React Concepts

Props rendering mtlb ye k parent component m children components ko render krna ab ye children components aap us parent component ko as a props b pass kr skty hn jis me function hu skta h array, objects ya any component usko phr hm aagy parent me as a children used kr ly gy ya rende

Hoc--> ye basically abstract concepts pe kaam krta h mtlb ek chz bnaao aur ko aagy kaai chzo me use kr lo

Mtlb ek component bnaya h style name ka jis me functionality define ki gaai h style krna ab ye function m b kr skty hn ya us component m as a simple component ya ek function bna kr us me component hu skta h h jo style ko define krega

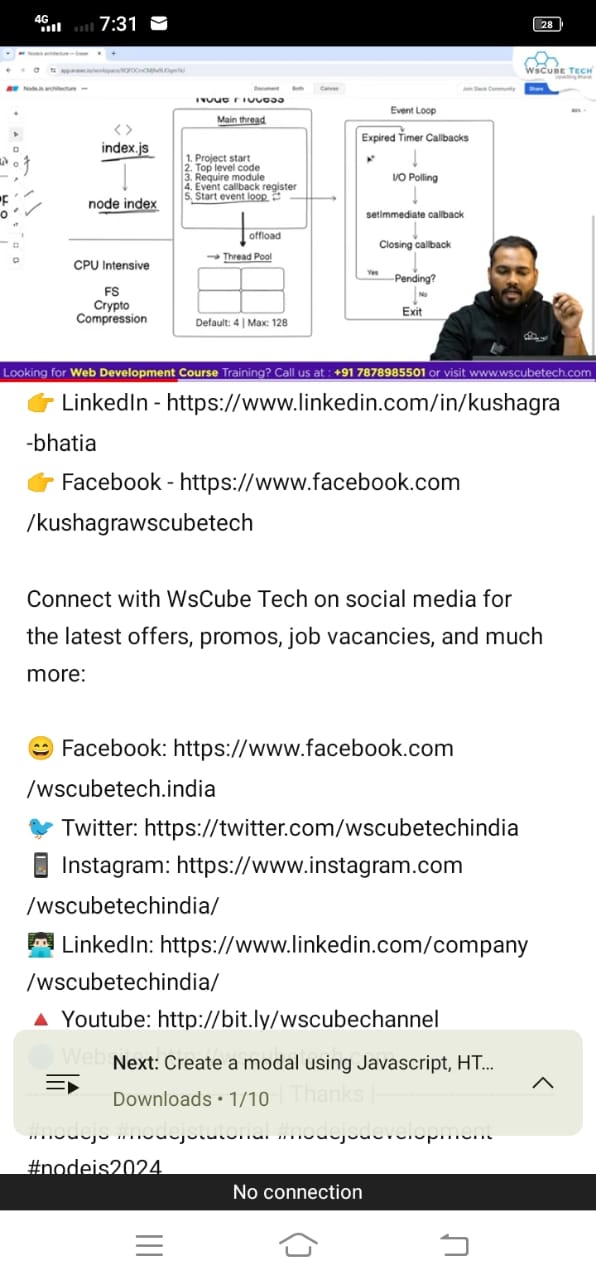
Aur phr es component ko hm different component me use kr skty hn

Mtlb k ek component define kro aur usko kisi aur component me pass kr do overall k HOC ek component as a input leta h jo k modified krta h mtlb wo input component define huta modified kliy aur phr ek component return krta h as a output ab ye simple hm function k through b kr skty hn, array ya aur b ways hu skty hn krny k

Prop drilling mtlb k most parent component se props ko pass krna child component m aur us me se phr aagy usi prop ko us k child me pass krna and so on aur es ki jaga hm provider context b use krty hn aur ye sb tb use krty hn jb hmy ek values ko multiple components m use krna hu but jo limited hu

Container presentation me view logic likha jata mtlb k jin components ko render krvana huta h unko likha jata h aur un components ka business logic separate likha jata h esi trh ye b ek design pattern ,opr waly b sb design pattern hn,     Hooks b design pattern hn

Node js

Nodejs is a runtime environment mean to use CPU related tasks in our project not just like js which runs only on browser to use the browser api's and other browser related tasks=>v8+LibUV library ka combination kr k bnaya gaya h aur v8 c++ me likha gaya jo k js k code execute krta h aur LibUV basically is designed to implement the concepts of threadpool and eventloop

**Backend**

To solve the problem of CORS

1-White listening we can set on server like appwrite or vercel to run the application locally as running on our local system as i.e. localhost

2-Proxy port--> most important coz in which we assign a localhost address to the variable and then use that variable in the URL as

api=https://localhost:3000

Now in API

"/api/jokes/" now here api is basically taking the address of api variable so on any server it will be used as proxy if there is no port available

So we can also solve this problem by proxy pkg and we also have to care about through which way we have created app i e. Either through create react or vite so for this we have to see the procedure to use them accordingly i.e. in vite

In config file we can create

server:{

proxy:{

"/api": https: localhost

}} So this not only will be appended but server will also consider that request is coming from this URL

But in production we will change this URL on deployment server remember this thing if we will deploy with this local it will show error

Also don't serve the static files from the front end to the backend after build coz whenever we will change in front it will not show any changes on front and also everytime we have to re build dist and added to the backend so avoid this practice

Moon modeler and eraser.io for creating the fields to store the data ,mongoose to make the data models and there is also a tool prisma

We will do this by code sandbox and also by GitHub codebase,also we can use stacklist which can manage everything cloudly through these we can practice our data modeling fields so that we can get ready made models for our project ,

When we create the models schema we usually don't store the buffers like images,pdf etc we use third party server or our own server to store them and then use the references of that server or URL i.e.cloudnary to store them and then we use/call  them.

