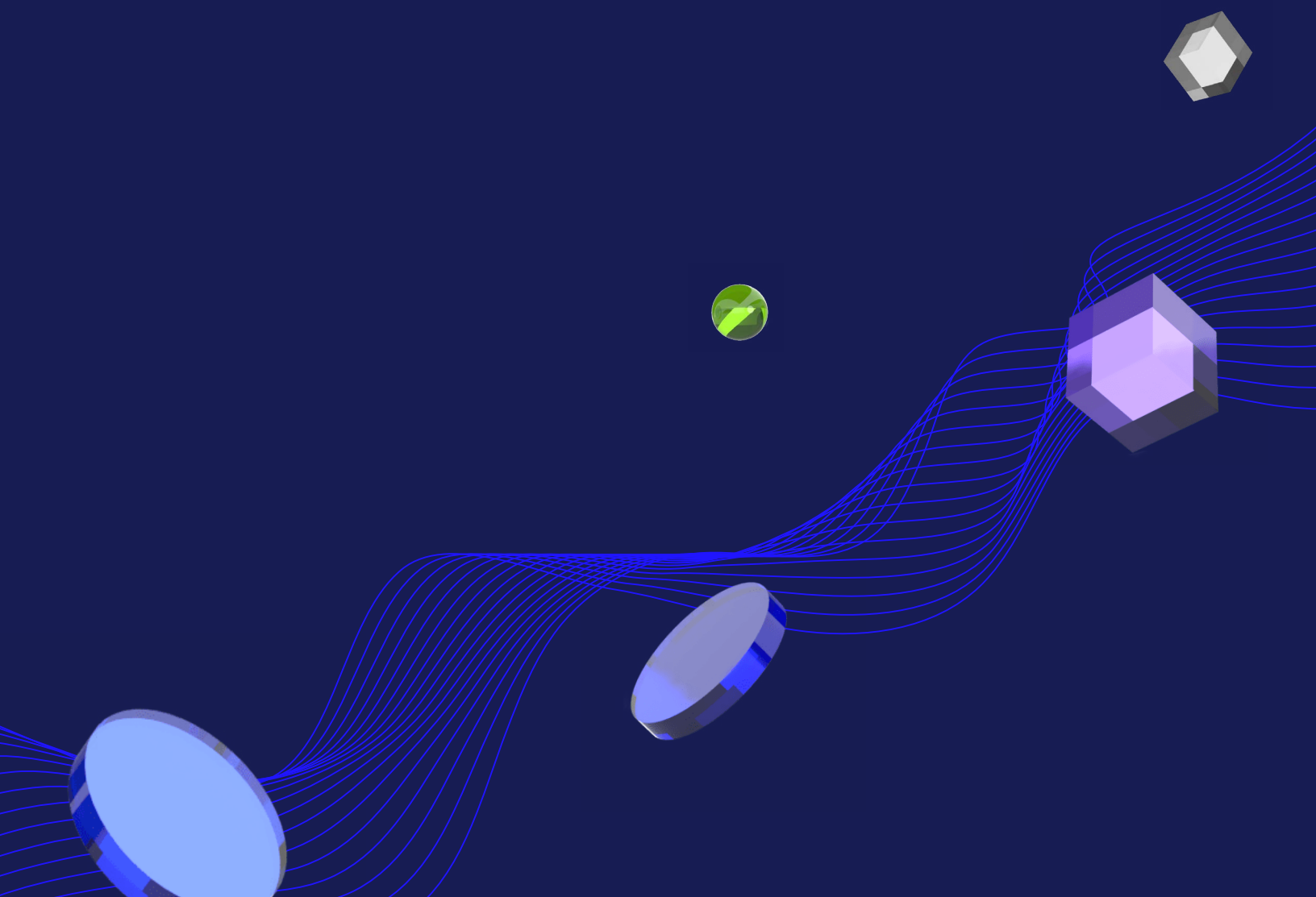
INDIVIDUAL LEARNERS

UDACITY

SCHOOL OF PROGRAMMING & DEVELOPMENT

React

Nanodegree Program Syllabus



Overview

Learners 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. Learners will also have an optional opportunity to develop mobile apps for iOS and Android devices using React Native.

Learning Objectives

A graduate of this program will be able to:

- 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).

Program information

Estimated Time

2 months at 5-10hrs/week*

Skill Level

Intermediate

Prerequisites

Learners should be familiar with the following topics:

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

Required Hardware/Software

Learners must have access to:

- 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)

*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.

Course 1

React Fundamentals

Mastering React begins with learning the 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. This course breaks everything down and enables learners with the foundational React skills that are necessary to build production-ready apps.

