

2024 EDITION REACT SPA DEVELOPMENT WITH V18.3

MASTERING REACT

PART I

A FIVE-PART SERIES FROM BEGINNER
TO ADVANCED

BY BRAUN GOODSON

MASTERING REACT

A FIVE-PART SERIES FROM BEGINNER
TO ADVANCED

2024 EDITION REACT SPA DEVELOPMENT WITH V18.3

BY BRAUN GOODSON

About this book

Throughout the five-part series, you'll be building a **Personal Task Manager**—a to-do list application

This series will guide you through the essentials of building a React application, starting from the basics and progressing to advanced topics. By the end of the series, you'll have a solid understanding of React and single-page app development, along with a deployable project to showcase your skills.

Part I: Introduction to React and Single-Page Applications

What is React?	1
Overview of React and its benefits	
The role of React in modern web development	
Understanding Single-Page Applications (SPAs)	2
Difference between SPAs and traditional web apps	
Advantages of using SPAs	
Setting Up the Development Environment	3
Installing Node.js and npm	
Using Create React App to bootstrap your project	
Preview of the Project Series	4
Brief overview of the application you'll build throughout	

1

What is React?

Overview of React and Its Benefits

By definition React is an open-source JavaScript library for building user interfaces, developed by Facebook. Its Component-Based Architecture allows you to build encapsulated components that manage their own state. Its Virtual DOM efficiently updates and renders just the right components when your data changes.

Some of the benefits of using react are:

- *Reusable Components*: Write code once and reuse it throughout your app.
- *Performance*: Faster rendering with the Virtual DOM.
- *Developer Tools*: Strong ecosystem with tools like React Developer Tools.

The Role of React in Modern Web Development

React's popularity is seen widely adopted in the industry, used by companies like Facebook, Instagram, and Netflix. Its ecosystem has extensive community support, libraries, and tutorials. React's versatility enables use for web, mobile (React Native), and even desktop applications.

2

Understanding Single-Page Applications (SPAs)

Difference Between SPAs and Traditional Web Apps

With Traditional Web Apps every new page request loads a new HTML page from the server. Page reloads can be slow and disrupt the user experience.

Whereas Single-Page Applications load a single HTML page and dynamically updates content as the user interacts. It uses AJAX and HTML5 to create fluid and responsive web apps.

Advantages of Using SPAs

At the top, is performance and faster load times after the initial load. Next is User Experience has smooth transitions without full page reloads. The Development Efficiency enables code reuse and easier debugging. They have Offline Capabilities meaning better support for offline scenarios using service workers.

3

Setting Up the Development Environment

Installing Node.js and npm

Download Node.js by visiting [Node.js website](#) and download the LTS version. Install Node.js by running the installer and following the prompts. Verify installation by opening your terminal and typing:

```
node -v && npm -v
```

to check the installed versions.

Using Create React App to Bootstrap Your Project

Install Create React App globally by running:

```
npm install -g create-react-app
```

Create a new React App by navigating to your desired directory and running:

```
npx create-react-app task-manager-app
```

Then navigate into the new `task-manager-app` directory. Start the development server by running:

```
npm start
```

Finally, open <http://localhost:3000> in your browser to view the app.

4

Preview of the Project Series

Brief Overview of the Application You'll Build

Build a personal Task Manager App. You can add new tasks by entering task detail and add them to your list. View Task List to see all your tasks displayed neatly. Manage Tasks as complete or delete them. Navigate Between Views using routing to switch between different pages like 'All Tasks' and 'Completed Tasks'.

What You'll Learn

React Fundamentals such as Components, props, and JSX. State Management using hooks like `useState` and `useEffect`. Advanced Concepts with Context API, and routing with React Router. Performance Optimization by code splitting, and memoization. Finally Deployment to deploy your app so the world can see it.

Closing Remarks

That's it for this introductory book for Part I in the series Mastering React! You've set up your development environment and have a roadmap of what we'll achieve together. In the next book Part II, we'll start diving into React fundamentals by building components and understanding JSX. Don't forget to like, share, and subscribe so you don't miss out on the next part of the series. See you soon!

Additional Resources:

- *React Documentation:* reactjs.org
- *Node.js Downloads:* nodejs.org
- *Code Editor Suggestions:*
 - i. [Visual Studio Code](#)
 - ii. [Atom](#)
 - iii. [Sublime Text](#)

Call to Action:

- *Subscribe:* Stay updated with the latest books in this series.
- *Comments:* If you have any questions or need clarifications, drop a comment Braun's e-mail `brn.gdsn@gmail.com`.
- *Share:* If you found this book helpful, share it with others who might benefit.

The End

Thank you for Reading! Up next: "React Fundamentals – Components, Props, and JSX".

MASTERING REACT

A FIVE-PART SERIES FROM BEGINNER TO ADVANCED

Throughout the five-part series, you'll be building a Personal Task Manager—a to-do list application

This series will guide you through the essentials of building a React application, starting from the basics and progressing to advanced topics.

By the end of the series, you'll have a solid understanding of React and single-page app development, along with a deployable project to showcase your skills.

Why This App?

- Beginner-Friendly: It's simple enough for newcomers to grasp basic concepts.
- Core React Features: Covers essential topics like components, props, state, and hooks.
- Advanced Concepts: Provides a foundation to introduce context API, routing, and performance optimization.
- Practical Use: Creates a functional tool that viewers can use and expand upon after the series.

GITHUB REPOSITORY