We also make mini schemas for our schema whenever required , instead of creating new mini schema we can also  create new objects and keep them in array of that particular schema field,there are different scenarios for creating the mini schemas ,we also use enum when we have different status like ordered, process, delivered etc enum are by default required so no need to write required and also we can give default value separately property in the field

**Cocnepts behind Algebra**: Analytical method is also called symbolic computation and explore it

We can do operation on scalar with vector ,with each element, with vector to vector,compare element to element, vector to vector, but matrix not to other matrx but itslef, sigmoid don't work with list but with array and with each element to compute

**Assembly Language:**Direct b kr skty hn but ram k through b data load kr skty hn in assembly language,ram me srf read ya write kr skty hn with register at a time but nit both especially in case of addresses, but can be done both only on ram and similar on register, there's are different register with specific names having bits capacity like 16 bits or 18,jb data load huta h tu hr value ko address assign huta h jo k base se start huta h es address ko change b kr skty hn with origin instruction k saath aur es trh address sb instructions ko from low addresses to high addresses assign huty jaaty hn i.e. 001A01,00b02 etc es m jb ram m load huty hn tu address ko most endian se least endian me change kr diya jata h but in register its reverse, since register if is 16 bits can have so it will only accept 16 bits though we pass 1000 bits etc.and on each address how many bytes mean value after converting into decimal, hexadecimal,or binary have this is the whole concept behind assembly language

Aur ram m mostly bytes me value huti h in the form of decimal or hexadecimal, ram m different segments huty hn i.e. code , data,heap,stack etc segments, hr segment ka apna ek logical address huta h aur us me agr ek 0 laga dy tu wo physical address bn jaay ga aur phr window ko jesy b move kre jis segment m us me us physical address ko kisi b way se bna kr access kr skty hn, since register 16 bits ka huta h aur es m bytes ko le jana difficult h so es k parts kr lety hn AH higher ax and AL lower ax es m alg alg bits load kiy ja skty hn ye intel k architecture m define hn

**Cross compilation process:** we do on physical machine but for some other machine

Intel(physical machine)-->ubuntu(linux)-->tiny emulator a virtual machine which is risc file base-->linux ubuntu os installed on tiny emulator VM aur yaha phr mount kr liya us physical machine k folder ko es trh ab us physical machine k code ko run kr skty hn

Set up for this process is k compilation toolkit install ki aur code ko compile kiya aur link kiya aur phr emulator install kiya aur phr esko build kiya es kliy libraries install ki aur phr make install command run ki ,discimage install kiya q k hmy VM k opr os install krna h bht hm chaty h k hmy pre installed os mily so wo es me already huta h that's why we do this aur phr me ne es pe ja kr VM jo new bnaai h wo es diskimage pe install kiya aur run kiya aur phr yaha VM m ja kr jo k risc based h apny physical machine k folder ko access kr k run kiya file ko

**Ssl certificate from vrisign and other companies** which offer same things and sub companies which take certificate fromthese firms and then sold to others certificate

**Github concepts**

Remote pe repository ko origin kehty hn aur us me phr branches wgra bnaai jaati hn ,hr project me bht c new branches bnaai ja skti hn jiska mtlb h new mtlb k jb b new branch bnaai jaay gi wo us project se separate hugi aur jb wo ready hu jaay gi tu phr usko main branch k saath merge kr k project m wo feature add hu jaay ga ye tha branch bnany ka mqsad,merge branch ka mtlb h k jo new branch me files thi aur mtlb k wo jo feature tha es feature ka mtlb h new files for specific functionality usko main branch me transfer kr dena us new branch ki all files ko aur rebase ka mtlb h ek branch se dosri branch m complete work ko transfer krny kliy rebase kiya jata h,

Stash: ka concept ye h k agr ek branch ki file me changes ki jaay aur usko ataged b kr diya but commit ni kiya aur dosri branch me switch kr k kaam krny ag jaay tu phr ye error dy ga k ya tu phr usko commit kre ya phr stash kre mtkb k usko commit kiy bgair save kre but issue ye h k ye data lost kr dy ga us file me se tu usko retrieve krny kliy git stash pop command run kre gy tu wo data b aa jaay ga aur phr usko commit wgra kr dy esi trh stash ki aur b commands hn wo b dekh skty hn aur eska b bht use uta

Stage ka mtlb h k project level pe jo b changes wgra ki jaay chaay phr wo file add ki jaay ya kisi file m changes ki jaay usko track krny kliy ya commit phase me le aany kliy stage area m add kiya jata h

Tag ka mtlb huta h k kis commit pe aapka product ka stable version tha es liy jis commit pe lagy k product stable h tu phr usko tag kr dy us commit ko aur ye sb local repository pe huta h aur agr remote repository pe koi changes ki jaay tu phr usko pull kr k usko local repository m laya jata h phr us ko staging ki jaati h aur phr push kr diya jata h

Es me card ka b concept huta h jiska mtlb h k agr ek name k multiple files hu aur srf un specific files ko he stage krna hu tu phr un k frst same characters ko le aur saath\* laga kr add kr dy

**CSS concepts:** Do center div 3 ways

1-Display:foex

Align-items:center// do horizontally center

Justify-content:center// do vertically center

2-

Display:grid

Place-items:center

3-parent ko position relative dy

Child ko position: absolute

Left, right,too,bottom b dy gy

Ya phr transform property laga kr b set kr skty hn

Px it is absolute unit mean fix takes from parent

%it takes from browser or its parent

Em it works on fontsize according to parent font size

Rem takes from browser or root font size not from parent

Vh/vw takes from browser ,vmax,vmin same