# Module\_44\_Modern\_Front\_end\_Core\_Concepts

#### **44-1 Module overview and modern application core parts**

* The journey of **React** started from today. **React** is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies (Wikipedia). React would be very simple if the core JavaScript concept was clear. Just complete the modules on time.

#### **44-2 Web component, 4 types of component, identify component**

* There are different parts to a website. Every single piece is called a **component.** Components can be differentiated into various types.
* A few of the components types are: **Similar in look but different in data,** **Container** component (contains a group of similar in look, different in data components), Container of different in look and different in data, and **Independent** / Stand Along / **Unique** Components.

#### **44-3 Template, dynamically create HTML elements based on data**

* array.**join(‘ ’)** - Join the elements of the array separated by a ‘ ’ (space) in a string and return it. If you don’t pass anything in the **join** method, array elements will be separated by a comma inside the string.

#### **44-4 Single Page Application (SPA), Performance benefit**

* **Single Page Application (SPA)** means a webpage where all the functions have been merged into one single page. When clicked on any button, the entire page will not reload the related content of that button instead. The best example of **SPA** is **Gmail**. Better user experience with fast and smooth loading is the primary benefit of SPA.

#### **44-5 Routing, Route parameter, Routing in React**

* **React Router** is a standard library for routing in React. It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL (Sorce: [GeeksforGeeks](https://www.geeksforgeeks.org/reactjs-router/)).

#### **44-6 Explore destructuring and send data to html elements using props**

* React Props are like function arguments in JavaScript *and* attributes in HTML. To send props into a component, use the same syntax as HTML attributes. Props are arguments passed into React components. Props are given to components via HTML attributes (Source: [w3school](https://www.w3schools.com/react/react_props.asp)).

#### **44-7 Website state, state management, purpose of state**

* Simply, state means the current situation of components like Facebook post **reacts**, total unread message, etc. When the user clicks on the Like button, the state of **reacting** increased by one, or when reading, a message is marked as read (state of the message has been changed).
* The state is a plain JavaScript object used by React to represent information about the component’s current situation. It’s managed in the component (just like any variable declared in a function). The difference is while a “normal” variable “disappears” when their function exits, the state variables are preserved by React (Taken from [medium.com](https://medium.com/edonec/state-in-react-an-overview-a182675cee2c)).

#### **44-8 Install React App and explore folder structure and edit**

* Install React by following the instructions below - [Create New React App](https://reactjs.org/docs/create-a-new-react-app.html) and explore the folder structure there. Edit some text on react starter page into **src/App.js**
* Keep an eye on [React Documentation](https://reactjs.org/docs/getting-started.html)

#### **44-9 Module Summary and concept recap**

* Summarised all of the topics of previous videos of today’s module.