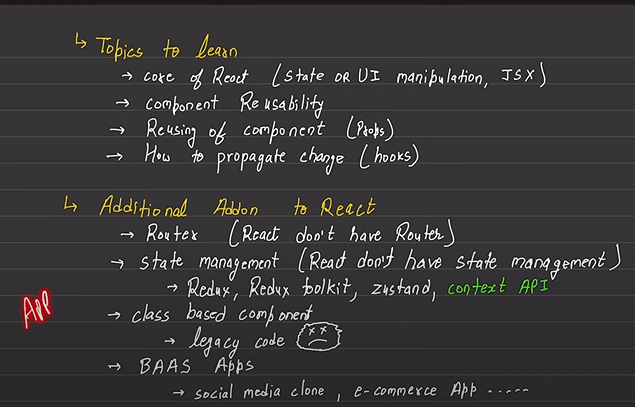
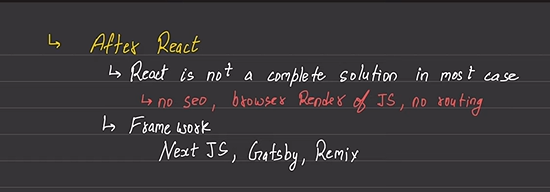


React=library--🡪so we can freedom to work on our role

In framework=-> ex jsx we want to work as given rules





8rh video

For counter if I said that in addCounter if

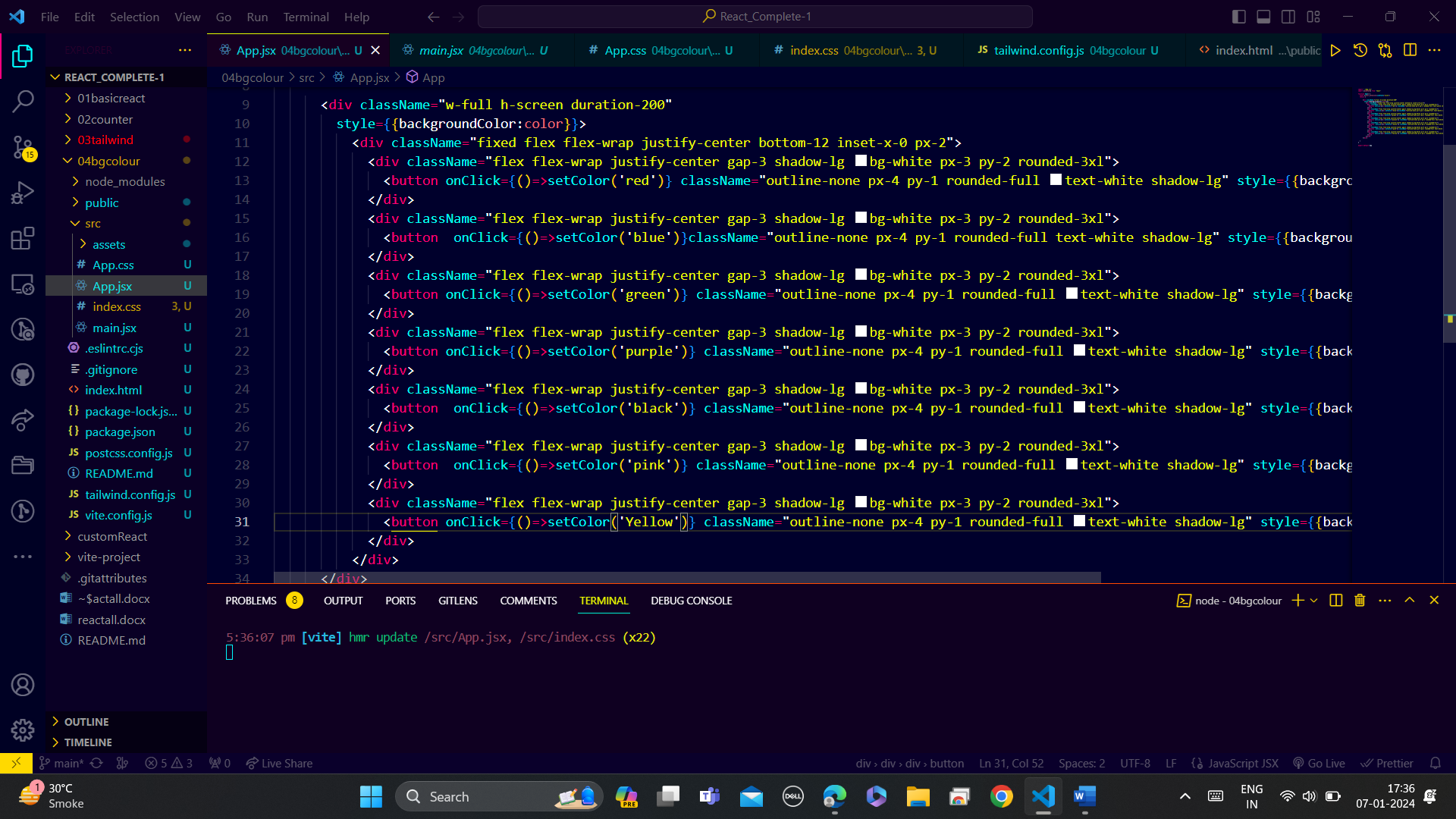
setCounter runs 5 times then what is the value



