ET THE ECONOMIC TIMES

Introduction Rohit Azad

Hello everyone, my name is **Rohit Azad**, and I am a UI developer with over **13 years of experience** in **frontend development**. During my career, I have worked on a wide range of **web applications**, from small startups to large enterprise systems, which has given me a deep understanding of the challenges faced by developers when building modern web applications.

ET THE ECONOMIC TIMES

Course Intro - Day - 1

1. Intro React
2. What is React
3. React Components
4. Export a Default and Named Component
5. React Dom Element

ET THE ECONOMIC TIMES

Course Intro - Day - 2

1. React Hooks
2. State and Props
3. LifeCycle Method
4. React Fragment
5. Code Split lazy loading
6. Context API
7. Next level of React with Next js

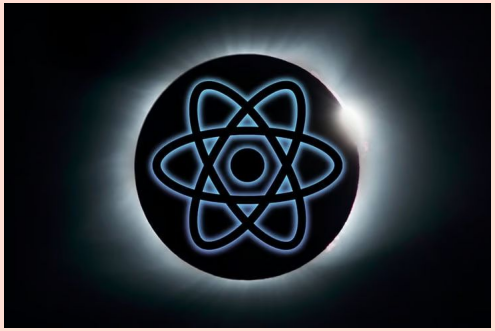
This training aims to provide developers with a practical introduction to **React JS**, so they can build **scalable**, **maintainable**, and **high-performing** web **applications**.

The training will consist of several modules that cover key aspects of React JS development, such as **components**, **props**, **state**, **forms**, **API calls**, and best practices.

By the end of the training , developers will have gained a solid understanding of **React JS**, and will be able to use it to **build modern**, **dynamic** web applications. They will also have the **skills** and **knowledge** to deploy and maintain their applications using best practices.

ET THE ECONOMIC TIMES

React JS is, how it works, and why it is important to learn.



Thank you for joining me today to learn about **React JS**. In today's rapidly evolving technology landscape, it's important to stay up-to-date with the latest tools and technologies. **React JS** is one such technology that is transforming the way we build web applications.

React JS is a powerful and flexible **JavaScript** library for building user interfaces.
It was developed by Facebook and is now used by many of the world's top companies to create fast, responsive, and scalable web applications. **React JS** allows developers to build complex user interfaces by breaking them down into smaller, reusable components. This approach makes it easier to maintain and update large codebases, and helps developers create better user experiences.

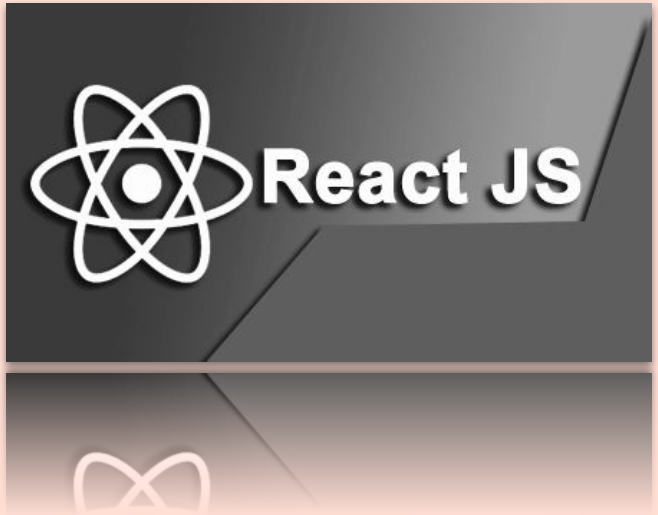
ET THE ECONOMIC TIMES

Outline the benefits of React JS

1. **Better performance**: React JS is designed to optimize the performance of web applications. By using virtual DOM, React JS can quickly and efficiently update the user interface, resulting in a faster and smoother experience for users.
2. **Reusability**: React JS makes it easy to reuse code components across different parts of an application, which can save a significant amount of time and effort when building large-scale applications.
3. **Scalability**: React JS is designed to work well with large codebases, making it a popular choice for building scalable web applications. As applications grow and become more complex, React JS allows developers to easily manage the codebase and make changes without affecting other parts of the application.
4. **Compatibility**: React JS is compatible with a wide range of web browsers, which makes it an excellent choice for building cross-platform web applications.
5. **Wide community support**: React JS has a large and active community of developers who contribute to the development of the library, create new tools and resources, and provide support to other developers. Learning React JS means employees can tap into this community and access a wealth of knowledge and resources.

