React Course — Thapa Technical

Starting Date — Sat, July 22, 2023

Lacture 1.

What is react?

React is a Javascript library for building user interfaces. It is not a framework. It is also known as react ds or react.js

React knows only one thing that is to create awsome UI.

React was first developed by Jordan walke, a software engineer on facebook. It was first deployed for facebook Newsfeed around 2021

In 2013 Reat was open sourced at ds conference.

About React

Component Baied Approach

Every App we will build using react will be make up of pieces called components.

Component is very important for reuse.

DOM update are handled very gracefully

Lacture 2.

Prerequisites for react

Basic knowledge for html, css & Javascript

Basic knowledge for ES6 features.

Lacture 3

Download VS Code

Lacture 4

Download React Environment

Lacture 5

React folder structure + first Hello World program

(1) node_modules folder

It is the repository of modules/library which you are using inside your project. What ever you are importing in your project that module or library should present inside the node_module folder. When you do npm install that time that module or library intail inside the node_module folder and one entry inside package.json file.

2) Package.json

React App meta data.

③ index.js (most important in which we write react code

④ index.css

⑤ index.html

Bavi Jo be hai onka as a begineer zarorat nai hai.

In b index.html there is jut

<body> <noscript> You need to enable javascript to run this app </script>

// noscript is used if browser is not // supporting javascript then this text // shows

</body>

Esky elawa our zarorat nai index.html mn

Index.js important

// in react we need to import // two modules that we need must

var React = require('react');

var ReactDOM = required('react-dom');

// in react dom module we had one
// function that is render() it means
// dahkana
// It take three parameter
// ReactDOM.render('kya dahkana hai', 'kahan dahkana hai', 'callback');
optional

Important

In react yahn kya dahkana hai ke jagh hum component all karty hai. Wo ksy bnaty hai ksy bnaty hain hum bad mn paramgy abhi hum start mn direct tags likh kar use kar lyngy.

ReactDOM.render(<h1> Hello world </h1>, document.getElementById('root'));

//<h1> here looks like html tag
// But it is not html tag It
// is JSX that we will
// study more in next lecture

Babel and webpack

Babel is a javascript compiler that convert modern javascript (ECMA 6) into a old javascript that browser can understand
We dont need to install it when we import 'react' module we automatically get it.

Lecture 6.

JSX in React Js

// We can import module using one
// way that we study in prev video
// and more in advanced way given below
import React from "react";
import ReactDom from "react-dom";
ReactDOM.render(<h1> Hello </h1>,
document.getElementById('root'));

That <h1> tag is not html tag its JSX syntax.

To used JSX here importing 'react' module is compulsory

Babel is converting this JSX syntax into the javascript that browsers can understand The babel will convert it into (the previous h1 tag (code)) as

function provided by react module

ReactDOM.render(React.createElement("h1", null, "Hello"), document.getElementById('root'));

These are the two ways by which we can print Hello world. If we want to used or print it by pure javascript then

```
var h1 = document.createElement('h1');
  h1.innerHtml = "Hello world";
var id = daument.getElementById('root');
  id.appendChild('h1');
```

Lecture 7

How to render JSX Multiple elements inside ReatDOM.render()

render() take just single element so if we want to add multip we need to wrapped it JSX element into single element like

reactDOM.render (<h1> Hi </h1> <P> Azan </P> <h3> Pakistan </h3> , document. getElementById('root'));

If we are using react version above 16 return It can array of n element so we can do something like this

reactDOM.render ([<h1> Hi </h1>, <P> Hello </P>,], document.getElementById('root'));

Lecture 8)

Understanding React Fragment

1. When we used div we need add one extra element div to add multiple element.

So far that we need React.Fragment to add multiple elements

```
ReactDOM.render(
  <React.Fragment>
    <h1> Hi </h1>
    <P> HeHo </P>
  </React.Fragment>, document.getElementById('root') );
```

We can used it as shortcut like

```
ReactDOM.render (
  <>
    <h1> Hi </h1>
    <P> Hello </P>
  </>, document.getElementById('root')
);
```

