UDACITY

NANODEGREE PROGRAM SYLLABUS

React

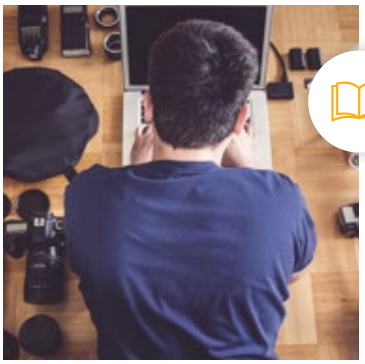
Overview

Students will go on a project-based learning journey to build declarative user interfaces (UI components) for web applications using React, make state more predictable in their applications with Redux, and make their apps more efficient and robust using newer React features and the Jest testing library. Students will also have an optional opportunity to develop mobile apps for iOS and Android devices using React Native.

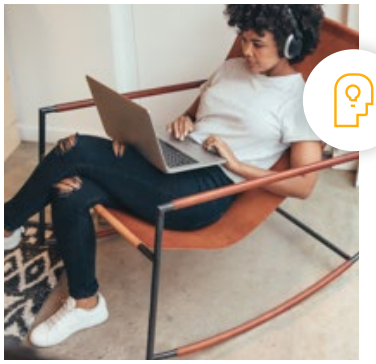
Educational Objectives:

- Create interactive React components for their application by using JSX to render UI, managing state, and handling lifecycle events.
- Write more powerful applications by using Redux to manage the global store, handle asynchronous network requests, and efficiently pass data through the UI.
- Build more efficient, robust React apps by using recent React features such as Hooks and testing React components using Jest testing library.
- Develop a mobile app that can run on both iOS and Android devices using React Native (optional).

Estimated Time:
2 Months at
5-10hrs/week



Prerequisites:
- HTML & CSS
- JavaScript
- Asynchronous JavaScript (AJAX)
- Web forms (front-end elements such as HTML, CSS, styling, and accessibility)



Flexible Learning:
Self-paced, so you can learn on the schedule that works best for you

Technical Mentor Support:
Our knowledgeable mentors guide your learning and are focused on answering your questions, motivating you and keeping you on track

Course 1: React Fundamentals

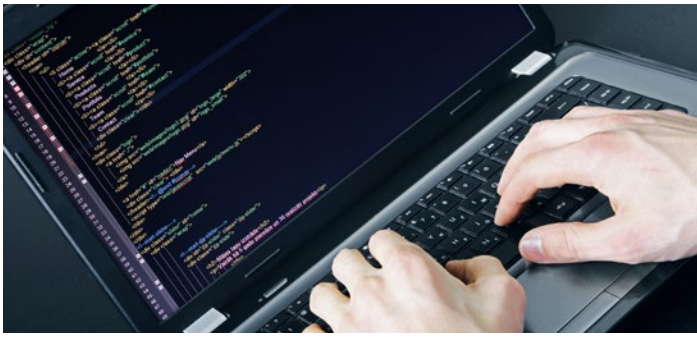
Mastering React begins with learning your fundamentals, and this can pose a bit of a challenge, because while the modularity of the React ecosystem makes it really powerful for building applications, there is a great deal to learn. So we'll break everything down, and enable you to learn the foundational parts of the React ecosystem that are necessary to build production-ready apps.

Course Project
MyReads: A Book Tracking App

In this project, you will create a React application from scratch and utilize React components to manage the user interface. You'll create a virtual bookcase to store your books and track what you're reading. Using the provided Books API, you'll search for books and add them to a bookshelf as a React component. Finally, you'll use React Hooks to build the functionality to move books from one shelf to another.