As this will not add the 15 to 19 it only ---but It will only add 1 value in it

Bez react fibre --- it will take all in bundle so only 15 will be go again in again in setCounter

9th bgcolour changer



Here onclickwants function as in the setColor we pass paramthessis but we can’t do link onClick={setColor(‘red’)} we want function to execeute the onclick

10th:

Here for increase the length use= useState(8)

For checkbox true and false will be there = useState(false) --- by default

useCallback

* If **useCallback** is not used, a new instance of **handleClick** would be created on each render of **ParentComponent**. This could lead to unnecessary re-renders of child components that use this callback, even if the function logic remains the same.
* By wrapping the callback with **useCallback**, the function is memoized and only recreated if the dependencies provided in the dependency array change. In this case, an empty dependency array **[]** indicates that the callback does not depend on any variables and will remain constant.

This callback done for rerender like something without refreshing the page depend on the dependencies if array [] is null then it will not depened upon dependencies and only run the function inside the useCallback

What done

1. Varibles will be initalize like password allowed , charallowed and all….
2. Password generator function str,pass,number and character … for loop loop pass+

useCallBack …. setPassword= optiize setPasswoord if I used password then it will go under loop

1. Input with onclick and defaultvalues
2. Copy to clipboard using useRef will be done--- using window
3. Then take reference like craete then pass that referech then use this passwordRef.current?.select() ---------------------do using use

Do by this useCallBack

11th

**Currencyconverter**

1st hooks folder

We want to fetch the data

So we fetch --- using api and store this in the personalized hook or function and then export

--------🡪 output is data what we want in json format bez data is in string

2nd components

Here we take all the input of the conversion

Label

Amount-🡪 enter by u

    onAmountChange,

    onCurrencyChange,

    currencyOptions = [],

    selectCurrency = "usd",

    amountDisable = false,

    currencyDisable = false,

    className = "",

here 2 section are made input and selection(with map) so that we can use by main file

3rd App.jsx

const [amount,setAmount]=useState(0)

  const [from,setFrom]= useState("usd")

  const [to,setTo]=useState("inr")

  const [convertedAmount,setConvertedAmount]=useState(0)

  const CurrencyInfo=useCurrencyInfo(from)

  const options =Object.keys(CurrencyInfo) *// this will give all key value pair but we want only keys so use this*

  const swap=()=>{

    setFrom(to)

    setTo(from)

    setAmount(convertedAmount)

    setConvertedAmount(amount)

  }

Here by using props we do all the things

Like in components we declare all the things like card is created and here we give only tha props name like lable name selectquery like only give props name and all the card will be printed

