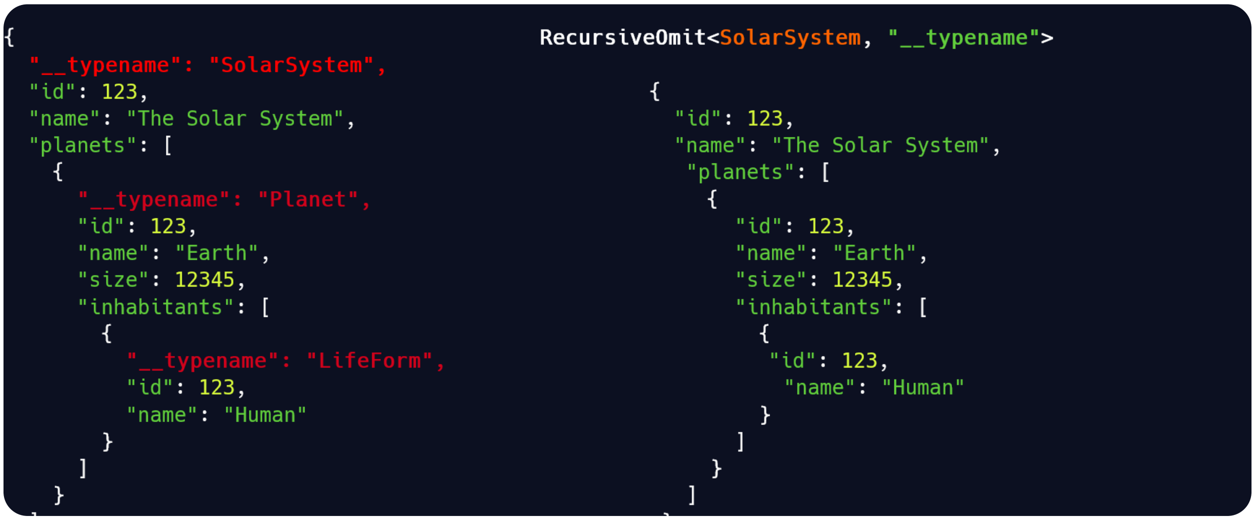
Topics Covered:

- Binary search trees
- Graphs
- Sorting algorithms
- Heaps



```
import ReactDOM from "react-dom";
import "./index.css";
import App from "./App";
import reportWebVitals from "./reportWebVitals";
import { Provider } from "react-redux";
import { store } from "./store";

ReactDOM.render(
```



```
{
  "__typename": "SolarSystem",
  "id": 123,
  "name": "The Solar System",
  "planets": [
    {
      "__typename": "Planet",
      "id": 123,
      "name": "Earth",
      "size": 12345,
      "inhabitants": [
        {
          "__typename": "LifeForm",
          "id": 123,
          "name": "Human"
        }
      ]
    }
```

```
RecursiveOmit<SolarSystem, "__typename">

         {
           "id": 123,
           "name": "The Solar System",
            "planets": [
              {
                "id": 123,
                "name": "Earth",
                "size": 12345,
                "inhabitants": [
                  {
                   "id": 123,
                    "name": "Human"
                  }
                ]
              }
            ]
```

Front End Development with ReactJS

26. Redux (Optional)

As your React applications grow, managing global state can become quite a challenge. While the Context API is an excellent option, sometimes you need a bit more when scaling. Redux is another option for state management that has the ability to scale to massive codebases including those at Facebook.

Topics Covered:

- Redux introduction
  - What is Redux?
- React/Redux
  - Integrating React with Redux
  - React/Redux hooks
  - React Redux patterns
- Async Redux
  - Async redux introduction
  - Redux thunk

27. TypeScript (Optional)

Typescript, a superset of JavaScript, has become increasingly popular in recent years. If you'd like to use Typescript for your final capstone, complete this unit before starting your project. Otherwise, complete this unit after you complete the course so you finish the course on a shorter timeline. Learning TypeScript will make you more competitive in your job search while keeping your coding skills sharp.

Topics Covered:

- Intro to Typescript
- Defining Types
- Types in other languages
- VS Code and Typescript
- Types and methods
- Object types
- Union types
- DOM casting

Building Your Portfolio Projects

While working through this program, you will complete four end-to-end projects to showcase in your programming portfolio, including two cumulative projects and two capstone projects.

These projects are an integral part of the curriculum that will allow you to apply all of the skills you develop while working through the program. You'll gain hands-on experience with each stage of the development process, from designing your web page or application to coding and testing your code to final review and putting it into production.

Building Your Portfolio Projects

Cumulative Project 1: Hacker News Clone

(Front End Development with JavaScript and HTML)

In this first major project, you'll make a functional clone of Hacker News from your API using the front end skills that you have been working on throughout the course. The site will have the same stories that someone can read on Hacker News, users will be able to create accounts and log in, save their favorite articles, and upload stories of their own.