ET THE ECONOMIC TIMES

React JS training program



1. **Duration**: The training program with 16 hours of training.

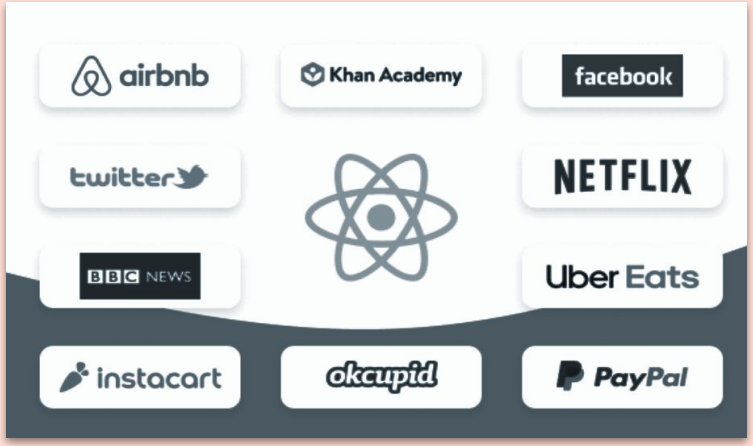
2. **Topics covered**: The program will cover the fundamentals of **React JS**, **including JSX**, **components**, **state**, and **props**. We will also cover more advanced topics such as **React Hooks**, **Context API**.

3. The React JS training program is designed to help employees gain proficiency in React JS and build the skills they need to create modern, efficient, and scalable web applications. By the end of the program, employees will have a solid understanding of React JS and will be able to confidently apply their knowledge to real-world applications.

ET THE ECONOMIC TIMES

Why You Should Use React for Web Development

React is a JavaScript library developed by Facebook which, among other things, was used to build Instagram.com. Its aim is to allow developers to easily create fast user interfaces for websites and applications alike. The main concept of React.js is virtual DOM. It is a tree based on JavaScript components created with React that mimics a DOM tree. It does the least amount of DOM manipulation possible in order to keep your React components up to date.

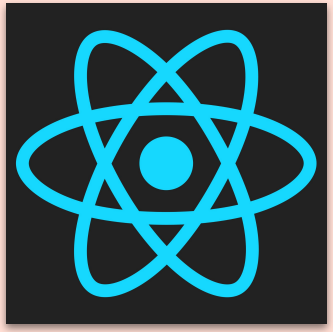


1. React is a JavaScript library created by Facebook
2. React is a User Interface (UI) library
3. React is a tool for building UI components

ET THE ECONOMIC TIMES

Declarative

React makes it painless to create interactive UIs. Design simple views for each state in your application, and React will efficiently update and render just the right components when your data changes. Declarative views make your code more predictable and easier to debug.

