UDACITY

NANODEGREE PROGRAM SYLLABUS

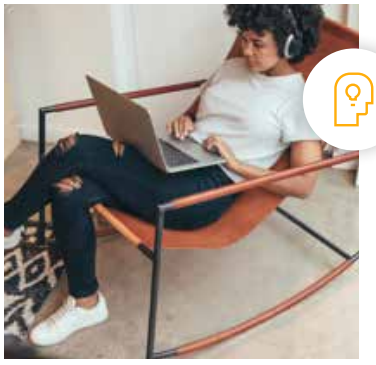
React Developer

Overview

Congratulations on considering the React Nanodegree program! Before you get started, make sure to set aside adequate time on your calendar for focused work, and double-check that you meet the requirements: you should have prior programming experience that includes HTML, CSS, and JavaScript programming.
The React Nanodegree program is comprised of 3 courses and 3 projects. Each project you build will be an opportunity to demonstrate what you've learned in your lessons. Your completed projects become part of a career portfolio that will demonstrate your mastery of React to potential employers.

Estimated Time:
4 Months at
10hrs/week

Prerequisites:
HTML, JavaScript,
& Git



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

Need Help?
udacity.com/advisor
Discuss this program
with an enrollment
advisor.

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.

Course Project
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.

		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 • Recognize that React is just JavaScript
LESSON TWO	**Rendering UI with React**	• Use create-react-app to create a new React application • Create reusable, focused Class components with composition • Leverage JSX to describe U

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 • 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 FOUR	**Render UI with External Data**	• Conceptualize the lifecycle of a component • Use React's componentDidMount lifecycle hook for HTTP requests
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 Redux and React work together to make your application's state bulletproof.

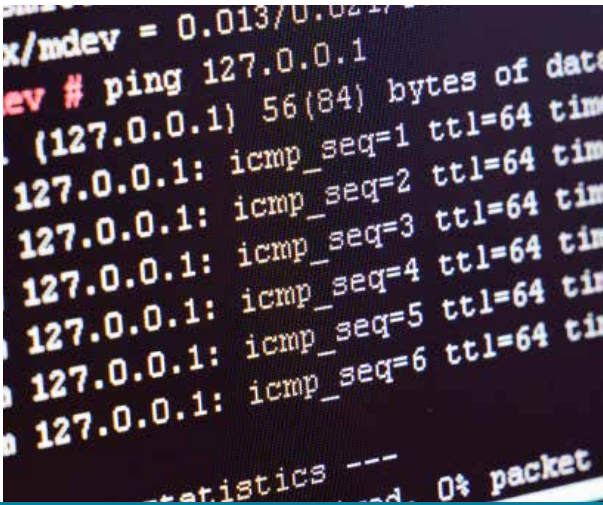
As with the previous course, this is hand-on curriculum, and building projects is what it's all about. Here, you'll leverage React with Redux to build "Would You Rather", a popular party game.

Course Project
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.

		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**	• Combine Redux with a user interface • Build intuition for when to use Redux
LESSON THREE	**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

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**	• Learn the pitfall of asynchronous requests in Redux • Leverage Thunk middleware to support asynchronous requests • Fetch data from a remote API
LESSON SIX	**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
LESSON SEVEN	**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!