Lacture 9)

Challenge 1

lacture 10)

Javascript Expression in JSX. in React Js

```
impost React from 'react';
impost ReactDOM from 'react-dom';
const fname = 'Azan imtiaz';
reactDOM.render(<h1> my name is {fname} </h1>, ...
..)
```

Output: my name is fname

```
reactDOM.render(<h1> my name is {fname} </h1>, ...)
```

Output: my name is Azan imtiaz

```
reactDOM.render <h1> 2+3 </h1>, ...)
```

Output: 2+3

```
reactDOM.render(<h1> {2+5} </h1>,
...D;
```

output: 5

Summary

If we want to used Javascript inside JSX we used { } Brackets in these curly brackets we can

write only Javascript expression Not Javascript statements (like conditional) If else etc

Lacture 11)

Template literal in JSX in react JS

```
const firstname = "azan";
const last_name = "imtioz";
```

If we want to produced

Output : my name is azan imtioz.

We can do it without template literals like

```
<h1> my name is {firstname} {lastname}
</h1>
```

Or like

```
<h1> my name is {firstname + " "+
lastname} </h1>
```

With template literals

```
<h1> my name is {` my
name is ${firstname} ${lastname} `}
</h1>
```

lacture 12) Challenge

lacture 13)

Html attributes and JSX attributes

Jsy hum html tag/element py attribute add karty hain wsy he JSX ad mn be karty hain But Esmy attributes camel case mn add karna hota hai mean agr ek attribute hai jismy 2 word hai to pahla chota 2say ka pahla word bara likhna pary ga like In JSX

<h1 contentEditable="true"> Hi </h1>

in html

<h1> contenteditable="true"> Hi </h1>

In JSX

Mini project. Add three image from which one is clickable. Add src with Javascript variable

import React from "react";
import reactDOM from "react-dom";

```
const img1 = "...";
const img2 = "...";  const img3 = "...";

reactDOM.render(<>
  <h1> My image </h1>
  <img src={img1} alt="random img" />
  <img src={img2} />
  <img src={img3} />
  </>,
  document.getElementById('root') );
```

Lacture 14)

Css Styling and importing Css Files in React Js

```
// In Css we was using class
// but in react its
// a keyword so we used className
// in attribute

style.css

* { Padding: 0px; margin: 0px; }
```

.changeColor {
text-align: center; color: Yellow;
font-weight: bold; }

Index.js

import React From 'react';
import ReactDOM from 'react-dom';
import './style.css';
ReactDOM.render (<>
<h1 className="changeColor"> My
Name is Azan </h1>
<P className="changeColor"> Hi i
m engineer </P>
</>, document.getElementById('root'));

Lecture 15)

How to used google fonts in React ds
Google font ka link index.html
ky head tag mn rkna hai.
And usKi css style.css mn
he add krni hai

Lecture 16)

Inline Css and internal Css in react

Abhi tak hummy external Css kuzna seekha hai jismy hum import karty hain css file ko Now lets study internel and inline Css in react

Inline css thora interesting hai Edx First thing do Css ke properties hoti hai usmy kabab case use hota hai like
text-transform capitilize;
But Yahan py Camble case use hoga inline Css karty hoy
textTransform: 'capitilize';

Second thin Yahan inline Css javascript ky through as a object pass karty hai like

```
const heading = {
    color: 'white',
    fontSize: '16px',
    textAlign: 'center'
}
```

<h1 style={{color:'white', fontSize:'16px', textAlign:'center'}}> Hi </h1>

style mn para object

But aesy humy [illegible] likhna parta hai esy achy

[illegible] hai jo pehly object bnanya hai usko

pass kary dy its simple and look

[illegible] and more

good

simple way hai ky para

object name pass karda

<h1 style={heading}> Hi </h1>

ya pehly object esi liye bnaye

tha

lecture 17)

Mini Project

lecture 18) Important

< React components in hindi >

In react component ke file ka first alphabet bara hota hai and sth mn extension .jsx hoti hai taky react ko pata chaly jsx hai-

Make new file with any name for component

- Heading.jsx

```
import React from 'react';
function Heading(){
  return ( <h1> Azan kahun </h1> );
// Above brackets are not necessary
// for one element but when we
// had more than one element we
// need to put it
}
export default Heading;
```

ya name kuch be likh skty hain.