Cumulative Project 2: Jobly

(Back End Development with Python)

You will take your back end knowledge and build a full stack application called Jobly, which will be used similarly to LinkedIn — where people can explore and apply for jobs online — but on a smaller scale. You'll make a queryable API for companies, add in database tables for users and jobs, and add in authentication and authorization.

Capstone Project 1

In this capstone project, you will create a database-driven website with an external API of your choice. You will submit portions of the project for evaluation three times: several project ideas, a project proposal, and the complete GitHub repo.

- Step one: Initial project ideas
- Step two: Project proposal
- Step three: Schema design and API selection
- Step four: Coding user flows
- Step five: Polishing your application
- Step six: Documenting and submission

Capstone Project 2

This capstone is similar to the first, but with Node/Express/React and more robust features. You will create a database-driven website with an external API of your choice and submit portions of the project for evaluation three times: several project ideas, a project proposal, and the complete GitHub repo.

- Step one: Initial project ideas
- Step two: Project proposal
- Step three: Source your data
- Step four: Coding user flows
- Step five: Polishing your application
- Step six: Documenting and submission

Assessments

Periodically throughout the program, you will be asked to complete assessments to help internalize and reinforce the learning material. Depending on the assessment, you will answer questions about topics you've learned or complete a coding project.

Assessment Breakdown

I: Front End Development with JavaScript and HTML

- Meme Generator: Build an app to generate memes from any image and text you input.
- Jeopardy! : Build an interactive game of Jeopardy!

II: Back End Development with Python

- Foreign Exchange Converter: Create a Flask app that converts between two currencies.
- Database DJ: Create a small Flask app to make song playlists.
- Lucky Number (Optional): Make a small full stack Flask application with an API and a front end.

III: Back End Development with NodeJS

- Broken App or Node Express 1: Fix a broken Node/ Express app and refactor some of the code.
- Even More Broken App: Debug a Node/Express app.

IV: Front End Development with ReactJS

- Mad Libs: Create an interactive game of MadLibs in React.
- Snack or Booze: Update an existing site as part of a rebrand for a coding cafe turned bar and restaurant.
- Star Wars **(Optional)**: If you have completed the Redux unit, you have the option to complete an assessment involving reading an unfamiliar codebase.

Career Support

Career units throughout the bootcamp will help you create a tailored job search strategy based on your background and goals. Learn to craft a resume that stands out from the pack, evaluate companies and roles, ace interviews, and negotiate the best possible salary.

Your career coach will be with you every step of the way, offering feedback and providing personalized tips based on your goals.

Topics Covered

- Types of industry roles
- Job search strategies
- Building a network and using it to land interviews
- Creating a high-quality resume, linkedin profile, and cover letter
- Preparing for technical and non-technical interviews
- Successful negotiation
- Building a portfolio

Build the Skills and Confidence to Transform Your Career

Learn through projects. Work 1-on-1 with a mentor and career coach.
Land a job or your money back.

HANDS-ON LEARNING

A high quality, project-based curriculum designed by industry experts helps students master their area of study so they're career ready.

REAL HUMAN SUPPORT

Students receive the dedicated support of a personal mentor, career coach, and student advisor, plus 24/7 access to a peer community.

MORE FREEDOM

100% online classrooms give students the flexibility they need to continue working while attending Springboard.

JOB GUARANTEE

Students who are job-qualified will get a job within 6 months of graduating or get a full refund of their tuition. **Terms apply**.

Springboard Students Achieve Life-Changing Outcomes

NUMBER OF ENROLLED STUDENTS

1,714

Enrolled students in the Software Engineering Career Track since January 2020. [1]

September 2022

12 MONTH JOB PLACEMENT RATE

96.8%

Of job-qualified individuals who reported an offer, received it within 12 months of graduation. [2]

September 2022

AVERAGE SALARY INCREASE

$27,608

Average salary increase of Software Engineering students who provided pre- and post-course salaries. [3]

September 2022

[1] Number of students refers to all students who enrolled in the career track excluding any that were refunded due to cancelation in the first 7 days following course start.

[2] Job-qualified individuals defined as all graduates who maintained Job Guarantee eligibility (terms are from the Software Engineering Career Track Job Guarantee) throughout their job search ("Job-Qualified Graduates"), or Job Guarantee-eligible students who receive a job regardless of completion status ("Early Offerees").

[3] Data on compensation was not self-reported by 45 students who reported receiving offers.

Ready for the next step?

Learn more and apply **here**

Questions? We're here to help

Email us at **hello@springboard.com**
or call **+1.415.966.2533**

Springboard