	LEARNING OUTCOMES	
LESSON ONE	**Why React**	• Identify why React was built • Use composition to build complex functions from simple ones • Leverage declarative code to express logic without control flow • Identify functional JavaScript concepts in React
LESSON TWO	**Rendering UI with React**	• Use create-react-app to create a new React application • Create reusable React components with composition • Use JSX to create user interface elements
LESSON THREE	**State Management**	• Manage state in applications • Use props to pass data into a component • Create functional components focused on UI rather than behavior • Add state to components to represent mutable internal data • Update component state with useState • Use PropTypes to typecheck and debug components • Use controlled components to manage input form elements

LESSON FOUR	**Hooks**	• Conceptualize the lifecycle of a component • Explain how Hooks in React enable state and lifecycle in functional components • Use React's useState hook to manage state • Use React's useEffect hook for HTTP requests and side effects
LESSON FIVE	**Manage App Location with React Router**	• Use React Router to add different routes to applications • Use state to dynamically render a different "page" • Use React Router's route component • Use React Router's link component



Course 2: React & Redux

Redux excels at state management, and in this course, you'll learn how React and Redux work together to make your application's state bulletproof. As with the previous course, the React & Redux course offers hands-on learning to build projects. Here, you'll leverage React with Redux to build "Would You Rather," a popular party game.

Course Project
Employee Polls Web App

You have been asked by HR to build a web app for creating polls for coworkers where an employee can post a scenario question with 2 possible responses and then other employees respond. Employees can then vote on these responses and see which have the most votes. The goal is to improve collaboration and transparency within the company. The web app will provide a dashboard that lists every employee ordered by the number of polls they've created and answered.

		LEARNING OUTCOMES
LESSON ONE	**Managing State**	• Recognize how state predictability improves applications • Create a store to manage an applications state • Leverage store API: getState(), dispatch(), and subscribe() • Create actions and action creators that describe state changes • Create Reducers that return state • Use Reducer Composition to handle independent parts of state
LESSON TWO	**UI + Redux**	• Create the same API that Redux uses • Build intuition for when to use Redux
LESSON THREE	**Redux Middleware**	• Identify the benefits of middleware within the Redux cycle • Apply middleware to a Redux application • Build your own Redux middleware

LESSON FOUR	**Redux with React**	• Combine Redux with the popular React library • Identify when to use component state vs. Redux state
LESSON FIVE	**Asynchronous Redux**	• Explain the pitfalls of asynchronous requests in Redux • Leverage Thunk middleware to support asynchronous requests • Fetch data from a remote API using Thunk and Saga
LESSON SIX	**React-Redux Bindings**	• Install the React-Redux bindings • Leverage React-Redux bindings to extend app functionality • Use the provider to pass a store to component trees
LESSON SEVEN	**Implementing React + Redux**	• Use connect() to access store context set by the provider • Build a complex, real-world application • Add Redux to an application scaffolded with Create React App • Normalize state shape to keep application logic simple with scale
LESSON EIGHT	**Testing with Jest**	• Install and configure Jest to begin testing your codes. • Understand the common testing patterns for React components • Build unit tests and test renderers using Jest • Test asynchronous code and mock API calls

Course 3: React Native (OPTIONAL)

