What to learn?

1. React Basics
2. React Router
3. Firebase
4. Redux
5. Stripe
6. CSS in JS (Styled components)
7. Redux-saga (handle asynchronous actions)
8. Hooks
9. Context-api
10. GraphQL
11. PWA
12. Jest for Testing

**React Key Concepts:**

1. Why react exists?
2. What problems do it solve?
3. Why it makes front-end problem easier?

**The birth of React:**

Before we have only JS, HTML, CSS. Then came Jquery which could communicate with DOM or Document Object Module. Angular Js came in to structure larger websites. But as the complexity gets bigger due to huge number of actions happening on a webpage Angular started to face problems. So Facebook developed React.

What made react so popular?

There are 4 key concepts.

1. **Don’t touch the DOM. I’ll do it.**

Many frameworks were manipulating the DOM directly. Which is the **Imperative** way. In Imperative way, one directly changes his app in response to various user events. Overhead cost due to Repaint and Reflow.

So, **React** came with the **declarative** approach. Instead of telling what to do next if this happens, we gave the blueprint or declare what could be happen if the page is in that state and react manipulates the DOM automatically. It results in less complexity, better code, faster development.

1. **Build websites like logo blogs.**

React uses Components which can be Re-used. Components are just JS functions that receives some input or attributes and return html inside JS. **JSX** file.

1. **Unidirectional Data flow**

Virtual Dom due to its dependence on State. When state changes, it changes the virtual DOM. Data can only move down. This additional restriction gives us better code which is easier to debug.

1. **UI, the rest is up to you.**

It only works as view. Anything you need just customize the app. Learn once write anywhere. It doesn’t depend on the development stack.

**Some React Keywords:**

State, Declarative, Props, JSX, Virtual DOM, Components.

**The Job of a React Developer**

1. Decide on Components.
2. Decide the State and where it lives.
3. What changes when state changes.

**React Basics**

1. Install latest version of Node.js (<https://nodejs.org/en/>)
2. Create React app (<https://reactjs.org/docs/create-a-new-react-app.html>)

>> Check the Node and Npm version by running “node -v” and “npm -v” in the terminal.

>> “Yarn” can be used too instead of npm.

**Create React App**

**To create the react app run:**

1. npm rm -g create-react-app
2. npm install -g create-react-app
3. npx create-react-app my-app
4. cd my-app
5. npm start

React uses **WEBPACK** and **BABEL**. Babel takes all JS files and make sure that it runs on every browser. Webpack builds modular files structure creating bundles and optimize it.

aa