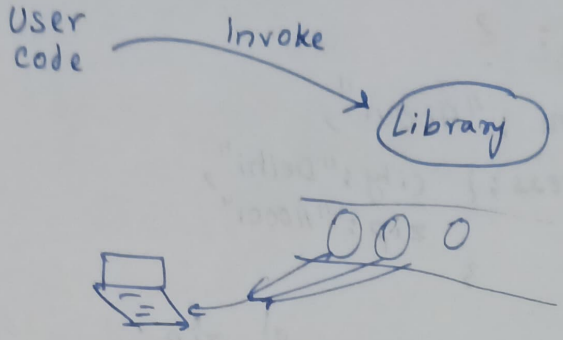
React

JS Library → UI create

React ek JS ki library hai jiska sole purpo hai UI create krna hai



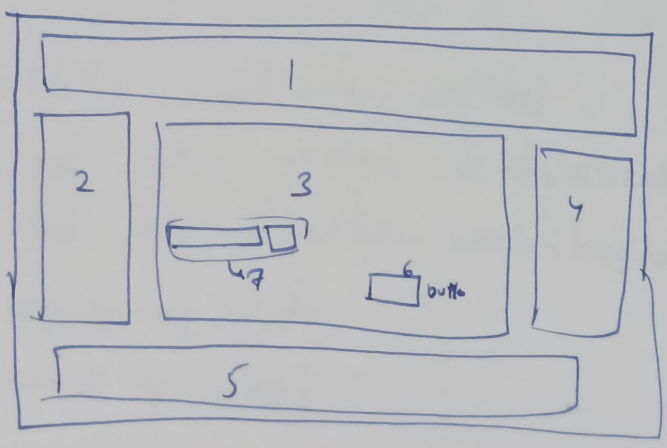
Magicline

"React is all about components"

React → component based architecture

↳ a reusable piece of code

- for understanding: component ek ONLY Custom HTML element create kar ke de raha hai
  - Don't tell INTERVIEWER /anyone!



Optimal no. of component to be used in a website

↓

understanding will come from experience.

- no of components depends on developer

Why React?

* more code = more low level detail
Normal JS is based on
Imperative Approach
(Step by Step, line by line)
(all things done)
(in detail)

only have to say end state.

* easy.

Eg: Love Babbar == React ; no need to tell how to teach web D.
You want to learn, chup chap padho.

React se easily UI create kar sakta ho, in comparison to JS.

In React, we say end state. This is declarative approach.

SPA Approach
↳ Single Page Application

Govt. Website
Netflix

(H.W) surf.; check responsiveness

React Alternatives : Angular, Vue, Next

↳ Core Syntax & Setup & JSX
↳ Component
↳ Data flow in component

React why?
- Component
  - reusability
  - DRY (DO Not Repeat Yourself)
  - Readability
  - maintenance
  - SoC (Separation of Concern)

Way ① to set up the React project

Step ① Install nodejs
② vs code
③ Create new folder → ReactFolio
④ Change directory to ReactFolio
⑤ npx create-react-app demoshopapp
⑥ change directory to demoshopapp [cd demoshopapp]
⑦ npm start

Way ②
replit

① Create Repl
→ React JS

Way ③

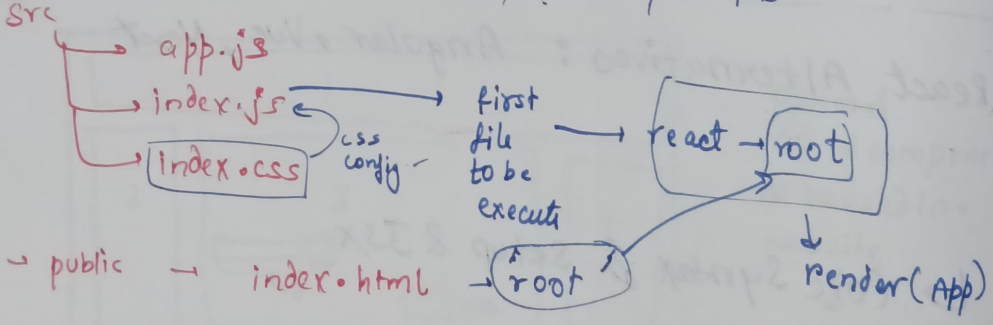
tailwind StarterPack

"react"
"react-dom" } These two packets are responsible for actual React Library.

---

src
index.js → jab page load hoga (i.e. HTml file)
then index.js is first file to be executed
This is entry point of JS

