UDACITY
FOR ENTERPRISE

SCHOOL OF PROGRAMMING AND DEVELOPMENT

React

NANODEGREE SYLLABUS

Overview

This goal of this Nanodegree Program is to teach you how to build declarative user interfaces for the web with React, and for iOS and Android with React Native. You'll also learn how to manage state more predictably in your applications with Redux.

A graduate of this program will be able to:

- Create a React application from scratch and utilize React components to manage the user interface
- Manage state in applications
- Conceptualize the lifecycle of a component
- Understand how Redux and React work together
- Build a complex, real-world application with Tyler
- Use React Native to build a mobile flashcard app
- Identify fundamental differences between web and native apps
- Identify differences between Android and iOS platforms
- Style applications with CSS in JS

This program is comprised of 3 courses and 3 projects. Each project you build will be an opportunity to demonstrate what you've learned in the lessons. Your completed projects will become part of a career portfolio that will demonstrate your acquired skills in React.

Program Information

ESTIMATED TIME TO COMPLETE

4 months; study 10 hrs/week

LEVEL

Practitioner

PREREQUISITES

- Prior development experience building and deploying front-end applications with HTML, CSS, JavaScript, Git, GitHub, and NPM
- Experience using the command line interface (bash, terminal)

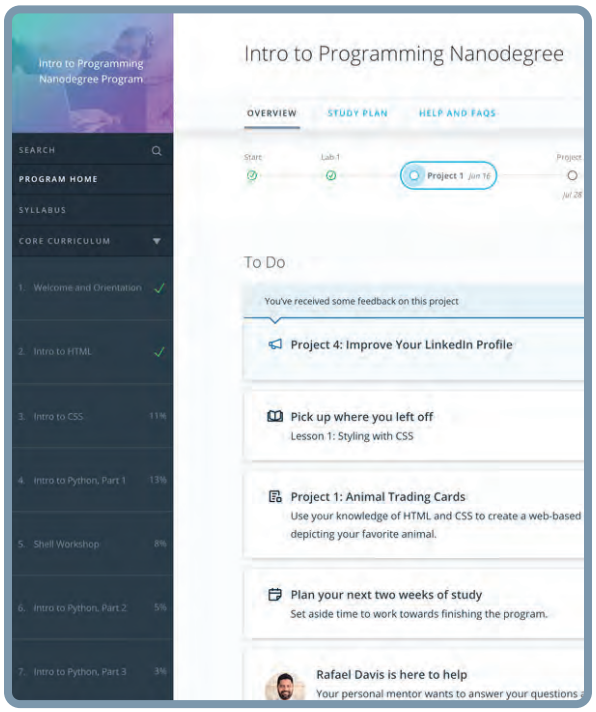
HARDWARE/SOFTWARE REQUIRED

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.)

LEARN MORE ABOUT THIS NANODEGREE

Contact us at enterpriseNDs@udacity.com.

Our Classroom Experience



REAL-WORLD PROJECTS

Learners build new skills through industry-relevant projects and receive personalized feedback from our network of 900+ project reviewers. Our simple user interface makes it easy to submit projects as often as needed and receive unlimited feedback.

KNOWLEDGE

Answers to most questions can be found with Knowledge, our proprietary wiki. Learners can search questions asked by others and discover in real-time how to solve challenges.

LEARNER HUB

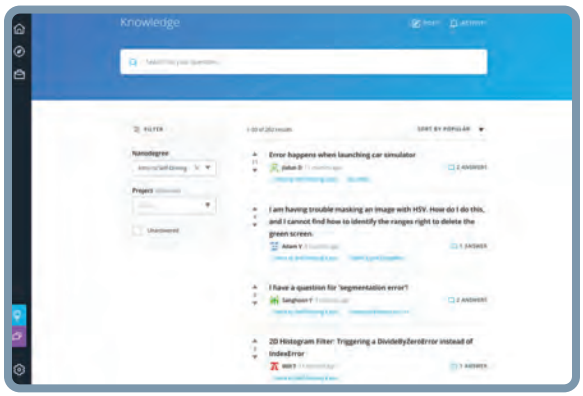
Learners leverage the power of community through a simple, yet powerful chat interface built within the classroom. Learner Hub connects learners with their technical mentor and fellow learners.