12th

**React Router**

Install react-router-dom

Then we want each of the pages and then route it

Using router--🡪 route all

1. Concept of outlet -🡪

Outlet is like that we will edit in tha page and inside this we will insert all pages

1. Then next one is 🡪 user id’s from url using paramas
2. import { useParams } from 'react-router-dom'
3. function User() {
4. const {userid}=useParams()
5. return (
6. <>
7. <h1 className='bg-gray-600 text-white text-3xl p-4'>User: {userid}</h1>
8. </>
9. )
10. }
11. export default User

3. then new is get github by using useState( fetch ())

Or another method is

Use useLoader so that when we take the arrow in Github page before clicking on page website get alll the info

By await fetch

const data=useLoaderData()

  return (

    <div className='text-center m-4 bg-gray-600 text-white p-4 text-3xl'>Github Followers:{data.followers}

    <img src={data.avatar\_url} alt='Git picture' width={300} />

    </div>

  )

}

export default Github

*// another method for loader in main.jsx we use loader so that this function will be called*

export const githubInfoLoader=async()=>{

    const response= await fetch('https://api.github.com/users/prajaktasangolkar')

     return response.json()

}

  <Route

         loader={githubInfoLoader}

         path='github'

          element={<Github/>}

          />

React -roder-dom

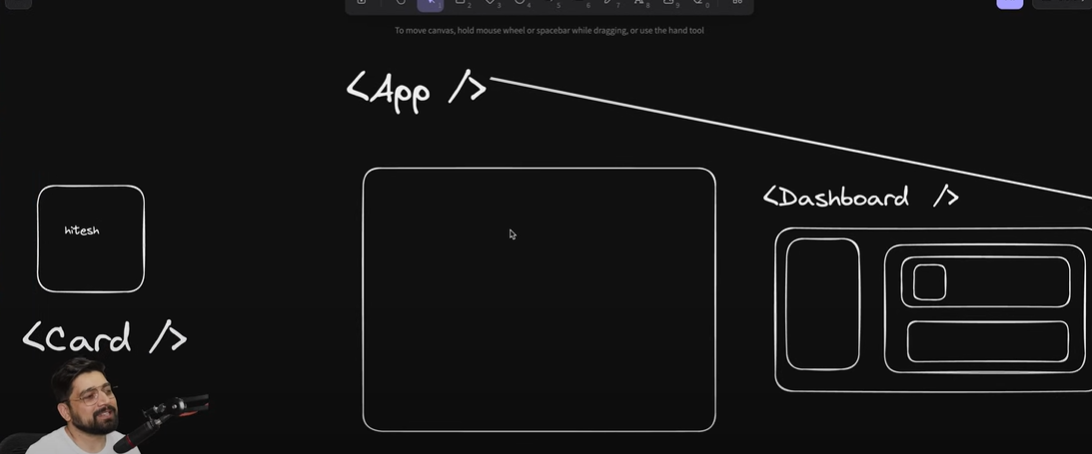
Done by router creation in main.jsx file