Index.ds

```
import React from 'react';
import ReactDOM from 'react-dom';
import Heading from './Heading';
ReactDOM.render(
  <Heading> </Heading>
// Because it is not sorry because
// we know if there is nothing
// to write inside element we
// can write it as
```

<Heading />,
document.getElementById('root'));

Es tara hum different component bna ky esmy add kar skty hain But Best practise hum dakhty hain ky index.js mn kabhi be bot zyada component nazar aa nai aty bulky sirf App component bna ky usmy sb dosry component add karky app component ko index.js mn dalnagy.

lets make one more componet

Para.dsx

```
import React from 'react';
function Para(){
    return <P> I m SE </P>;
}
export default Para;
```

Now lets make App.dsx

```
import React from 'react';
import Heading from './Heading';
import Para from './Para';
```

```
function App() {
  return (
    <Heading />
    <Para />
  );
}

export default App;
```

index.js

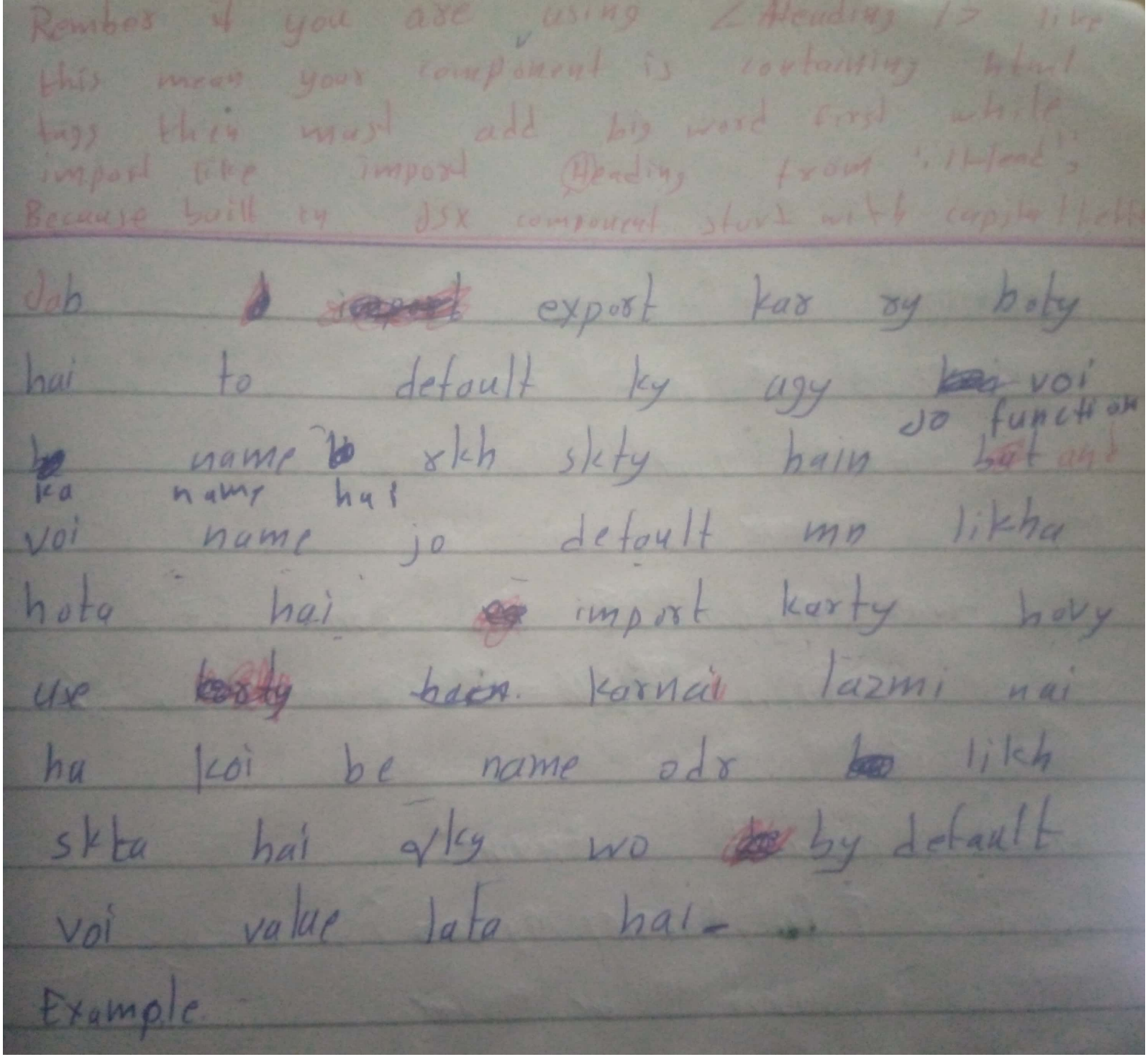
```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App';

ReactDOM.render( <App />,
document.getElementById('root') );
```

