React

Single page application:SPA always ek hi page pay data render hora hta hay aur wo refresh nhi hta oska

Url change hta hy or data chnge hta hay

Map: In JavaScript, map() is a method of the Array object. It creates a new array by calling a function on every element of the original array and storing the results in a new array.

React create user interfaces from components

Components means web ky tokra krna

Promises:there are types 1)pending2)fulfiled3)error

Pending :the result is undefined

resolve:the result is a value (fulfilled)

reject:the result is an error object

promise.then((res)=>{..})

promise.catch((err)=>{..})

Fetch Api :the fetch api provides an interface for fetching(sending/reciving) resources.

It use request and response objects

API:stand for application programing interface

stt=

ASYNC-Await

Async function always return a promise

200 status means data successful aya hay

Let promise=fetch(url[option] )only url du tou aesy get req kehty ha ye kam krta data recive krny ky liye

Json data denay ka format hay

AJAX : is ayschronus js

JSON:is javascript object notation

Json()Method: returns a second promise that resolve with the result of parsing the response body text as json.(inpu as json , output is js object)

Json method sec promise return krta hay

400 status menas bad req and 404 req means page not found

Props:passing data between parent and child

Props object return hay wo wala jo apny custom diya hoga

Children props:kisi bhi header ky bitch ka data children khlata hay

Usestate ma jo value dete ha wo defalaut val hti ha

Setcount ma update val dete ha

Ctrl . suggestion

Material ui

Use effect: render function ky bd kiya karna hay ye bata hay

Push is random id generate

Ddynamic routing

Var : var ma val re-render nhi hoti jiski waja say ui par purani val dekhti hay

Difference both

State:state ma value re-render hti hay jiski waja say ui pr purani value nhi dejhye deti

useEffect:

initializing

updation

unmountain

assginmen

login signup :page

home//emmoerce products dami card

ecommerce

products upload sary feeds make

difference between functional component and class component

react ma jo bhi likthy hay wahi render hota hay

useeffect:its work is after rener component what work will do its work is perform useeffect

[01:17, 20/07/2024] +92 319 7963330: Asynchronous

1 set interval

2 set time out

3 fetch

4 fire base Ke jitne function ha wo Asynchronous function ha

Asynchronous Ka response synchronous code Ke baad chalta ha

.then ya .catch Laga ho ya asyn await laga ho to wo Asynchronous function ha ya

[01:17, 20/07/2024] +92 319 7963330: Jab hum function main .then .catch Chala rhe the to wo navigate phele chal rha th kiunke JavaScript asynchronous Ka intezaar nahi krta.

Firebase wale function main se .then .catch hataya ha or wo login wale button pr call krdiya th

Usse bhi acha ye kaam ha .then .catch hata do

Phir humne asyn lagaya login Ke function se phele

Phir await lagaya.or in cheezon ko try Ke andar band krdiya or catch Chala kr error handle krlia

21/7/2005

Firestore data base:addt data firebase data

2eham cheze firestore database m hti hay (collection,documentation)

A database

Basic Topic its concept and uses must be clear

1)State vs variable

2)Props

3)Useeffect 3 stages

4)Conditional rendering

5)Async await

6)array maping

Useeffect :3stages

1)initialization:”component ky start hty hi koi kam krwana ho”

For example: api fetching or any logical wok

2)Updation:”jab bhi state ya props ky update pay koi kam krwana ho”

For example: jab count(state) ki value 5 hoaji tu ek alert dekha dena

3)unmount:”jab component khtam hony wala hn tu koi kam krwana ho”

For e.g:koi listener khtm krwana ho

Map or for loop ek hi kam krty h but es liye use krty h wo return krta hay

[00:37, 24/09/2024] Dilawer 🎓: Asynchronous or synchronous programming ka interview ke perspective se samajhna zaroori hai, kyun ke yeh fundamental concepts hain, jo software development mein kaafi use hotay hain.

Synchronous Programming:

Is mein tasks ek ke baad ek chalti hain, aur agla task tab tak start nahi hota jab tak pehla task khatam na ho jaye.

Yeh ek blocking approach hai, jahan ek function doosray function ko wait karne par majboor karta hai.

[00:43, 24/09/2024] Dilawer 🎓: Promises asynchronous tasks ko handle karti hain. Yeh ya to fulfill hoti hain (resolve) ya fail hoti hain (reject).

[00:44, 24/09/2024] Dilawer 🎓: Network requests: Data fetch ya send karna (HTTP requests, WebSockets).

Timers: Delayed actions (setTimeout, setInterval).

User interactions: Event listeners (click, keypress, etc.).

File operations: Reading/writing files (Node.js, Web APIs).

Database queries: Data fetch ya save karna (SQL, NoSQL databases).

Web workers: Heavy computations ko background mein run karna.

UI updates: Animations, requestAnimationFrame.