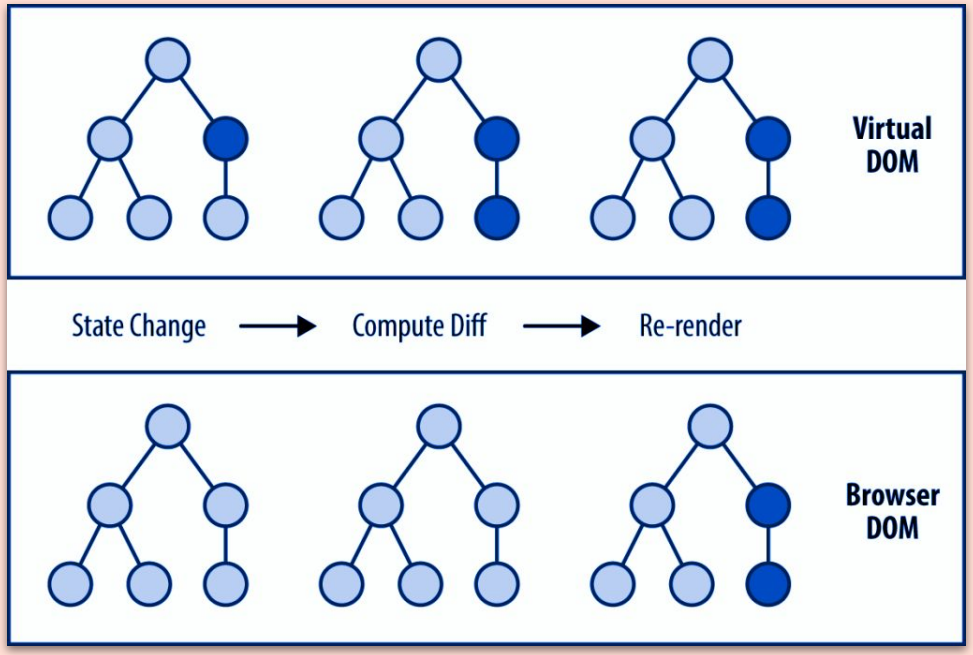


Component-Based

Build encapsulated components that manage their own state, then compose them to make complex UIs. Since component logic is written in JavaScript instead of templates, you can easily pass rich data through your app and keep state out of the DOM.

ET THE ECONOMIC TIMES

High performance reinforced by Virtual DOM



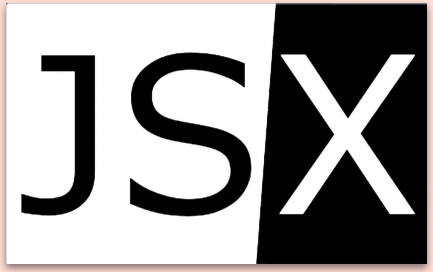
ET THE ECONOMIC TIMES

High performance reinforced by Virtual DOM

By virtualizing and keeping DOM in memory, React grants outstandingly fast rendering capacities with all view changes being readily reflected in the virtual DOM. The specialized diff algorithm juxtaposes former and existing virtual DOM states, calculating the most efficient way to apply new changes without requiring too many updates. A minimum number of updates is then introduced to achieve the fastest read/write time, which results in an overall performance boost. DOM changes make systems slow, and by virtualizing DOM, those changes are minimized and optimized in an intelligent way. All the virtual DOM manipulations happen “behind the scenes,” i.e., internally and autonomously, which also allows to significantly save hardware resource-consumption rates (CPU power and battery in mobile devices, for instance, which should give you a hint as to when to use React.js as well).

ET THE ECONOMIC TIMES

JSX syntax for extended HTML. JSX stands for JavaScript XML. JSX allows us to write HTML in React. JSX makes it easier to write and add HTML in React.



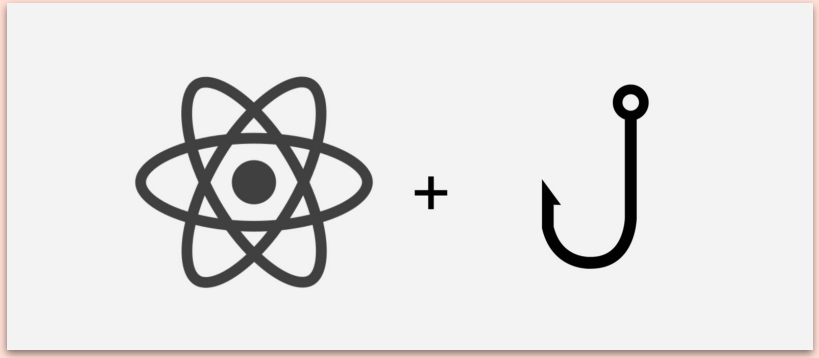
Why use React for web development? With React.js, you may utilize the declarative HTML syntax directly in JavaScript code. To show the user interface, browsers decode HTML texts. They do it by building DOM trees, which can then be manipulated using JavaScript to create interactive UI. Manipulation of DOMs by manifolds is more efficient using JSX. Developers may build tidy, maintainable code by passing HTML and React.js components into the browser's tree structures. React.js apps are faster and more efficient thanks to JSX and the Virtual DOM. Other frameworks and libraries can also be utilized with JSX. As you can see above, JSX is not JavaScript nor HTML. JSX is a XML syntax extension to JavaScript that also comes with the full power of ES6 (ECMAScript 2015). Just like HTML, JSX tags can have a tag names, attributes, and children. If an attribute is wrapped in curly braces, the value is a JavaScript expression.