package.json → has dependencies / version / scripts

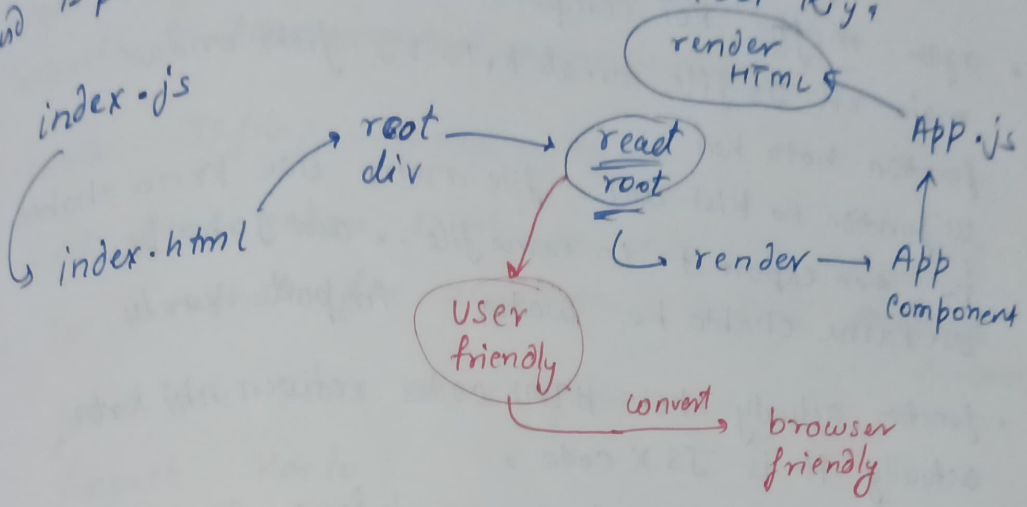


→ public → index.html
→ node modules

somewhere app.js
" app.jsx

(JS XML)
(special syntax)

sabse pehle index.js execute hui, wo index.html se root element uthake leayi, is root div ko convert kiya react ka root main and is par App component ko render kiya,



JS mein class is reserved, so we use className.

functions / class → to use in other file

export

import

when, npm start command run;

react mein likha hua code → convert → equivalent HTML/JS code that is understood by browser.

- As a good dev, components ka alag file banaenge

component naming like this = ItemDate.js

Capital

(char-word ka first letter cappital)

Eg- ItemDate.js
ItemDate.css

container ka folder in src

* css file ko link karne keliye import.

- agar mujhe koi component create karni hai, main ek JS file create krta hu, jiske andar function hota hai.
  Us function ko kisi aur file mein use krna chahta hu, then export it in same file.. aur jaha pe use karna chahta hu, waha import kar lu

- function actualy koi HTML code return nhi krta, actually it is JSX code.

- To convert JSX code to browser's lang., done using npm start → HTML code inside JS code (JavaScript XML code)

Eg: you create a file ItemDate.js in components also create ItemDate.css.

```
[import './ItemDate.css']

function ItemDate()
{return(<div>
    <div>20</div>
    <div>June</div>
    <div>1998</div>
</div>);
}
export default ItemDate;
```

in ItemDate.js

→ do not Hardcode like this, but in actual case, we make API calls & insert data dynamically

- If I wish to use this ItemDate component in App.js, in App.js, have to import:

```
import ItemDate from './components/ItemDate';
```

Steps to Create your Component // Can you create JS component?

- Inside components file, create a file as my-comp-name.js
- create my-comp-name.css
- import : ./my-comp-name.css
- Inside JS file, create a fnc.
- Inside fnc, return HTML ka code with one parent.
- To use that fnc anywhere, export default fnc-name;
- jaha use krna ho component ko, import karlo

Eg: import comp_name from '...'

Note: fnc. name & comp. name should match

Eg: App.js

```
function App()
{ return (<div>...</div>);
}
```

Can you style that component?

resam ✓✓

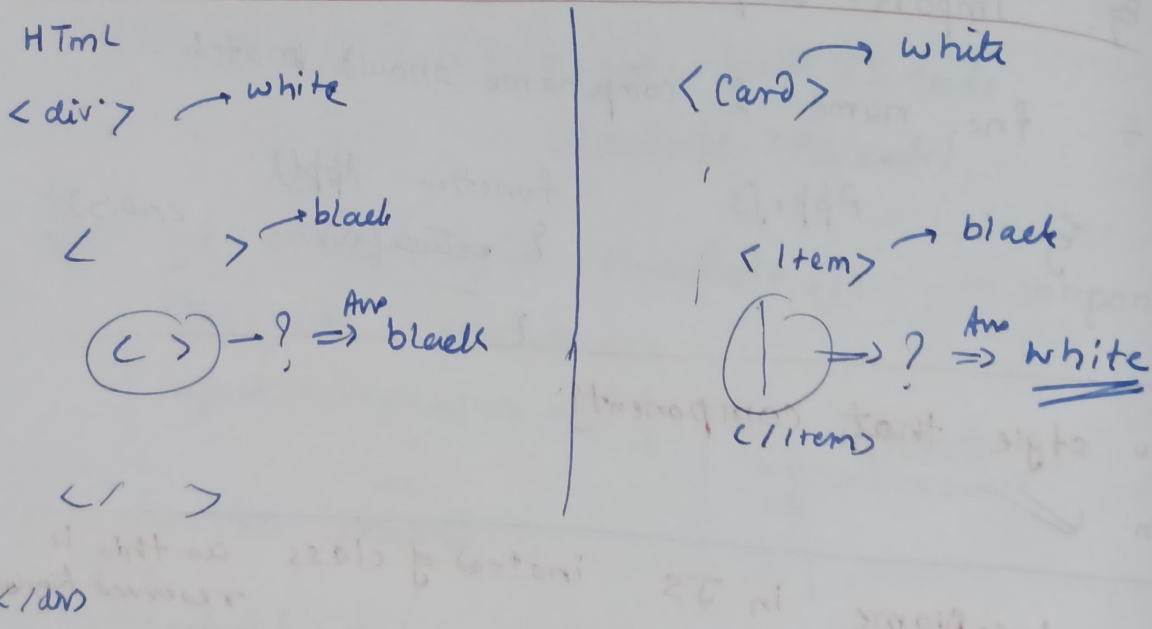
- use className in JS instead of class as this is reserved keyword.

- concept of props

- function Item (props)
  {

concept of props.children

component ke andar ka content by default not visible
To make it visible, use props.children



{ props.className }

H.W

const element = <h1> Hello </h1>

↳ neither string nor HTML

↳ Its JSX.