Lecture 19)

Challenge on Components

Lecture 20)

ES6 modules import export in React JS


Remember if you are using <Heading /> like this mean your component is containing html tags this must add big word first while import like import Heading from './Heading'; Because built by JSX component start with capital letter

Jab export kar sy boty hai to default ky agy koi jo function ka name likh skty hain but and koi name jo default mn likha hota hai import karty hovy use karty hain. Karnai lazmi nai ha koi be name odr likh skta hai q/ky wo by default koi value lata hai-...

Example

Example

App.ds

```
const name = "Azan";
export default n;
```

Index.ds

```
import [redacted]name from './App';
ReactDOM.render ( {[redacted]name} .. );
```

Ab mind mn abra boga
o/ky by default hum ek
he variuble pass kar skty
hai to agr 2 ya usky
zyado variable hon phir ken
krungy. And agr sth function be
kanny hon to

App.ds

```
const firstname = "Azan";
const LastName = "Imtiaz";

function MyJob() { let job = "SE";
    return Jobs
}
export default firstname;
export { lastname, MyJob }; // Esmy
//name must be same
```

Index.ds

```
import firstname, {lastname, MyJob} from
'./App.ds';

ReactDOM.render( <>
  <ol>
    <li> {firstname} </li>
    <li> {lastname} </li>
    <li> {MyJob()} </li>
  </ol> </>,
  document.getElementById('root') );
```

In shortcut we can do

In index.js

import * as Azan from

ReactDOM.render(<>

 {Azan.default}

 {Azan.lastname}

 {Azan.MyJob()}

<h1> document.getElementById

);

Lacture 21

Challenge of import and expor

Build calculator

Calc.jsx

import React from 'react';

function add(a,b){

return a+b; }

function sub(a,b){

return a-b; }

function mul(a,b){

return a*b; }

```
function div(a,b){
    let d = a-b;
    d = d.fixed(2);
    return d;
}

export {add,sub,mini,div};
```

index.ds

```
import React from 'react';
import ReactDOM from 'react-dom';
import {add,sub,mini,div} from './calc';
ReactDOM.render( <>
  <ol>
    <li> {add(a,4)} </li>
    <li> {sub(1,9)} </li>
    <li> {min(2,3)} </li>
    <li> {div(9,11)} </li>
  </ol>
  </>,
document.getElementById('root') );
```

Mini Project from video 14

In it we learn how to use IF and else in react js

index.js

```
import React from 'react'
import ReactDOM from 'react-dom';
let greeting=''; let cssStyle= { }
let curdate= new Date();
curdate = curdate.getHours();
If ((curdate >=1 && curdate <=12){
greeting="Good Morning"; cssStyle.color="red";
}
else{
greeting = "good Afternoon"; cssStyle.color="blue"
}
```