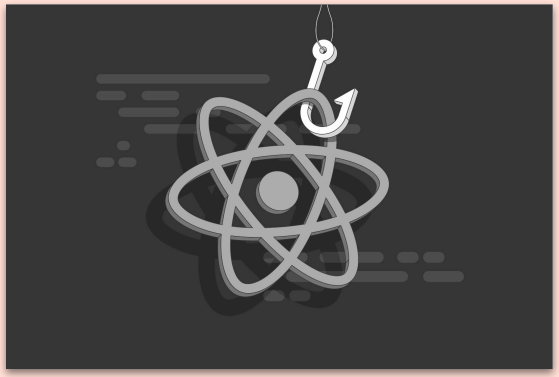
ET THE ECONOMIC TIMES

Unique React hooks

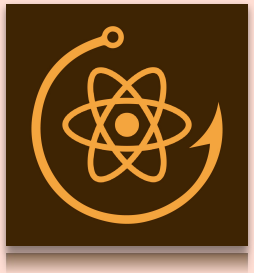
Hooks are a new addition in **React 16.8**. They let you **use state** and other React

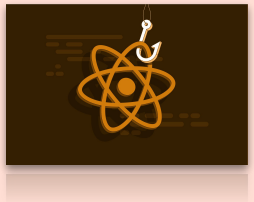
React **Hooks** are simple **JavaScript** functions that we can use to isolate the reusable part from a functional component. Hooks can be stateful and can manage side-effects.





Unique React hooks





React provides a bunch of standard in-built hooks:
useState: To manage states. Returns a stateful value and an updater function to update it.
useEffect: To manage side-effects like API calls, subscriptions, timers, mutations, and more.
useContext: To return the current value for a context.
useReducer: A useState alternative to help with complex state management.
useCallback: It returns a memorized version of a callback to help a child component not re-render unnecessarily.
useMemo: It returns a memoized value that helps in performance optimizations.
useRef: It returns a ref object with a .current property. The ref object is mutable. It is mainly used to access a child component imperatively.
useLayoutEffect: It fires at the end of all DOM mutations. It's best to use useEffect as much as possible over this one as the
useLayoutEffect fires synchronously.
useDebugValue: Helps to display a label in React DevTools for custom hooks.

ET THE ECONOMIC TIMES

Prerequisites

We'll assume that you have some familiarity with **HTML** and **JavaScript**, but you should be able to follow along even if you're coming from a different programming language.

We'll also assume that you're familiar with programming concepts like **functions**, **objects**, **arrays**, and to a lesser extent, **classes**. Note that we're also using some features from **ES6** — a recent version of **JavaScript**.

In this tutorial, we're using **arrow functions**, **classes**, **let**, and **const** statements.

ET THE ECONOMIC TIMES

Where To Get Support

There are many online forums which are a great place for discussion about best practices and application architecture as well as the future of React.

If you have an answerable code-level question, **Stack Overflow** is usually a better fit.

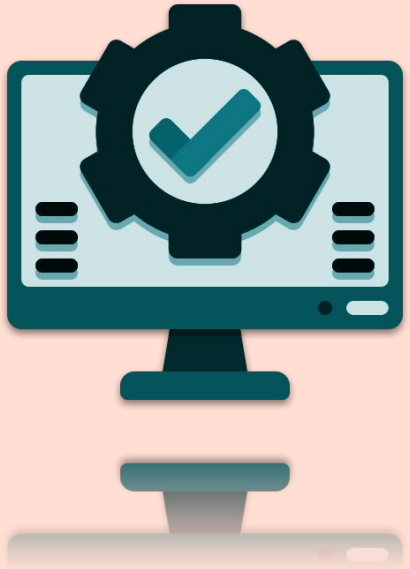
Each community consists of many thousands of React users.