JSX produces react elements

WHY ? JSX ?

React separates concerns with loosely coupled units called "components"

React do not require using JSX, but it helps visually, as allows React to show useful error & warning message

Embedding expressions in JSX

```
const name = 'master900';
const element = <h1> Hello, {name} </h1>;
```

we can put any valid JS expression inside { } in JSX.

Components

Let us split the UI into independent, reusable pieces & think about each pieces in isolation.

Start component names with Capital letters bez, React treats component starting with lowercase letters as DOM tags.

props ke dwara data send krte,
" " " event handling bhi ✓

<button onClick={clickHandler}>Add to Cart</button>

onClick → Starts with on

clickHandler → button click hone par yeh fnc call hoga

Variable mein value change krne se screen repaint nhi hota, UI update ✗

To update UI, use concept of STATES.

* react is dec. approach ⇒ need to ~~state~~ mention target state

W/o state ke sath no re-render,
with state, re-render ✓

React ~~ka~~ mein event handle karne ke liye, use props.

In vanilla JS, it was imperative approach, first fetch element from document then . . .

In React, do props addition to handle events
All these props starts with "on".

<button onClick={clickHandler}>Add me</button>

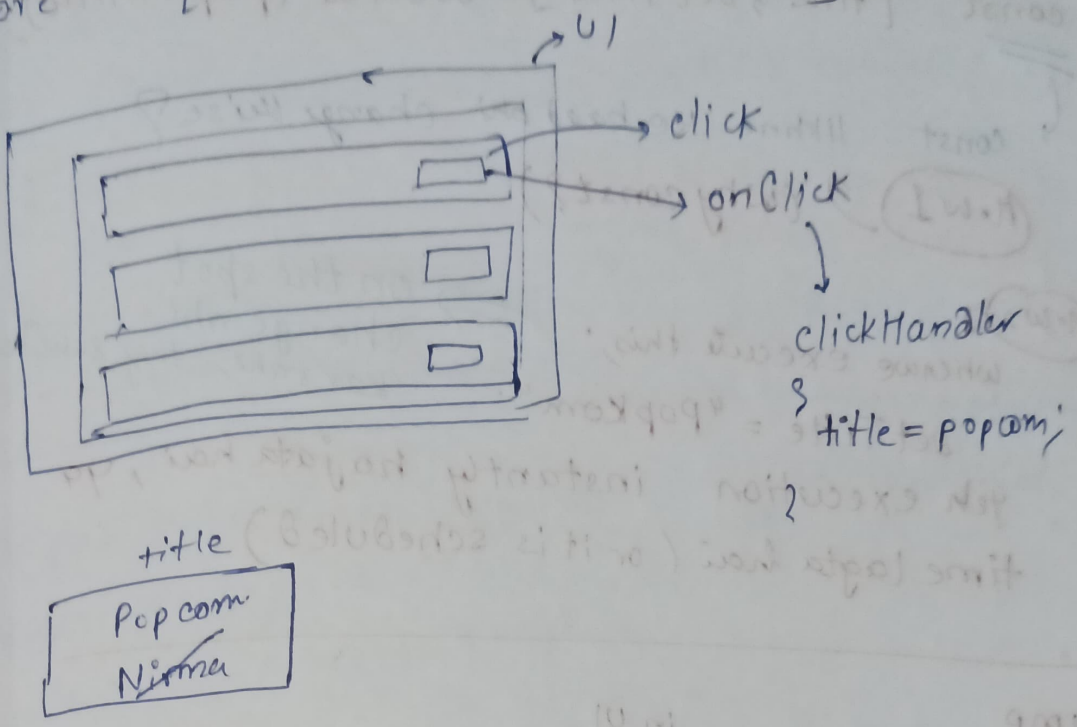
onClick → event ~~ki~~ props

clickHandler → do not use ( ).