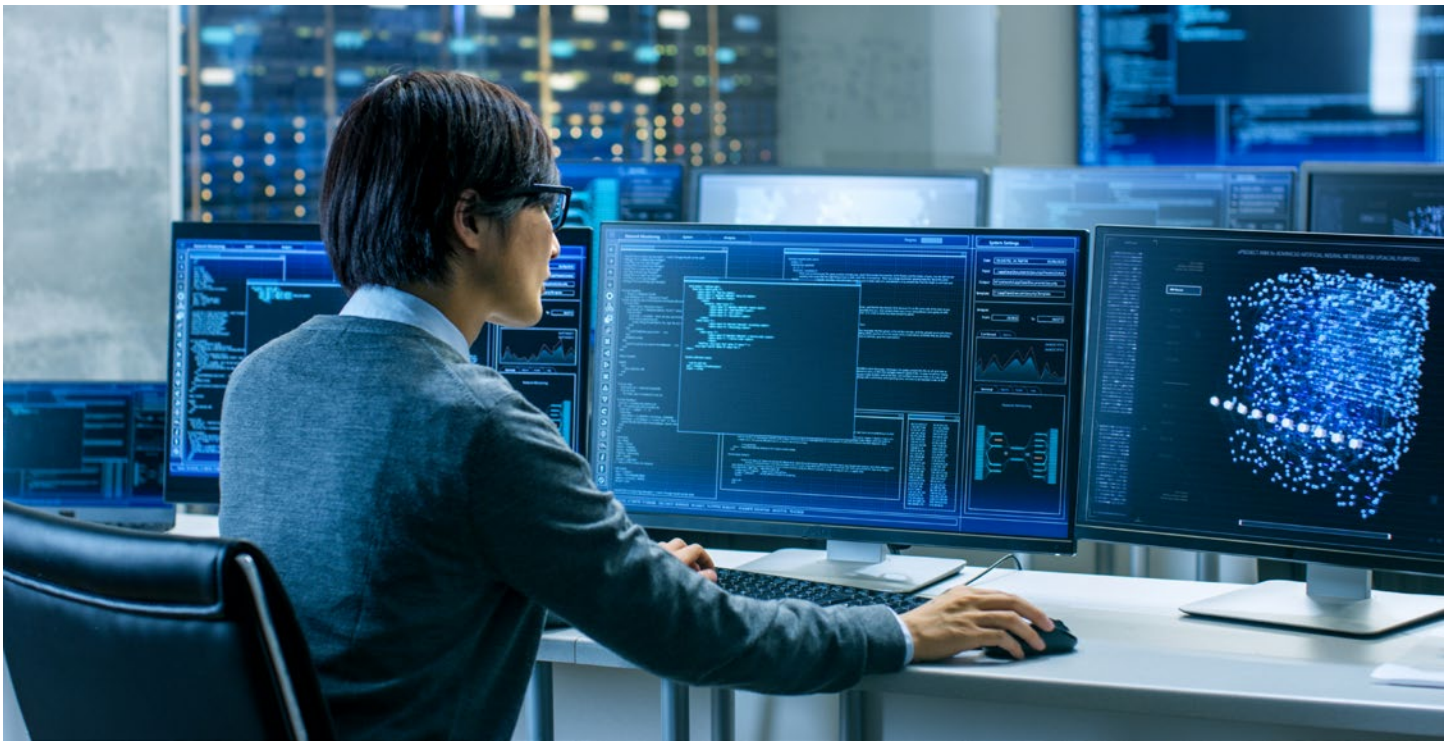
Develop React applications that run on both iOS and Android devices using React Native. You'll explore everything from setting up a proper development environment, building and styling a cross-platform mobile application. You'll incorporate native APIs such as geolocation and local notifications, and even learn how to get your app ready for the App and Google Play Store.

Course Project [Optional]: Build a Customer Relationship Management App

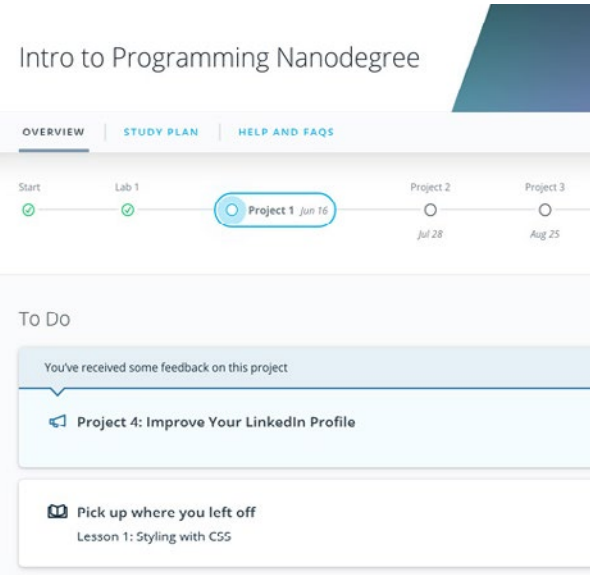
In this project, you'll use React Native to build a mobile customer relationship management app that lists the customer information by region along with their details. You will create key React Native components such as cards, menus, and tabs, integrating animations to make your app fully styled for your business branding.