ReactDOM.render (<h1> Hello, sir {greeting} </h1>, document.getElementById

<Remember>

Agr return koi html tag
nai horu component mn
to dosri file mn import
karty hovy koi be name
rkh skty hain but agr

component me html tag jo
JSX return hota hai
diya humne <Heading /> ek
tag ka use karna hai to
eska import time pe first
alphabet compulsory aana hai

Lacture 22)

Props in react

Props is used to make components flexible In React prop short for properties it is a fundamental concept used to pass data from a parent to a child component. Props are like function argument that are provided to react components allowing them to customize and flexible When you define a React Component, you can pass data to it as props making it easy to reuse component with different data. Props are read only mean that a component cannot modify the prop

It recieve they can only be used to render ~~components~~ content or pass data down to child components

let Make our Card component

CardComponents.jsx

CardComponents.jsx

```
import React from 'react';
function Card(Props){
return ( <>
<div className="container">
<div className="image">
<img src={Props.imageName} alt="
my pic" /> </div>
<div className="title">
<h3> {Props.title} </h3>
</div>
</div> </> );
}
export default Cards;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import CardComp from './CardComponent';
```

```
ReactDOM.render ( <>
    <CardComp   imageName=" . . .   "
     title=" . . . . . "   />
    <CardComp   imageName=" . . .  "
     title=" . . . . . "   />
    <CardComp   imageName=" . . . "
     title=" . . . . . "   />
    </>,
     document.getElementById('root')
 );
```

// sary custom attribute as a object wahan pass ho jatty hain jinny hum . operator sy access karty hai jsy hum pracly CardComponent main kra hai.

Lecture 23

Array in React

1) Rendering a list

```
import React from "react";
const MyListComponent = () => {
  const items = ["item 1", "item 2",
                 "item 2", "item 3"];
```

```
return ( <>
<ul>
{ items.map( (item, index) => (
<li key={index}> {item} </li>
))}
</ul>
</>
);
}
export default MylistComponent;
```

(2) Working with array object

```
import React from 'react';
function Component() {
const data = [
{ id:1, name: "Arun" },
{ id:2, name: "Jaya" },
{ id:3, name "Alice" },
];

return ( <>
<ul>
{data.map( (item) => {
<li key={item.id}> {item.name} </li>
});}
```

export default component

③ State Management with array
we will study later

Lacture 25.

Netflix completion main points

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import Card from "./Cards";
import $data from "./data"; // it is the file that contain array of object

function mcard(val){
  return(
    <Card
      imgsrc={val.imgsrc}
      title={val.title}
      sname={val.sname}
      link={val.link}
    />
  );
}
```

```
ReactDOM.render (
    <>
    {$data.map(ncard)}
    </>
                    );
```

If we want to do it without ncard function

```
ReactDOM.render (
    <>
{ $data.map(function ncard(val){
return(
    <Card
        imgsrc={val.imgsrc}
        title={val.title}
        sname= {val.sname}
        link = {val.link}
        />
    );
})}
    </>
                    );
```

Video: 25

React Developer tools

Debugging and error handling in React

One important thing it is very important to add one unique key in child in a list.
Mean must add one unique key in each component.
JSX directly inside map() calls always need keys
key ~~prop~~ is mandatory while using map

React developer tool jo ky console ky pass hota hai unny ek component ka option hota hai jiscy hum apny sary component and onky undar ke properties ko dhek skty hain

Lecture 26

If Else statement in React

(Exp1)

```
import React from 'react';
const MyComponent = (prop) => {
    if(prop.isLoggedIn){
        return <div> Welcome, user </div>;
    }
    else{
        return <div> Please log in to
                continue </div>;
    }
};
export default MyComponent;
```

(Exp2)

```
import React from 'react';
import ReactDOM from 'react-dom';
import Netflix from '...';
import Amazon from '...';

const favSeries = 'amazon';
const favS = () => {
    if(favSeries == 'amazon'){
        return <Amazon/>;}
    else{
```