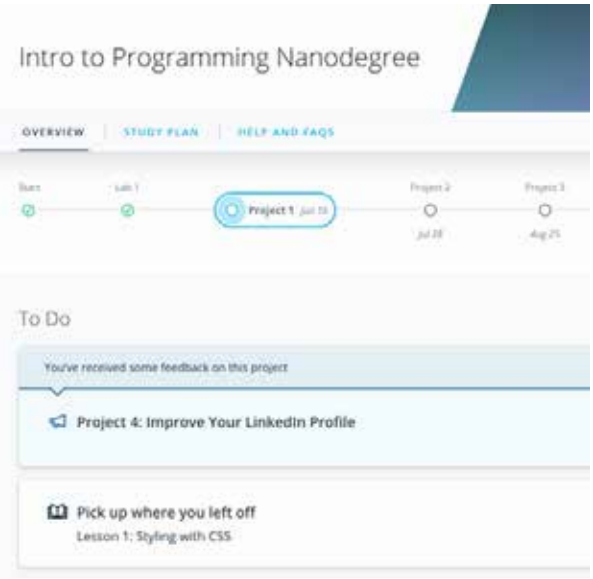
Course Project
Mobile Flashcards

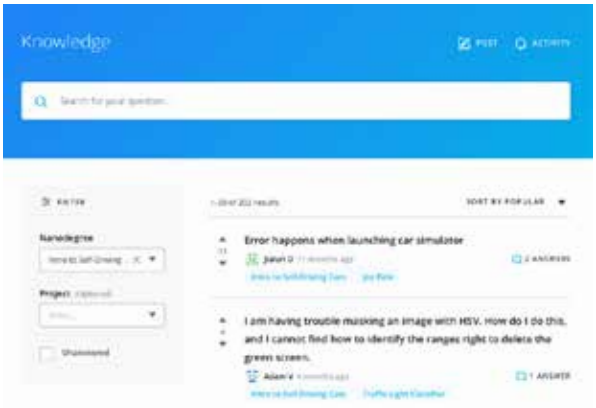
In this project, you'll use React Native to build a mobile flashcard app. Users will not only be able to create custom cards and decks, but they'll also be able to set up notifications to remind them to study. You'll leverage React Native components, AsyncStorage, proper styling, as well as device APIs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

		LEARNING OUTCOMES
LESSON ONE	**Up and Running with React Native**	• Identify the ideology behind React Native • Set up an ideal development environment • Inspect and debug applications
LESSON TWO	**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 • Incorporate Redux to manage shared application state

LESSON THREE	**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
LESSON FOUR	**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
LESSON FIVE	**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 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.

STUDENT HUB

Leverage the power of community through a simple, yet powerful chat interface built within the classroom. Use Student Hub to connect with fellow students in your program as you support and learn from each other.

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

Preschedule your study times and save them to your personal calendar to create a custom study plan. Program regular reminders to keep track of your progress toward your goals and completion of your program.

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.

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.

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

- Resume support
- Github portfolio review
- LinkedIn profile optimization

Frequently Asked Questions

PROGRAM OVERVIEW

WHY SHOULD I ENROLL?

Learning React through this Nanodegree program can significantly improve your skills and career prospects as a front-end developer, and Udacity believes it's one of the best career moves you can make right now. Udacity has partnered with React expert Tyler McGinnis to bring you this world-class learning experience—quality React instruction with a leading expert in the field, detailed code reviews, and support throughout the Nanodegree program.

In our React Nanodegree program, you will:

- Learn React, Redux, and React Native to build performant, interactive, and data-driven applications.
- Practice and refine your newly acquired skills by building 3 hands-on portfolio projects.
- Benefit from receiving detailed, personalized project feedback from experts in the field.

WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

Graduates of this Nanodegree program will be valuable additions to any team working in the domain of web development, app development, software development, digital marketing, and e-commerce. Opportunities exist in companies ranging from Fortune 500 organizations to startups.

Specific roles include: Front-End Web Developers, Full Stack Web Developers, and UI/UX Developers. For salary information, please visit the salary module on the React Nanodegree Program home page. You can also find industry insights on React in the Stack Overflow 2017 Developer Survey Results.

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

We designed our React Nanodegree program with one priority—your success as a developer. Whether you're pursuing a new role, advancing further in your existing career, or refreshing your skills and staying up to date with the latest technologies, this program is built to ensure you achieve your goals. The addition of React skills to your developer toolkit is an excellent move for any developer seeking a critical career advantage.

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.

FAQs Continued

WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

Students should have prior development experience building and deploying front-end applications with HTML, CSS, JavaScript, Git, GitHub, NPM, and experience using the command line interface (bash, terminal).

Students will need to be able to communicate fluently and professionally in written and spoken English.

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 three (3) projects. We estimate that students can complete the program in four (4) months, working 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.

FAQs Continued

SOFTWARE AND HARDWARE

WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

For this Nanodegree program, you will need 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.)

WHICH VERSION OF REACT IS TAUGHT IN THIS PROGRAM?

The React Nanodegree program teaches version 15.5+ of the React Library.