		LEARNING OUTCOMES
LESSON ONE	**Introduction to React Native**	• Identify the ideology behind React Native • Set up an ideal development environment • Inspect and debug applications • Identify fundamental differences between web and native apps • Identify differences between Android and iOS platforms • Identify key strengths and weaknesses of React Native among other mobile development tools and platforms
LESSON TWO	**Mobile Screens and Styling**	• Leverage common React Native components • Recognize the core philosophies and techniques of CSS flexbox • Identify key differences between flexbox on the web and React • Native's implementation of flexbox • Style applications with CSS in JS • Identify best practices in how professionals handle styling • Identify differences in use cases between styling with inline styles, object variables, and Stylesheet API

LESSON THREE	**Navigation in React Native**	• Manage navigation through a React Native application • Leverage common React Native components for Navigation • Identify fundamental differences between web and native apps for Navigation
LESSON FOUR	**Data Management with Redux**	• Incorporate Redux and Hooks to manage shared application state and access synchronous APIs • Read and write data to Redux global store • Create forms in React Native applications
LESSON FIVE	**Native Features**	• Utilize AsyncStorage to persist global application data • Incorporate geolocation, animations, notifications, and ImagePicker to take advantage of device features and data



Our Classroom Experience

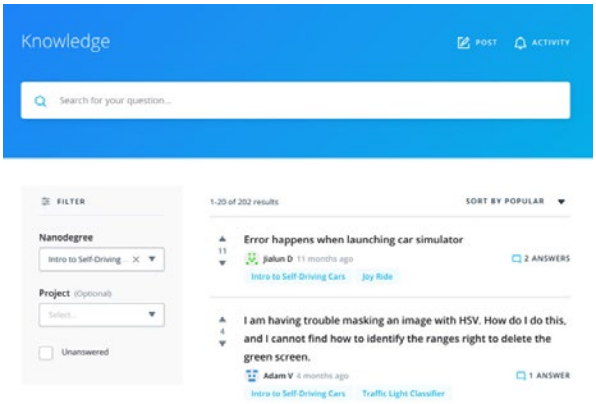


REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

KNOWLEDGE

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

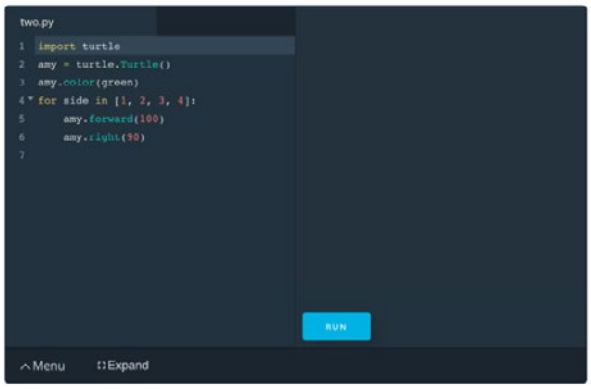


WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.



CUSTOM STUDY PLANS

Create a custom study plan to suit your personal needs and use this plan to keep track of your progress toward your goal.

PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

Learn with the Best

Andrew Wong

INSTRUCTOR

Andrew is a course developer who enjoys making the world a better place through code. He first discovered his passion for teaching as an instructor at App Academy, and continues to enjoy empowering students to advance their education.

Michael Tsamis

SENIOR SOFTWARE ENGINEER

Michael N. Tsamis is a senior software engineer at Catchpoint Systems Inc. He graduated summa cum laude in 2011 with a Bachelor of Science degree in Computer Information Systems. He is also an active member and speaker for React NYC, a Meetup group for ReactJS and React Native developers in New York City.

Alyssa Hope

SENIOR SOFTWARE ENGINEER

Alyssa is a full stack developer, focusing for the last few years in Javascript. She also spent time as the lead instructor of a coding bootcamp. Her passions are clean code, clear thinking, and mentorship.

Tyler McGinnis

INSTRUCTOR

Tyler found his love for teaching at DevMountain, where he was lead instructor and curriculum engineer. He's a Google Developer Expert and is entrenched in the React community organizing React Utah, and running React Newsletter.