```
return <Netflix />;
}
};
```

```
ReactDOM.render(<><h1> Hi, Hello </h1>
<FavS /> </>, document.getElementById('root'));
```

Lecture 27

Ternary Operator

Syntax

```
condition ? exp if true : exp if false;
```

```
favSeries == 'amazon' ? <Amazon /> : <Netflix />;
```

Lecture 28

Slot machine in React js

Lecture 29

How to add emoji in Vs Code

Important

Lecture 30

Hooks in React Js

16.8 version of react Js introduced it

It allows you to use state and other react features without writing a class. Hooks are functions which "hook into" React state and lifecycle features from function components

It does not work inside classes. It is basically made so we can use state in functional components.

To use hook Node version must be 6 or above and NPM versio 5.2 or above

Before studying hooks lets study what was issue and why we need hooks. (Remember this will show just one use of hook). There are many more

Ok lets try to make one

project One count in h1 and one button when we click on button count get increament and render

.import - - -

let count = 0;

function increament() {

count ++;

console.log(count);

}

ReactDOM.render(<> <h1> {count} </h1>

<button onClick={increament}> Print

</button>, document. . . . .);

// console mn increamented count he show hoga but h1 mn hmesha count 0 he show hoga To ya issue that pahly hum function leg through nai kar skty thy aesa hmy class component use karny pdty thy now we can do it with hooks in react Js

There are many we of hook
we will study one here

(1) useState
It allows you to add state
to functional component. usestate
function return an array of
state variable at first index
and at second index function to
update that state. for example
import React, {useState} from
'react';

function Counter(){

// array destructuring state variable function to change state
const [count1, setCount] = useState(0); initial value

const increament = () => {
setCount(count++); }

return (<> <h1> {count}
</h1> <button onClick={increament}>
</button> </> ;
);

3;

its just an use we will see many use like useEffect, UseContext, UseReducer in later lactures-

Lacture 31 Challenge of hook

```
import React, {useState} from 'react';

function AppComponent() {
  let newTime = new Date().toLocaleTimeSt

const [ctime, setCtime] = useState(newTime);
const UpdateTime = () => {
  newTime = new Date().toLocaleTimeString();
  setCtime(newTime);
  };

return (
    <>
    <h1> {ctime} </h1>
    <button onClick={updateTime}> Print
    </button>
    </>
  );
};
```

export default App;

Lecture 32- Mini project
Create Digital Clock

import . . . ;

function App(){
const [time, setTime] = useState(new Date());
functe updateTime(){ setInterval(()=>{
setTime(new Date()); }, 1000);
}

return (
<>
<h1>{date.toLocaleTimeString()}</h1>
<button onClick={updatetime}>
Print </button>
</>
);
}

Lacture 33
Handling Events in React Js
We have used some events during learning hook but we had not study event in detail so

lets discuss it

In Javascript we use onload but in react we used camel case and for calling function we didn't give () those parenthesis why ya agr lagay gy ! to react wont we directly call ho jatta hai like

function alertPoint () {
  alert("Hi"); }

ReactDOM.render(<> <button onClick = {alertPoint} > "Click Me </button>
<I>, document.-----);

lets Make project whers user click on button background Color change

import React, {useState} from 'react';
const App=()=>{
let colos = "#8e44ad";

const [bg, SetBg] = useState(colos)

```
const bgChange = () => {
  let newBg = "#3498db";
  setBg(newBg);
};

return (
  <>
    <button onClick={bgChange}>
      Click Me
    </button>
  </>
);
```

Lecture 34.

Form in react js - React

Controlled vs uncontrolled Component

Uncontrolled Component

Uncontrolled component is the form element whose state is handled by the DOM. rather than being managed by React state. In this case input element maintain its state internally and you rely on DOM method to retrieve its value when needed

