

Introduction to React

React is a **JavaScript library** for building **user interfaces**, especially **single-page applications (SPAs)** where you can create fast, interactive user experiences using reusable components.

Lab Objectives

- Install and run a React project.
- Understand and structure React app folders.
- Create and call components.
- Style components using CSS.
- Use Bootstrap with React.
- Learn about React-specific attributes.

React Installation

1. **Install Node.js** (if not already installed):

Download from [Node.js Official Website](https://nodejs.org/en/).

Verify installation:

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

2. **Create a new React project**

Open the terminal/command prompt and run the following command to create a new React app using `create-react-app`:

```
npx create-react-app my-app
```

3. **Navigate to the project directory:**

After the command runs, navigate to the newly created app directory:

```
cd my-app
```

4. **Start the development server:**

```
npm start
```

React Folder Structure:

Default Folder Structure:

Upon creating a new React app, the `create-react-app` tool generates the following folder structure:

```
my-app/
├── node_modules/    # Contains all the npm dependencies
├── public/          # Static files like index.html
│   └── index.html   # Entry point for the React app
├── src/             # Application source code
│   ├── App.js       # Main app component
│   ├── App.css      # CSS for App component
│   └── index.js      # Entry point for React rendering
├── package.json     # NPM package configuration
└── README.md        # Project documentation
```

React Components Creation

Steps to Create a Functional Component:

Create a `Header.jsx` component:

```
// src/components/Header.jsx
import React from 'react';

function Header() {
  return <h1>Welcome to the React Lab</h1>;
}

export default Header;
```

Create a `Footer.jsx` component:

```
// src/components/Footer.jsx
import React from 'react';

function Footer() {
  return <footer>React Lab © 2025</footer>;
}

export default Footer;
```

React Components Calling:

Steps to Use Components in `App.js`:

```
import React from 'react';
import Header from './components/Header'; // Import Header component
import Footer from './components/Footer'; // Import Footer component

function App() {
  return (
    <div>
      <Header />
      <p>This is a simple React app demonstrating components.</p>
      <Footer />
    </div>
  );
}

export default App;
```

React Components CSS:

Approaches to Styling:

External CSS:

You can create a separate `.css` file and import it into your component.

Example: Create a `Header.css` file:

```
/* src/components/Header.css */  
.header {  
  color: white;  
  background-color: #007bff;  
  padding: 10px;  
  text-align: center;  
}
```

Then import and use it in `Header.jsx`:

```
import React from 'react';  
import './Header.css'; // Import the CSS file  
  
function Header() {  
  return <h1 className="header">Welcome to the React Lab</h1>;  
}  
  
export default Header;
```

Inline CSS:

Inline styling allows you to pass a style object directly into the JSX elements.

```
function Header() {  
  const headerStyle = {  
    color: 'white',  
    backgroundColor: '#007bff',  
    padding: '10px',  
    textAlign: 'center',  
  };  
  
  return <h1 style={headerStyle}>Welcome to the React Lab</h1>;  
}
```

React Bootstrap Integration:

Steps to Install and Use Bootstrap:

Install React Bootstrap:

Run the following command in the terminal:

```
npm install react-bootstrap bootstrap
```

Import Bootstrap CSS:

In `index.js`, import Bootstrap's CSS:

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

Use React-Bootstrap Components:

You can now use components like buttons, cards, etc., from React-Bootstrap.

```
import { Button } from 'react-bootstrap';

function App() {
  return <Button variant="primary">React Bootstrap Button</Button>;
}
```

variant="success" makes it green. Other variants: primary, danger, warning, etc.

Card:

```
import { Card } from 'react-bootstrap';

function InfoCard() {
  return (
    <Card style={{ width: '18rem' }}>
      <Card.Body>
        <Card.Title>React Bootstrap</Card.Title>
        <Card.Text>
          React-Bootstrap makes it easy to use Bootstrap components in
        </Card.Text>
      </Card.Body>
    </Card>
  );
}
```

React Attributes:

Common React Attributes:

- **className:** Replaces HTML `class` attribute.
- **htmlFor:** Replaces HTML `for` attribute in form elements.
- **onClick, onChange, etc.:** React event handlers.
- **style:** Inline styling (camelCase properties).

Example of Using Attributes:

```
function ButtonComponent() {  
  const handleClick = () => {  
    alert('Button clicked!');  
  };  
  
  return (  
    <button className="btn btn-primary" onClick={handleClick}>  
      Click Me  
    </button>  
  );  
}
```

Custom Attributes:

```
function Welcome({ name }) {  
  return <h2>Hello, {name}</h2>;  
}
```

Props with Bootstrap Example:

```
import { Button } from 'react-bootstrap';

function CustomButton({ label, variant }) {
  return <Button variant={variant}>{label}</Button>;
}
```

// Usage

```
<CustomButton label="Save" variant="success" />
<CustomButton label="Delete" variant="danger" />
```