- DEV's React community
- Hashnode's React community
- Reactiflux online chat
- Reddit's React community

ET THE ECONOMIC TIMES

Let’s Start The :- System Requirements

1. Laptop with Internet Connection
2. Node.js 14.6.0 or newer
3. MacOS, Windows (including WSL), and Linux are supported
4. visual studio code / Text editor
5. html css javascript knowledge



ET THE ECONOMIC TIMES

You can continue using DOM methods to create a new <h1> element:

```html
<!-- index.html -->
<html>
  <body>
    <div id="app"></div>
    <script type="text/javascript"></script>
  </body>
</html>
```

```javascript
// Select the div element with 'app' id
const app = document.getElementById('app');

// Create a new H1 element
const header = document.createElement('h1');

// Create a new text node for the H1 element
const headerContent = document.createTextNode(
  'Develop. Preview. Ship. 🚀',
);

// Append the text to the H1 element
header.appendChild(headerContent);

// Place the H1 element inside the div
app.appendChild(header);
```

ET THE ECONOMIC TIMES

HTML vs the DOM

If you look at the DOM elements inside your browser developer tools, you will notice the DOM includes the <h1> element. The DOM of the page is different from the source code - or in other words, the original HTML file you created.

ET THE ECONOMIC TIMES

Imperative vs Declarative Programming

The code above is a good example of imperative programming. You're writing the steps for how the user interface should be updated. But when it comes to building user interfaces, a declarative approach is often preferred because it can speed up the development process. Instead of having to write DOM methods, it would be helpful if developers were able to declare what they want to show (in this case, an h1 tag with some text).

In other words, imperative programming is like giving a chef step-by-step instructions on how to make a pizza. Declarative programming is like ordering a pizza without being concerned about the steps it takes to make the pizza.

ET THE ECONOMIC TIMES

React: A declarative UI library

As a developer, you can tell React what you want to happen to the user interface, and React will figure out the steps of how to update the DOM on your behalf.

```
<div id="app"></div>
<script src="https://unpkg.com/react@17/umd/react.development.js"></script>

<script
src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

<script type="text/javascript">
    const app = document.getElementById('app');
    ReactDOM.render(<h1>Develop. Preview. Ship. 🚀</h1>, app);
</script>
```

ET THE ECONOMIC TIMES

What is JSX?

JSX is a syntax extension for JavaScript that allows you to describe your UI in a familiar HTML-like syntax.
The nice thing about JSX is that apart from following three JSX rules, you don't need to learn any new symbols or syntax outside of HTML and JavaScript.

Note that browsers don't understand JSX out of the box, so you'll need a JavaScript compiler, such as a **Babel**, to transform your JSX code into regular JavaScript.

```
<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
```

ET THE ECONOMIC TIMES

Finel Code of React Js

```
<html>
    <body>
        <div id="app"></div>
  <script src="https://unpkg.com/react@17/umd/react.development.js"></script>
  <script src="https://unpkg.com/react-dom@17/umd/react-dom.development.js"></script>

<!-- Babel Script -->

<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
<script type="text/jsx">
       const app = document.getElementById('app');
        ReactDOM.render(<h1>Develop. Preview. Ship. 🚀</h1>, app); </script>
  </body>
</html>
```

ET THE ECONOMIC TIMES

React Js Components

```
<script type="text/jsx">
    function Header(props){
        return (
            <div>
                <h1>{props.data}</h1>
                <h2>Hello heading 2</h2>
                <button onClick={()=>{
                    props.changeTxt('Change Text Hello React to ET React Webstie')
                }}>Change Text head 1</button>
            </div>
        )
    }
    const PageComponents = ()=>{
        const [text,setText] = React.useState('Hello React Js 1')
        const changeTxt = (getText)=>{
            setText(getText)
        }
        return (
           <Header changeTxt={changeTxt} data={text} />
        )
    }
    const app = document.getElementById('root');
    ReactDOM.render(<PageComponents />, app)
```

ET THE ECONOMIC TIMES

React Js CLI (Create a New React App)

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

Connect with Us

"If you would like to connect with me or access more coding resources, you can find me on **Facebook** and **LinkedIn** or check out my **YouTube channel 'Learn Coding with Bhai'** for free coding tutorials and projects."

Email ID :- **rohit.azad@timesinternet.in**

Contact No :- **+91-9953878727**

Facebook :- **https://www.facebook.com/IamRohitAzad**

Linkedin :- **https://www.linkedin.com/in/rohitazad/**

Github :- **https://github.com/rohitazad**

Youtube :- **https://www.youtube.com/@azadMalikRohit**