

# React Essentials

## Corporate Training Syllabus

---

## Course Overview

If you have some programming experience and want to learn how to build responsive web applications using React, this course is for you or your team. In this 3.5-day course, students craft responsive websites and tame complexity through maintainable user interface components. Students get a very brief refresher on CSS, HTML and Javascript as they dive into the React framework. Students then learn the ins and outs of the React framework so that they can build solid, responsive React applications.

### Who Should Take This Course

- Native developers who want to create a web presence that supports an app, grows a user base and collects user feedback.
- Backend developers who need to create full-stack web apps.
- Web developers transitioning to ReactJS.

### Prerequisites

- A basic understanding of object-oriented or functional programming is expected.
- Software development experience is necessary, though it does not have to be with JavaScript.
- If you have no prior HTML and CSS experience, study Codecademy's [Intro to HTML](#) and [Intro to CSS](#).
- Before class starts, make sure you have the right tools installed. You will need a browser, an IDE, and Node 14. We recommend Chrome for your browser and VS Code as your IDE.
- In terms of Operating System, we recommend Mac, Linux, or Windows System for Linux (WSL) but using Windows without WSL is possible.

# Syllabus

## Building Responsive, Mobile-Friendly Websites

- Style and lay out an application with HTML and modern CSS.
- Create a responsive web page with CSS Grid Layout, Flexbox, and Media Queries.
- Implement visual effects and animation using CSS3 transitions.
- Write JavaScript functions to respond to user interactions with the page.
- Use HTML5 form elements to create an order form.
- Ensure correct input with client-side validation.

## Creating Web Applications With React

- Split your interfaces into easily-managed, reusable components using the React framework.
- Utilize JSX to describe how the interfaces should be rendered.
- Persist and synchronize state between components using React hooks.
- Manage cart contents with a state reducer.
- Access, create, and delete remote server data using AJAX and REST.
- Asynchronous data fetching with React hooks and promises.
- Manage routing with React Router.
- Creating orders and login with a backend API.
- Sync open orders with websockets.
- Pass user details throughout the application with React Context.

## Testing React

- Ensure your website behaves as expected through tests
- Write unit tests with React Testing Library
- Write end to end tests with Cypress