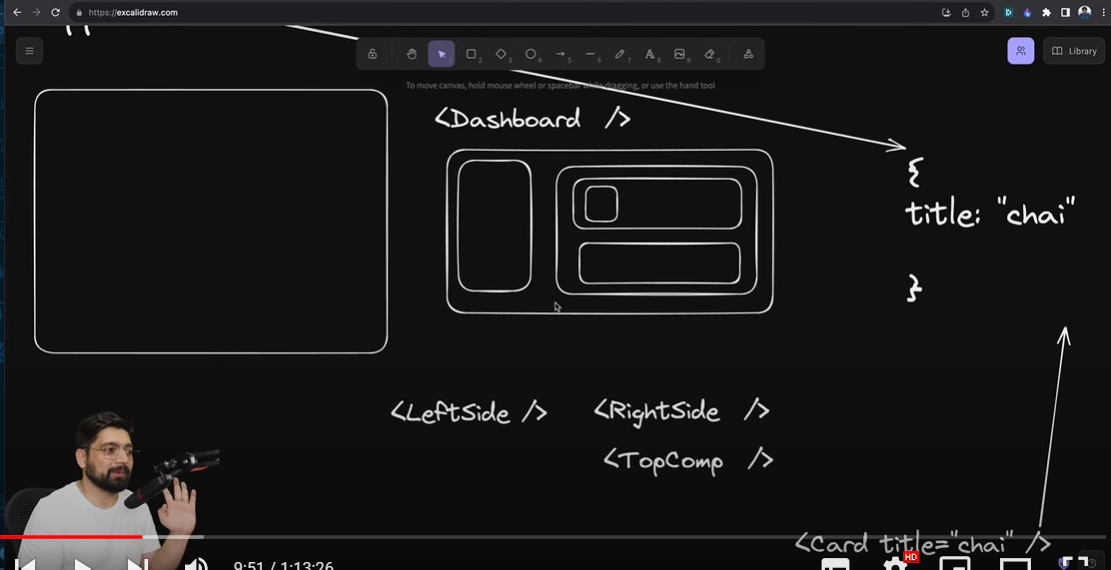
Then do all the things

And make layout page first

13th

Context Api



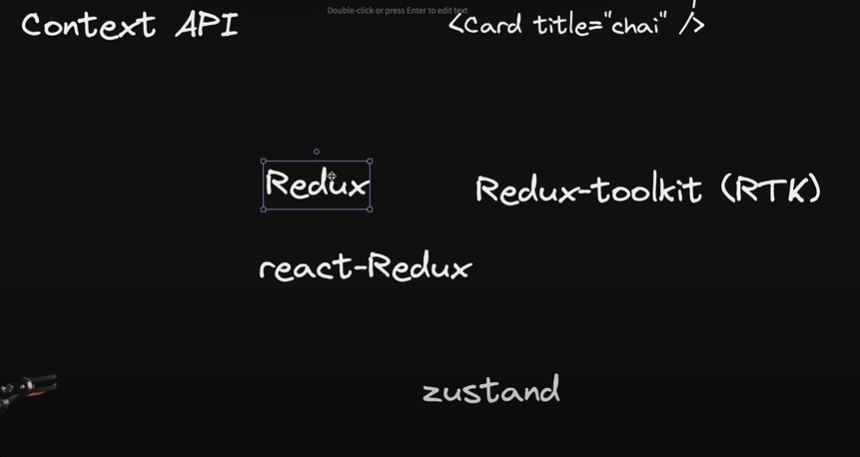


Like want card detail inside app ->rightside->topcomponent->card

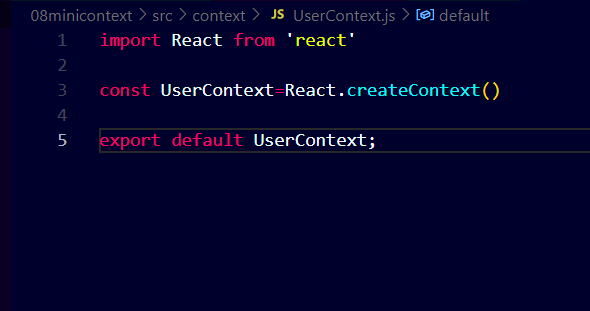
So early we want to write like <App title=’chai’/> -> <rightside title=’chai’/> -🡪 <topcomponent title=’chai’/> 🡪 <card title=’chai’/>

Then context api is evolved where not need to write for all components the same things

