Friday, December 15, 2023 3:29 PM

Good Afternoon : )

React ------- JS Library with Model view Controller architecture
Model = data
View = presentation/UI
Controller = the code that MAPS the model with view

---React uses Component Archtitechture-----
Allow us to write components.
What is a component ? Unit that has some responsibility .

React Application ---- is made by integrating multiple components!!!

-----React is used to build SPA = Single Page Application
1. first request will fetch the React-App with all UI parts
UI parts( html ,css, js, images etc )
2. subsequent requests will fetch data or send data to server using
AJAX .--- the end user keeps seeing the UI
while the data is fetched or sent to server
ADVANTAGE -- end user gets a SEAMLESS UI experiance

3. Server side services will only send DATA ( JSON ) to
the client so ADVANTAGE ---subsequent requests will be
ligher as they do not have VIEW components in them

React Development Phase
React template project has a DEV-Server that is a web server
we will use this Dev server while developing and testing our
React APP

React Deployment Phase
Integrate the React App with Your Fullstack Server(Tomcat, IIS)

TO start with React
1. Download the template project
npx create-react-app projname
OR
npm install create-react-app projname

cd projname
npm start } deploy the project ON DEV SERVER @ http://localhost:3000
server is also started
Open the browser --- give url http://localhost:3000
observe index.html

2. We get a template project for React --
template project with projname is a NODE project
so it has node configuration file = package.json
node version control file= package-lock.json

all node libraries are stored in node_modules folder
HTML is stored in public folder
React components are stored in src folder

---

App.js = FUNCTIONAL REACT COMPONENT

Two types of React Components ---
1. Functional Component
2. Class Component

The react component must return a JSX object !!!

JSX = Javascript Extension
= this has html like tags
= these JSX tags are CONVERTED To HTML using a library Babel
JSX ======>BABEL ========> HTML
= this HTML is added in the  HTML PART GENERATED 
= Everything in  ___ is displayed on browser !!

EX 1 = create another component Welcome - functional component !!

1. create a file welcome.js in src
2. add the function to it
3. export the component
4. import in App.js
5. Add the component in the JSX of App.js

function must return a SINGLE JSX object - so if we have many tags, then we should WRAP the tags in ONE outer tag and then return

How is the component Used in Another component ?
Ans = It is used in the form of custom/user-defined TAG !!!

We are not calling the welcome function . So who calls welcome function ?
The ReactDOM = i.e. the React container will call the welcome function

Ex 2 - Write Greeting functional component in the same file and add it to App.js

Ex 3 - Write a Maths functional component in maths.js

To use variables in JSX - use Interpolation notation { }
= we can write anything in the { } that evaluate to single value

To send data from OUTER component to INNER component ====> PROPS
OUTER COMPONENT = Parent
Inner component = Child
In our ex - OUTER component = <App>
Inner Compnent = <Maths>
<App> -----> <Maths >

Ex 4 - Write a Total functional component in total.js
pass array of numbers in the props

Ex 5 - write Icard component , pass json object in props