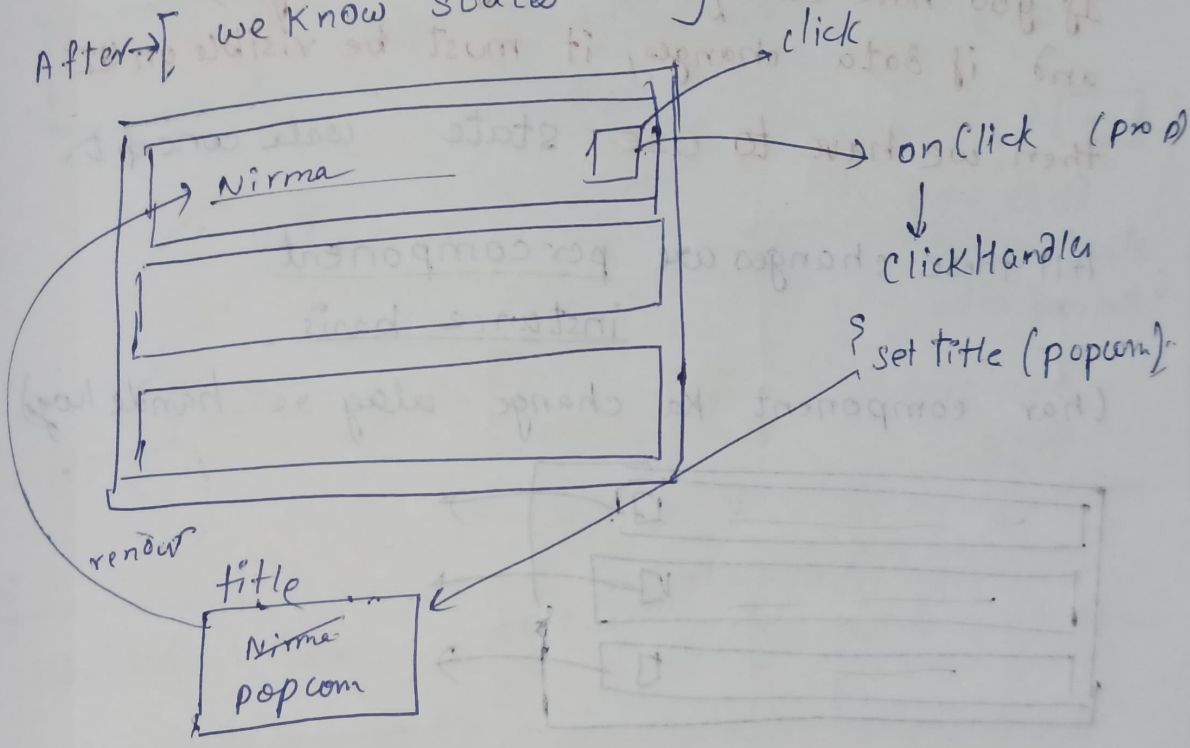
onChange,
onSubmit, etc.
onInput

har ek event prop ~~ki~~ handler fnc. se map krna pdega

Before → [we don't know about state]



After → [we know states now]



const [title, setTitle] =

useState Hook always returns an array with two elements. First is value of variable, second is fnc to update the value.

const [title, setTitle] = useState(props.title);

↳ const likhne ke baad bhi change kaise?

H.W 1 → (why const?)

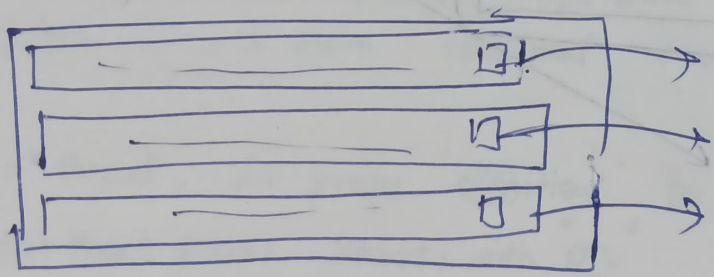
H.W 2 → whenever execute this, setTitle = "Popkorn". → on the spot change nhi kar rahe, they schedule yeh execution instantly hojata hai, yq time lagta hai (or it is scheduled)

Imp

If you have data in UI that might change, and if data changes, it must be visible on UI then we have to use state wala concept.

All these changes are per component instance basis.

(har component ka change alag se handle hoga)



---

state → useState (1, f)

const [,]

title, setTitle

handling multiple parameters / State

```
const [fullProductInfo, setfullProductInfo] = useState({
    title: '',
    data: ''
});
```

but, this doesn't increase performance

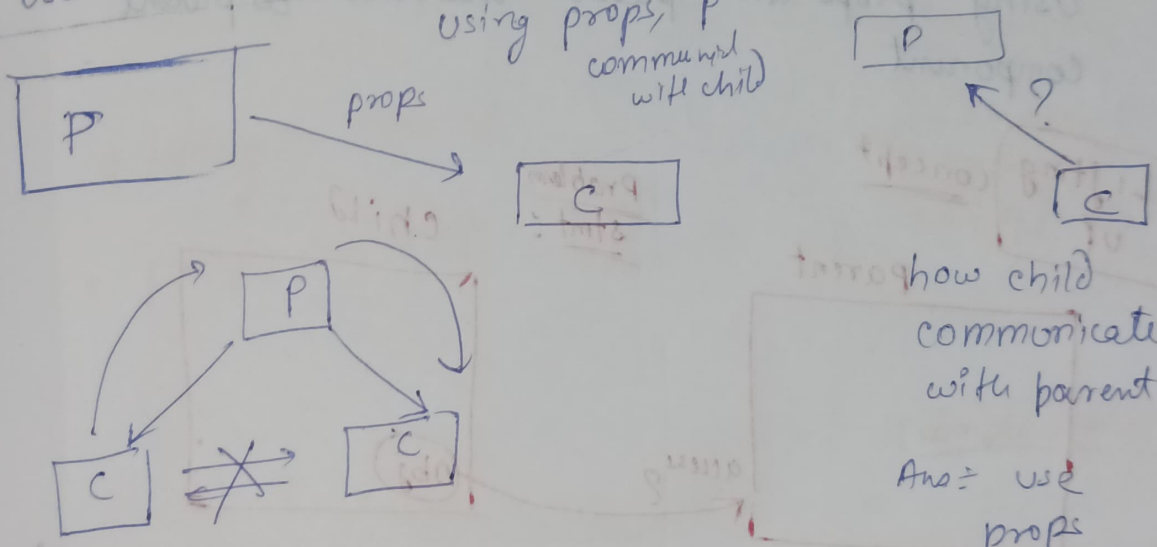
(skip for now).

---

we pass props in a component & it gets rendered.

state belongs to a component

useState is a hook



---

useState hook returns a array of two items.