Richard Kalehoff

INSTRUCTOR

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

All Our Nanodegree Programs Include:

EXPERIENCED PROJECT REVIEWERS

REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve

TECHNICAL MENTOR SUPPORT

MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions

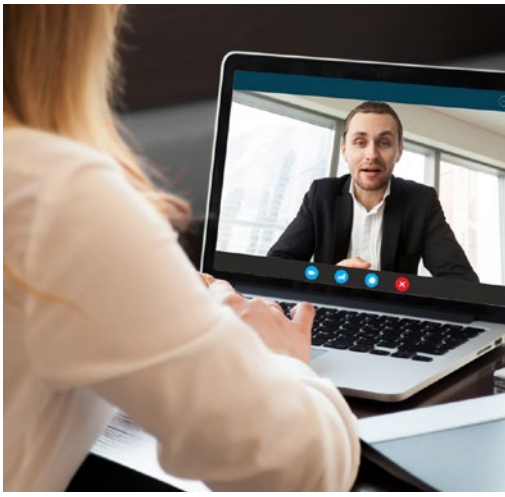
PERSONAL CAREER SERVICES

CAREER SUPPORT

- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW



WHY SHOULD I ENROLL?

This program is designed to help you take advantage of the growing need for skilled React developers. Prepare to meet the demand for qualified React developers that can respond to real-life, high-stakes workplace challenges.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

The skills you will gain from this Nanodegree program will qualify you for jobs in several industries as countless companies are trying to build superior declarative user interfaces for the web.

HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

The course is for individuals who are looking to advance their React developer careers with skills in a burgeoning field.

ENROLLMENT AND ADMISSION

DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

No. This Nanodegree program accepts all applicants regardless of experience and specific background.

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

A well-prepared student is currently able to build front-end web applications with:

- HTML & CSS
- JavaScript
- Asynchronous JavaScript (AJAX)
- Web forms (front-end elements such as HTML, CSS, styling, and accessibility)

You should also have familiarity with the following:

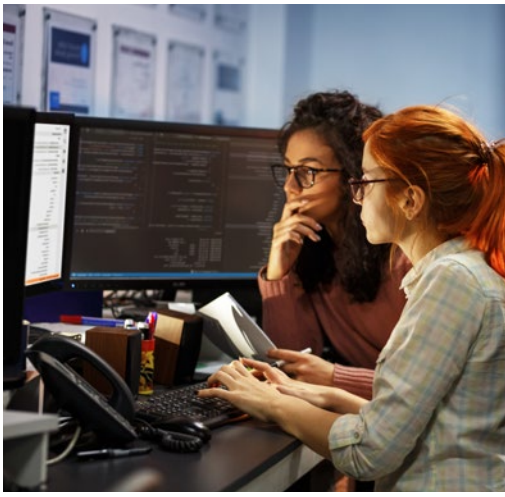
- The Document Object Model (DOM)
- Web Accessibility Standards
- Using Node Package Manager (NPM)
- Git & GitHub
- Unix/Linux Command Line Basics

FAQs Continued

IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

We have a number of Nanodegree programs and free courses that can help you prepare, including:

- Front-End Web Developer Nanodegree Program
- Intro to HTML and CSS
- JavaScript Design Patterns
- Front End Frameworks
- Version Control with Git
- GitHub & Collaboration
- Asynchronous JavaScript
- ES6 - JavaScript Improved



TUITION AND TERM OF PROGRAM

HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The React Nanodegree program is comprised of content and curriculum to support two (2) projects. We estimate that students can complete the program in two (2) months, working 5-10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the **Terms of Use** and **FAQs** for other policies regarding the terms of access to our Nanodegree programs.

CAN I SWITCH MY START DATE? CAN I GET A REFUND?

Please see the Udacity Nanodegree program FAQs for policies on enrollment in our programs.

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

- React 16.8 or above that includes Hooks (The most recent version--17.0.2--is recommended).
- Redux Saga (optional)
- Jest testing Library
- React Native 0.64 (React Navigation 5.x)