{ title=’chai’} only and direct connnect to card inside App



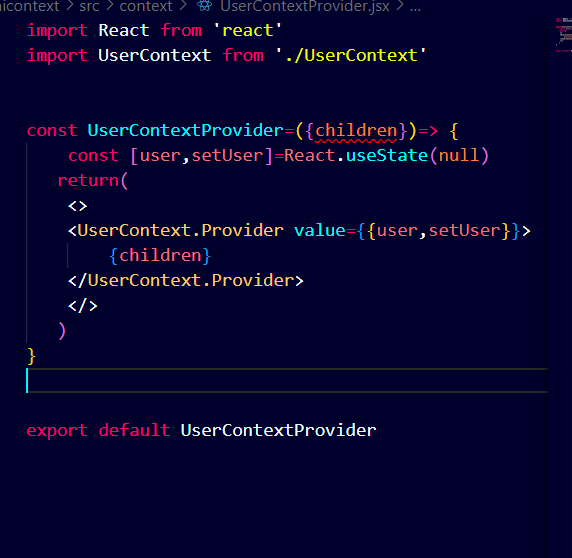
All type of contextAPI



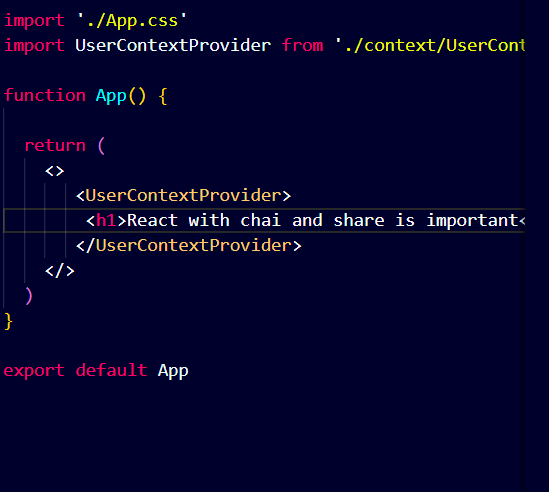
Context file created

Context is nothing but global provider inside that whatever will be done that will goes for all

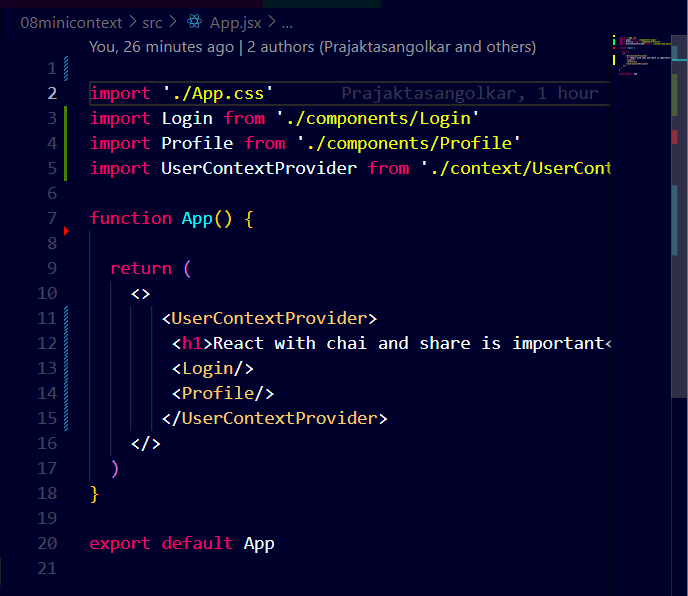
*// it will take all state excess*



Use that useContext and rapping of all the components will be done her

e

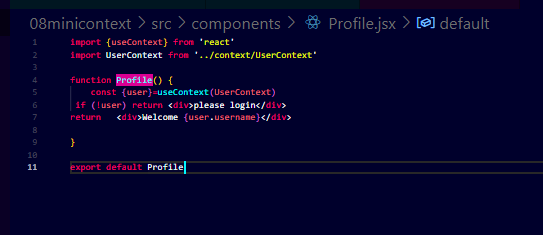
We will wrap all the children where that variables will be exceed



Take input in Login page



And then use it in Profile page



1. in context js🡪 create context then jsx 🡪 file use context like wrap things inside usercontext inside this user will be take that is globle variable 🡪 here we use userContext.Provider and give value={{}}
2. then in app.jsx main file we want this we wrap all pages inside the userContextaProvide all the pages are inside this who use the global variables

///////////////////////this 3 and 4 done by using useContext////////////////////////////

const {user}=useContext(UserContext)

1. then in login user.username and password will be inserted inside setUser
2. in profile user.username will be get from global variable User