Course Project

MyReads: A Book Tracking App

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

Lesson 1

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 2

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 3

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 4

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 5

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, learners will begin to understand how React and Redux work together to make an application's state bulletproof. The React & Redux course offers hands-on learning to build projects. Here, learners will also gain knowledge on how to use advanced testing techniques using JEST.

Course Project

Employee Polls Web App

Learners 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.

Lesson 1

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 2

UI + Redux

- Combine Redux with a user interface.
- Build intuition for when to use Redux.

Lesson 3

Redux Middleware

- Identify the benefits of middleware within the Redux cycle.
- Apply middleware to a Redux application.
- Build your own Redux middleware.

Lesson 4

Redux with React

- Combine Redux with the popular React library.
- Identify when to use component state vs. Redux state.

Lesson 5

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 6

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 7

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 8

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. Learners will explore everything from setting up a proper development environment, building and styling a cross-platform mobile application. They'll incorporate native APIs such as geolocation and local notifications, and even learn how to get the app ready for the App and Google Play Store.

Course Project

Build a Customer Relationship Management App (Optional)

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

Lesson 1

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 2

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 3

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 4

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 5

Native Features

- Utilize AsyncStorage to persist global application data.
- Incorporate geolocation, animations, notifications, and ImagePicker to take advantage of device features and data.

Meet your instructors.

Andrew Wong

Course Developer

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

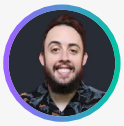
Senior Software Engineer

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

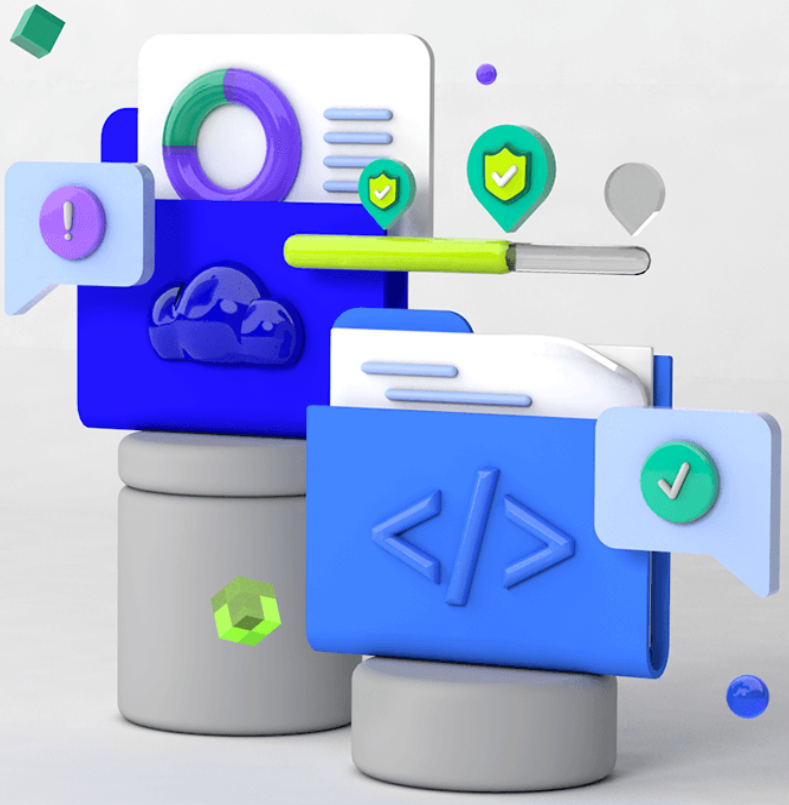
Tyler found his love for teaching at DevMountain, where he was lead instructor and curriculum engineer. He's a Google Developer Expert and is entrenched in the React community organizing React Utah and running React Newsletter.

Richard Kalehoff

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.

Quizzes

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

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.

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.

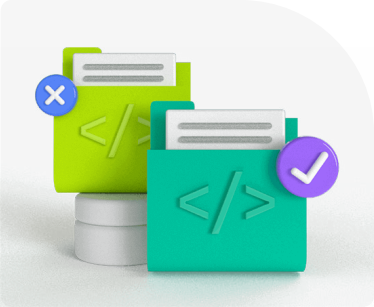
Workspaces

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.

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

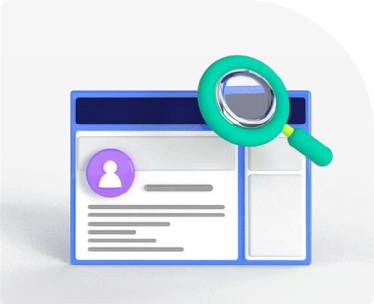
Verify skills mastery.

- 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.

UDACITY

Learn more at

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

12.12.22 | V1.0