1. value of variable
2. a function that can update the value of variable.
   (basically a setter fnc)

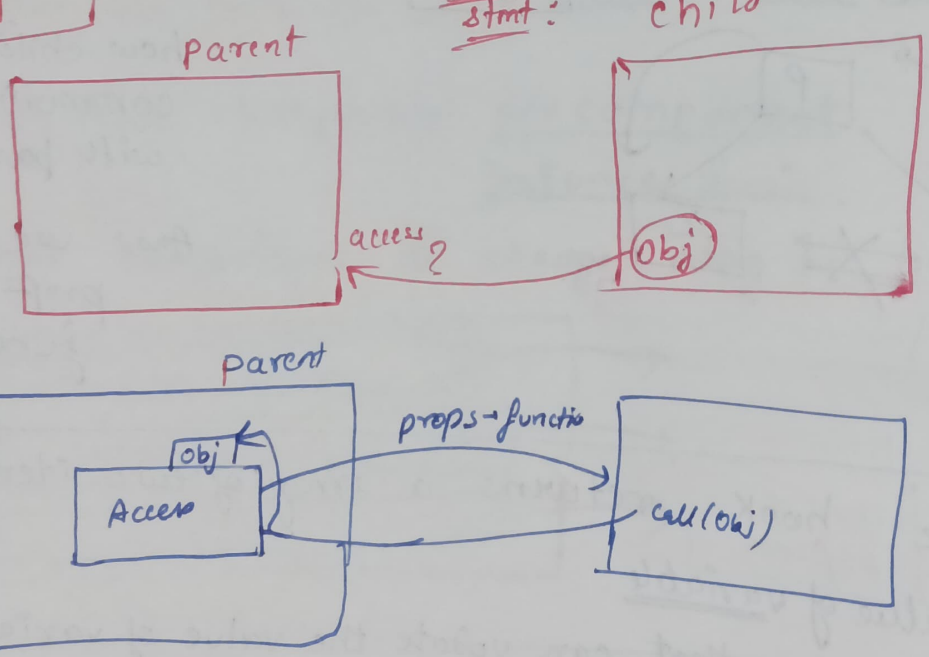
To pass data from parent comp. to child comp, do it using prop.

Steps to pass data from child components to parent component :-

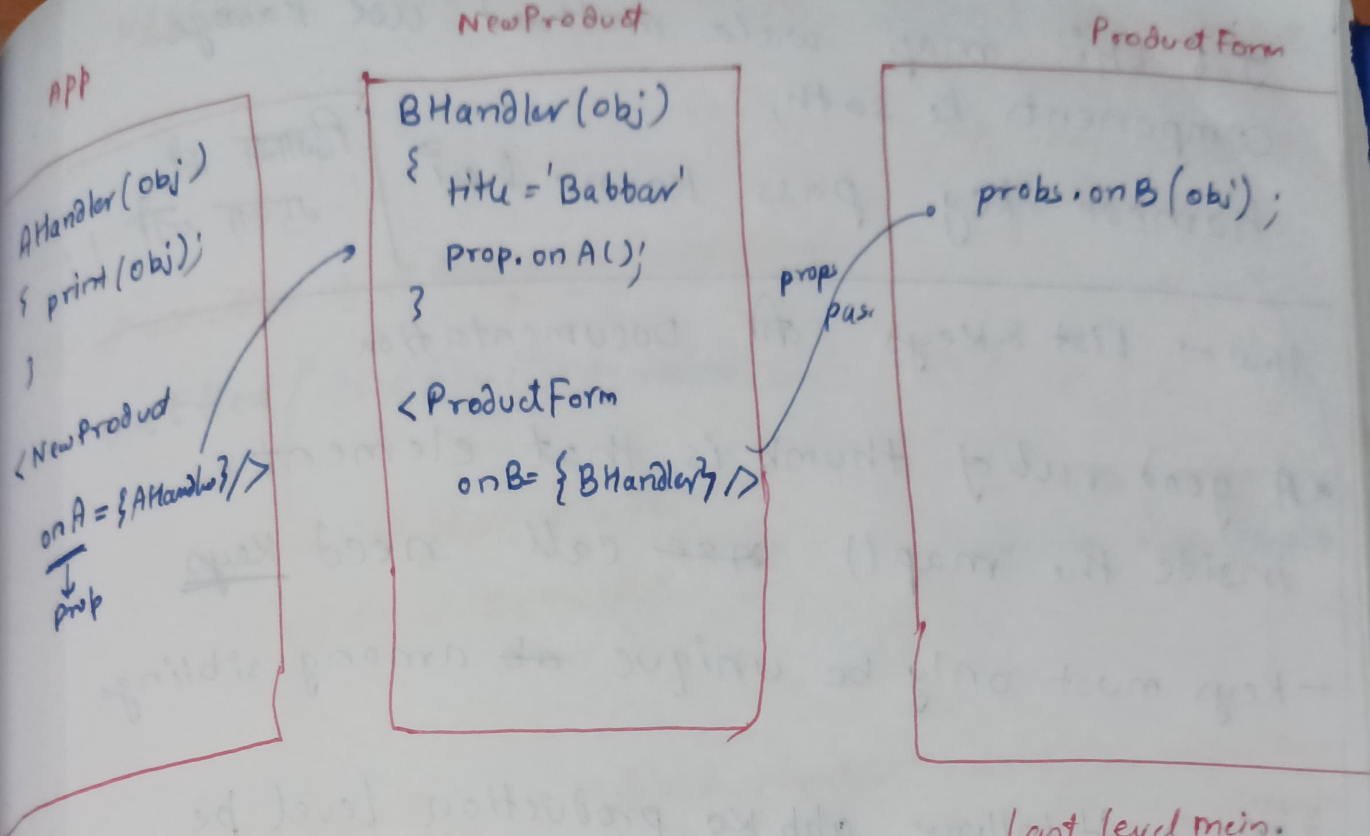
- In parent component, create a callback function. This callback fnc will retrieve the data from the child component.
- Pass the callback fnc to child as a props from parent component
- Child component calls the parent callback function using props and passes the data to parent component

Lifting up concept

Problem stmt :



isko prop drilling bolte



Last level mein only prop call ho raha, baki mein Handler fn bhi h.