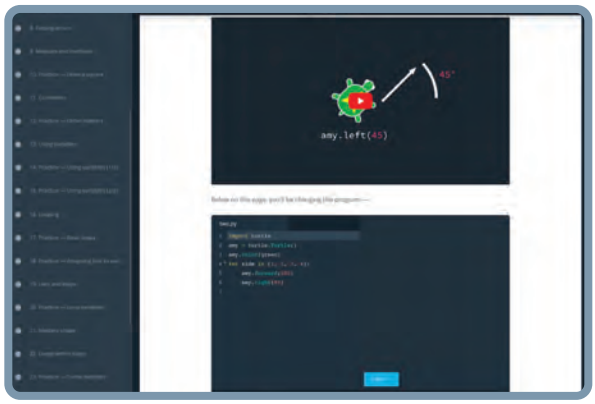
WORKSPACES

Learners can check the output and quality of their code by testing it on interactive workspaces that are integrated into the classroom.

QUIZZES

Understanding concepts learned during lessons is made simple with auto-graded quizzes. Learners can easily go back and brush up on concepts at anytime during the course.





CUSTOM STUDY PLANS

Mentors create a custom study plan tailored to learners' needs. This plan keeps track of progress toward learner goals.

PROGRESS TRACKER

Personalized milestone reminders help learners stay on track and focused as they work to complete their Nanodegree program.

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.

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 backend programming, to database and server management.

Nanodegree Program Overview

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.

As this is a project-based course, you're going to start building right away. This gives you an opportunity to get your hands dirty with React, and start mastering the skills you'll need. Plus, every project you build is reviewed by an expert Project Reviewer, and their detailed feedback will be instrumental in helping you to advance.

Project 1 — MyReads: A Book Lending 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's setState to build the functionality to move books from one shelf to another.

LESSON TITLE	LEARNING OUTCOME
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 • Recognize that React is just JavaScript
RENDERING UI WITH REACT	• se create-react-app to create a new React application • Create reusable, focused Class components with composition • Leverage JSX to describe U
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 • Use the this keyword to access component data and properties • Update state with setState() • Use PropTypes to typecheck and debug components • Use controlled components to manage input form elements

LESSON TITLE	LEARNING OUTCOME
RENDER UI WITH EXTERNAL DATA	• Conceptualize the lifecycle of a component • Use React's componentDidMount lifecycle hook for HTTP requests
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 Redux and React work together to make your application's state bulletproof.

As with the previous course, this is hand-on curriculum, and building projects is what it's all about. Here, you'll leverage React with Redux to build "Would You Rather", a popular party game.

Project 2 — Would You Rather

Leverage the strengths of Redux to build a "Would You Rather" application in which users are given questions and must choose one of them. You'll build this dynamic application from scratch while combining the state management features of Redux and the component model of React. When complete, you'll be able to create your own sets of questions, choose between them, and keep track of question popularity.

LESSON TITLE	LEARNING OUTCOME
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

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOME
UI + REDUX	• Combine Redux with a user interface • Build intuition for when to use Redux
REDUX MIDDLEWARE	• Identify the benefits of implementing middleware in applications • Identify the role of middleware within the Redux cycle • Apply middleware to a Redux application • Build your own Redux middleware
REDUX WITH REACT	• Combine Redux with the popular React library • Identify when to use component state vs. Redux state
ASYNCHRONOUS REDUX	• Learn the pitfall of asynchronous requests in Redux • Leverage Thunk middleware to support asynchronous requests • Fetch data from a remote API
REACT REDUX	• Install the react-redux bindings • Leverage react-redux bindings to extend app functionality • Use the Provider to pass a store to component trees • Use connect() to access store context set by the Provider
REAL WORLD REDUX	• Build a complex, real-world application with Tyler • Add Redux to an application scaffolded with Create React App • Normalize state shape to keep application logic simple with scale

Course 3: React Native

In this course, you'll learn how to to develop React applications that run on both iOS and Android devices. We'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 Google Play Store and the App Store!

