



SEPTEMBER 2025

# Comprehensive Study Guideline on REACT.JS

WHITE MATRIX

PREPARED BY:  
WHITE MATRIX  
TRAINING TEAM

👋 Hey folks,

### why are we even learning React?

Imagine building a normal website with HTML, CSS, and JavaScript. Every time you want to update something (like a counter or a list), you have to rewrite chunks of code. Painful, right? 😞

### React changes the game!

- It lets us build apps in small, reusable pieces (components).
- It only updates what changes (super fast ⚡).
- It makes complex apps (like Instagram, Netflix, WhatsApp Web) easier to build and maintain.

### Before Learning React

To get started, you should already be comfortable with:

- HTML → the structure of a webpage (tags, forms, links, images).
- CSS → styling (colours, layouts, flexbox, grid).
- JavaScript (ES6) → logic (variables, functions, arrays, loops, objects, arrow functions).

### 👉 Mini Project:

Build a Personal Card Website using HTML, CSS, and a little JavaScript.

- Show your name, photo, and skills.
- Add a button that pops up an alert → "Hello, I'm [Your Name]!"

# 01 Setting up React.js

Alright, let's get our hands dirty 💻

1. Install Node.js from [nodejs.org](https://nodejs.org).
2. Create your first React app:

```
bash
```

```
npx create-react-app my-app  
cd my-app  
npm start
```

### 3. Boom 🌟!

You'll see a React app running at `http://localhost:3000`.

#### 👉 Mini Project:

Change the homepage text to → `Hello, my name is [Your Name] and I'm learning React!`

## 02 Components & JSX

Think of **components** like Lego blocks 🧱. You build small pieces and plug them together.

JSX lets us write **HTML inside JavaScript**.

#### 👉 Mini Project:

- Create three components: `Header` , `Content` , and `Footer` .
- Import and use them inside `App.js` .
- Congrats! You just built your first "multi-block" React app 🎉

## 03 Props (Passing Data Around)

Props are like arguments for components. You pass them in, and the component shows them.

#### 👉 Mini Project:

- Build a `StudentCard` component that takes `name` , `course` , and `year` as props.
- Render a list of 5 students.
- Example: "Name: Rahul | Course: CSE | Year: 3rd"

# 04 State & Events

Props are fixed... but what if we want something to change?  
That's where state comes in → like a memory inside your component.

## 👉 Mini Project:

- **+** button → increases count
- **-** button → decreases count
- **Reset** button → sets count to 0

# 05 Lists Conditional Rendering

Need to show many items? Use **.map()**.

Need to show/hide stuff? Use conditions.

## 👉 Mini Project:

Create a To-Do List App:

- Add tasks
- Show tasks in a list
- Mark tasks as complete (strike-through).

# 06 useEffect & API Calls

Now let's connect to the real world

`useEffect` lets you run code when the component loads.

## 👉 Mini Project:

Build a Random Joke App:

- Fetch jokes from `https://official-joke-api.appspot.com/random\_joke`
- Show a new joke every time the page reloads.

# 07 Splitting Apps into Multiple Components

Big apps are messy → split them up into smaller pieces.

## 👉 Mini Project:

Refactor your To-Do App into:

- `InputBox` (adds tasks)
- `TaskList` (shows tasks)
- `TaskItem` (one task).

# 08 Routing (Multiple Pages)

Most websites need more than one page → Home, About, Contact.  
Use React Router.

## 👉 Mini Project:

Create a 2-page React website with navigation links:

- Home page → "Welcome!"
- About page → "About Me"

# 09 Styling React Apps

React doesn't care how you style → you pick!

- CSS files → classic way
- Inline CSS → quick fixes
- Tailwind CSS → fast modern styling

## 👉 Mini Project:

Style your To-Do App with Tailwind. Add:

- Colors
- Hover effects
- Dark mode switch

# 10 Advanced Stuff

Want to go pro? Learn:

- Context API → share data across components.
- useReducer → handle complex states.
- useMemo / useCallback → make apps faster.

## 👉 Mini Project:

- Add a Dark/Light Mode Toggle to your app using Context API.

# 11 Deployment

Show off your work! 🚀

- Run `npm run build.`
- Deploy on **Netlify** or **Vercel** → free and instant.

## 👉 Mini Project:

Deploy your To-Do App → share the live link with friends.

# Final Practice Projects (Portfolio-Ready)

# Final Practice Projects (Portfolio-Ready)

Once you finish, try these:

1. Weather App (fetch weather API).
2. Movie Search App (OMDB API + search).
3. Notes App (save to Local Storage).
4. Calculator.
5. Personal Portfolio Website (show your skills + projects).

## 👉 Mini Project:

- Add a Dark/Light Mode Toggle to your app using Context API.

## Recommended Resources

Official Docs: [React Docs](#) (Beginner-friendly, well-structured)

YouTube Crash Course: [FreeCodeCamp React Crash Course \(2 hrs\)](#)

Project-Based Learning: [Traversy Media – React Crash Course](#)

Deployment Guides: [Netlify](#) / [Vercel](#)

“**Next**, what you gotta learn is **Next.js** — it’s built on top of React and gives you extra advantages like faster loading, built-in routing, API support, and overall makes your apps production-ready.”