* lifting the state up
or
* prop Drilling

(complete H.Ws)

•) Documentation follow !!

Props are argument passed into REACT components

props stands for properties

Jab bhi map wala method use karoge components ke sath, hamesh key pass karna hai | दिमाग में घुसा लो !!

H.W → List & Keys की Documentation

* A good rule of thumb is that elements inside the map() ~~ke~~ call need Keys
- Keys must only be unique ~~ab~~ among siblings

↳ if u don't follow, app ko production level pe build hi nahi kar paoge, host nhi kar paoge

---

→ useEffect Hook → to manage SIDE EFFECT → ?

A side effect in React is a change that affects something outside the component being rendered.

eg: of side effects : ✓change in document title,
✓modifying state of parent component,
✓making an API call,
✓modifying browser's history
✓HTTP request
✓API call

jis bhi component ke andar useEffect Hook likhte hai, us component ke render hone ke baad, iske andar ka code render hoga

// variation 1

```
useEffect ( () => { console.log("UI render done"); });
```

har ek UI render par useEffect ke andar ka code chalne wala h

usecase : har ek render pe yek ek particular xyz chiz krni h. do using useEffect

// variation 2 // Only on First Render

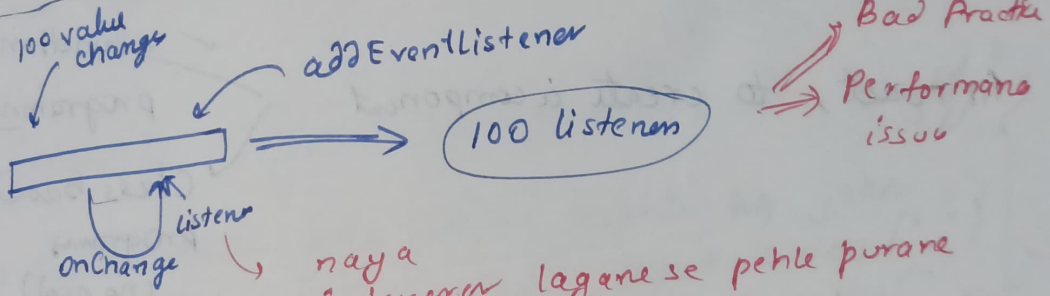
```
useEffect( () => { c.l.("UI rendered"); } , [ ] );
```

* only @ first render code is executed

empty array denotes dependencies

// variation 3 : on first render + when dependency chang

```
useEffect( () => { console.log ("change observed"); } , [text] );
```



naya listener laganese pehle purane wale remove karna hoga.

// variation 4 : to handle unmounting of compond

```
useEffect ( () => {
    [B]
};
    [A] (executed first)
return (
    [ ]
);
```

for cleaning purposes (like remove Event Listener)

useEffect (____ , [])

↓ call back function

↳ array of dependencies

" [component ke normal flow of execution के अलावा तुम कुछ additional task perform करना चाहते हो, उस Task = side effect, to handle side effect, tarika is useeffect.] "

component mount ho gya ⇒ i.e. component DOM pe render ho chuka

component unmount Krna ⇒ DOM se component remove kar diya

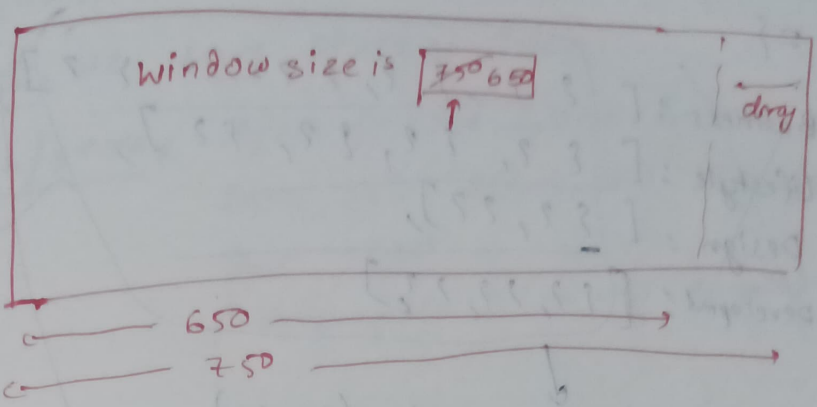
In react, to create a component → functional programming / class based programs (no new)

class based components

- componentWillUnmount
- componentDidMount
- componentDidUpdate

(lifecycle)

H.W]



To add toast in react :-

npm i react-toastify

in parent file

```
import { ToastContainer } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";
```

harek toast ko show karne ka pre-req. hai ToastContainer banana padega.

```
import { toast } from "react-toastify";
```

↳ jaha toast use karna chahte ho, jaha code likha hai, like toast.success (^_^)
toast.warning (-_-)
toast.error (>_<)

course

data = {

Business : [{ ? , { ? , ? ? } ?]

Lifestyle : [{ ? , { ? , { ? , ? ?] ,

Design : [{ ? , ? ?] ,

Developmd : [{ ? , ? ? , ? ?]

Object.values(courses)

I want

[{ ? ?

{ ? ? ,

{ ? } ,

{ ? ?

:

:

:

{ ? ?