Project 3 — Mobile Flashcards

In this project, you'll use React Native to build a mobile flashcard app. Users will not only be able to create custom cards and decks, but they'll also be able to set up notifications to remind them to study. You'll leverage React Native components, AsyncStorage, proper styling, as well as device APIs to create a fully dynamic experience.

LESSON TITLE	LEARNING OUTCOME
UP AND RUNNING WITH REACT NATIVE	• Identify the ideology behind React Native • Set up an ideal development environment • Inspect and debug applications
REACT VS. REACT NATIVE	• Identify fundamental differences between web and native apps • Identify differences between Android and iOS platforms • Leverage common React Native components • Create forms in React Native applications • Utilize AsyncStorage to persist global application data • ncorporate Redux to manage shared application state
STYLING & LAYOUT	• Style applications with CSS in JS • Identify differences and use-cases between styling with inline styles, object variables, and the Stylesheet API • Recognize the core philosophies and techniques of CSS flexbox • Identify key differences between flexbox on the web and React Native's implementation of flexbox • Identify best practices in how professionals handle styling

Nanodegree Program Overview

LESSON TITLE	LEARNING OUTCOME
NAVIGATION	• Manage navigation through a React Native application • Utilize StackNavigator to render screens from a stack • Implement TabNavigator to switch between screens by using tabs • Utilize DrawerNavigator to switch between screens from a drawer menu
NATIVE FEATURES	• Leverage native APIs to extend app functionality • Incorporate Geolocation, Animations, Notifications, and ImagePicker to take advantage of device features and data • Prepare applications for the Google Play Store and the App Store

Our Nanodegree Programs Include:

Pre-Assessments

Our in-depth workforce assessments identify your team's current level of knowledge in key areas. Results are used to generate custom learning paths designed to equip your workforce with the most applicable skill sets.

Dashboard & Progress Reports

Our interactive dashboard (enterprise management console) allows administrators to manage employee onboarding, track course progress, perform bulk enrollments and more.

Industry Validation & Reviews

Learners' progress and subject knowledge is tested and validated by industry experts and leaders from our advisory board. These in-depth reviews ensure your teams have achieved competency.

Real World Hands-on Projects

Through a series of rigorous, real-world projects, your employees learn and apply new techniques, analyze results, and produce actionable insights. Project portfolios demonstrate learners' growing proficiency and subject mastery.

Our Review Process

Real-life Reviewers for Real-life Projects

Real-world projects are at the core of our Nanodegree programs because hands-on learning is the best way to master a new skill. Receiving relevant feedback from an industry expert is a critical part of that learning process, and infinitely more useful than that from peers or automated grading systems. Udacity has a network of over 900 experienced project reviewers who provide personalized and timely feedback to help all learners succeed.

Vaibhav
UDACITY LEARNER

"I never felt overwhelmed while pursuing the Nanodegree program due to the valuable support of the reviewers, and now I am more confident in converting my ideas to reality."

now at
CODING VISIONS INFOTECH

All Learners Benefit From:

- Line-by-line feedback for coding projects
- Industry tips and best practices
- Advice on additional resources to research
- Unlimited submissions and feedback loops

How it Works

Real-world projects are integrated within the classroom experience, making for a seamless review process flow.

- Go through the lessons and work on the projects that follow
- Get help from your technical mentor, if needed
- Submit your project work
- Receive personalized feedback from the reviewer
- If the submission is not satisfactory, resubmit your project
- Continue submitting and receiving feedback from the reviewer until you successfully complete your project

About our Project Reviewers

Our expert project reviewers are evaluated against the highest standards and graded based on learners' progress. **Here's how they measure up to ensure your success.**

900+

Expert Project Reviewers

Are hand-picked to provide detailed feedback on your project submissions.

1.8M

Projects Reviewed

Our reviewers have extensive experience in guiding learners through their course projects.

3

Hours Average Turnaround

You can resubmit your project on the same day for additional feedback.

4.85 /5

Average Reviewer Rating

Our learners love the quality of the feedback they receive from our experienced reviewers.

UDACITY
FOR ENTERPRISE

2440 W El Camino Real, #101
Mountain View, CA 94040, USA - HQ

For more information visit: **www.udacity.com/enterprise**