]

Object.values(courses).forEach(

(array) => {

array.forEach( )

}

Object.values(courses).forEach(array => {

array.forEach((courseData) => {

allCourses.push(courseData);

})

});

App.js

<h1>

<Testimonials/>

underlying

<Card/>

<

>

{SurpriseMe}

{icon('')}

{icon('')}

Forms (in React)

In React, we save state of Form.

In HTML forms, After onClick Submit button, sara data accumulate kiya jayega, then create a object, finally pass it (to an API call)

While in react, using the useState Hook, I can save the state of all fields of the form.

* No need to accumulate
* Just need to send data to API

Homework: ① go through Documentation
② <fieldset>
③ <legend>

```
const [formData, setFormData] = useState(
    { firstName: "", lastName: "", email: "", comments: "",
      isVisible: true, mode: "", favCar: "" }
);

function changeHandler(event)
{
  const { name, value, checked, type } = event.target;
  setFormData( prevFormData => {
      return
          { ...prevFormData,
            [name]: type === "checkbox" ? checked : value
          }
  })
}

function submitHandler(event)
{
  event.preventDefault();
  console.log("Finally printing entire form ki data...");
  console.log(formData);
}
```

```
return (
  <div >
   <form onSubmit={submitHandler}>
   <br/>
   <input
    type="text"
    placeholder='firstname'
    onChange={changeHandler}
    name="firstName"
    value={formData.firstName} />  <br/><br/>

   <input
    type="text"
    placeholder="lastname"
    onChange={changeHandler}
    name="lastName"
    value={formData.lastName} />  <br/><br/>

   <input>
    type="email"
    onChange={changeHandler}
    name="email"
    value={formData.email}
    placeholder="emailid"
    />  <br/><br/>

   <textarea
     placeholder='entercomment'
     onChange={changeHandler}
     name='comments'
     value={formData.comments} />  <br/><br/>

   <input
     type="checkbox"
     onChange={changeHandler}
     name='isVisible'
     id='isVisible'
     checked={formData.isVisible} />  <br/><br/>

   <label htmlFor='isVisible'>Am I visible?</label>
   <br/><br/>
```

<fieldset>
<legend> Mode : </legend>
<input
type = "radio"
onChange = {changeHandler}
name = "mode"
value = "Online-Mode"
id = "Online-Mode"
checked = {formData.mode === "Online-Mode"} />
<label htmlFor = 'Online-Mode' > Online </label>
</br>
<input
type = "radio" onChange = {changeHandler} name = "mode"
id = "Offline-Mode"
checked = {formData.mode === 'Offline-Mode'} />
</fieldset>

we use fieldset for grouping of multiple input elemnts

note, these radio btns have same name, tabhi can chose one

<label htmlFor = 'favCar' > Tell your favourite Car </label>
<select
name = "favCar"
id = "favCar"
value = {formData.favCar} onChange = {changeHandler}
>
<option value = "scarpio" > Scarpio </option>
<option value = "Thanr" > Thanr </option>
<option value = "Fartuner" > Fartuner </option>
</select>

// <input type = 'submit'
value = 'submit' />

<button > Submit </button>

</form>

- form ke andar button banaoge to by default onSubmit wala event trigger hoga

event.target ⇒ jo element ne event fire kiya, woh element is darshata by event.target

setFormData (prevFormState ⇒)

return

{

... prevFormState,

[event.target.name] : event.target.value

Syntax error hi h !

single statement se

text, email, comments sambhal li.

(not checkboxes)

because it do not have value

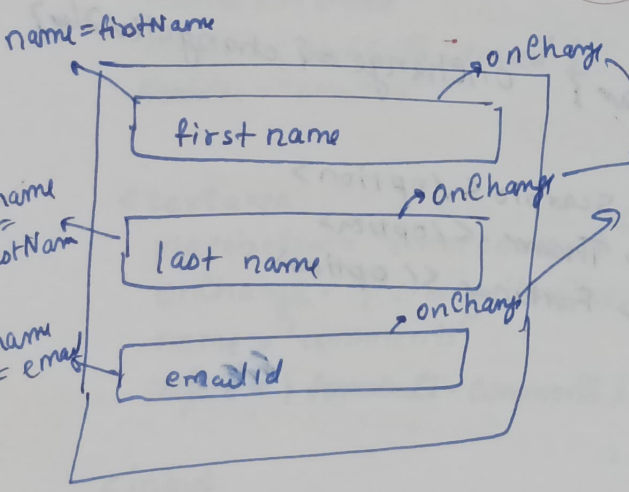
}

})

console.log (formData.firstName)

console.log (formData.lastName)

console.log (formData.email) → field ki name !



changeHandler

{

setFormData (prevstate ⇒

);

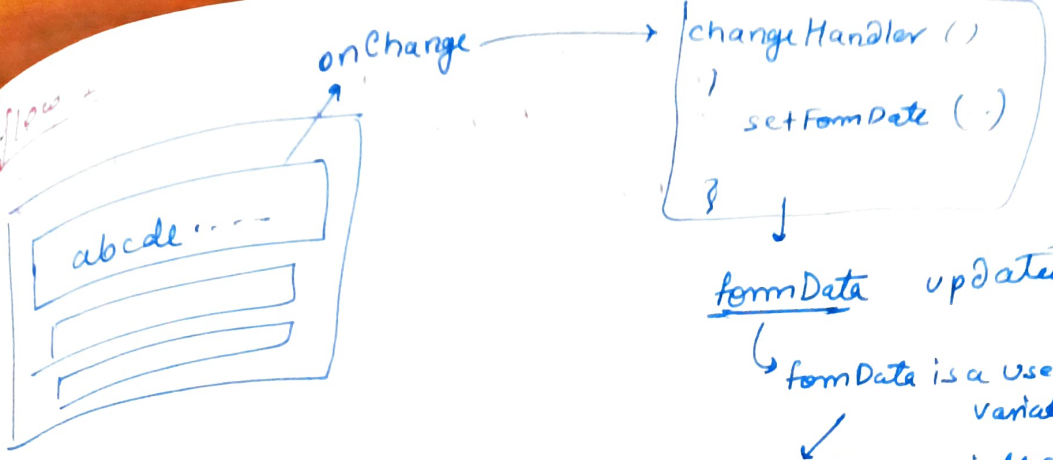
① Copy prev. state

② jis element ki wajah se trigger hua, uski value update kar rha hu

return

}

}



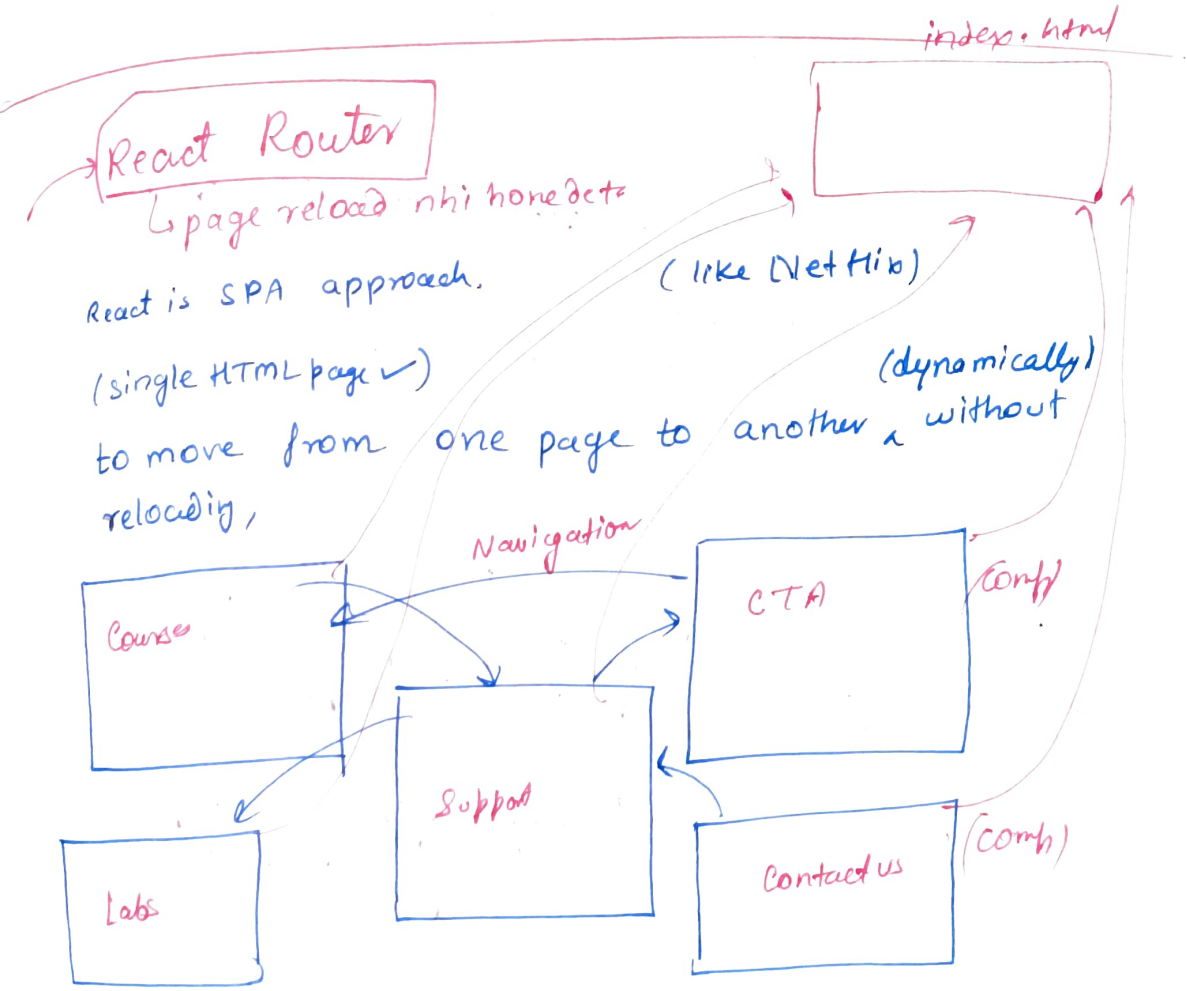
formData updated ✓

↳ formData is a useState variable

useState variable gets updated

re-render UI ← JSX code ← run

```
<input
  value={formData.firstName}
```



hume aisa lag raha h ki hum dusre page par ja rahe hai, but reality mein different diff. component same page par render ho raha hai

(React is all About COMPONENTS!!)

react router → a framework using we can navigate btw multiple pages without refreshing the page.

npx create-react-app

npm start

npm install react-router-dom

- use browser router tag

- <BrowserRouter>
  <App />
  </BrowserRouter>
  → App component must be wrapped in <BrowserRouter>

- import { BrowserRouter } from 'react-router-dom'

~~<BrowserRouter>~~
  <Routes>
    <Route path="/"
    element={ Home Page }
    />
  → define a single path/route

~~</BrowserRouter>~~
  </Routes>

[you cannot render a <Router> inside another <Router>. Not more than 1 router in 1 app]

react ke andar <a> anchor tag fnc. is replace by <link> tag