```
import React from 'react';
```

```
const SimpleForm = () => {
    const handleSubmit = (event) => {
        event.preventDefault();
const inputElement = document.getElementById('inp');
const inputValue = inputElement.value;
const modifiedValue = inputValue.toUpperCase();
console.log("Modified Value", modified Value);
};

return (
    <form onSubmit={handleSubmit}>
    <div>
        <label htmlFor="inp">Input: </label>
        <input type="text" id="inp" />
    </div>
        <button type="Submit">
            Submit </button>
        </form>
    );
};
export Default Simpleform;
```

Controlled Component

A controlled component is a form element whose value is controlled by React state. In other words the state of the input Element is set explicitly through value Prop. When the use interact with input the state is updated with new value & re render to reflect value in the input

```
const ContactForm = () => {
  const [name, setName] = useState('Hi');
  const handleNameChange = (event) => {
    setName(event.target.value);
  };

return ( <>
  <div>
    <input type="text" id="name"
    value={name}
    onChange={handleNameChange} />
  </div>  </> );
};
```

Watch thapa technical lacture 34
If i forget value, defaut value,
onClick Event and onChange Event
etc.

Lacture 35)

Form Submit in React ds

onsubmit in html attribute allow you to specify ds function that will be executed when form submitted. This attribute is used to handle form submission events

In React we used it as onSubmit prop.

Remember

preventDefault() is used to avoid default behaviors in ds When call for in. an event handler it tells browser

not to perform that default action for that event

In context of form submission event.preventDefault is commonly used to prevent the default behaviour of a form submission initiating by clicking submit Button

```
import React {useState} from 'react';

function App(){ const [fullName, setFullName] =
useState();

const [name, updateName] = useState("");

function whenSubmit(e){
e.preventDefault();
setFullName(name);
};

const inputEvent = (event) => {
set updateName(event.target.value);
};

return ( <> <form onSubmit={whenSubmit}>
<h1> {fullName} </h1>
<input type="text" defaultValue={fullName}
placeholder="Hi - Enter Name" />
```

button

<[redacted] type="submit" /></button>

</>, document.getElementById('root')

);

Lacture 36)

Handling Complex Multiple input form State in react js

In previous way in which we was submitting data was long If element in form increases we have to make different fate to maintain each input. But if we want to make it just once and that work for all inputs so it a better and more usable way So first lets remind that we can give name to each element in html

```
mport React, {useState} from 'react';
```

```
function App{
const [UserName, UpdateName] =
useState( ()=>{
  FName:"",  LName:""  });

const inputEvent = (event) => {
  const value= event.target.value;
  const name = event.target.name;
  // we call use callback function in set func
  updateName( (preValue) =>{
    IF (name ===  "fName") {

return {  fName: value, lName: preValue.lname};
}
else  If(name === "lName") {
    return {
        fName = PreValue.fName
        lname : value
        };
    }
  });
};

function Submit(e) {
  e.preventDefault();   alert("submit");
}
```

```
return ( <>
    <div>
        <h1> Hello {userName.fname}
{userName.lName}  </h1>
    </div>
    <form  onSubmit={submit} >
<input type="text"  name="fName"
value={userName.fName}  onChange="input
Event()" />

<input type="text"  name="lName"
    value={userName.Lname}  onChange="input
    Event()" />

<button type="submit" > Submit </button>
```

In a more concise way we can use spread operator to make shallow copy and then update and return like

```
const [illegible] = function (event) {

const {name, value} = event.target;
function [illegible] updateName ( function (prevFormData){
    return {
        ...prevFormData, [name]: value, };
    });
};
```

Lacture 37

Nothing Just one easy challenge

Lacture 38

Spread Operator in React Js

We already know about Spread Operator in Javascript

Lacture 39

Here we study

Lecture 40 :-

Thapa Technical

Create Project Tudo List

New things we have learned

How to create an array a and update its data using the useState hook

```
Function MyComponent(){
    const [myArray, setMyArray] = useState([]);

//Example function to add data to array
const addToArray = (item) => {
  setMyArray([...myArray, item]);
    };

//Example to update data in array
const updateArrayItem = (index, newItem) => {
const updateArray = [...myArray];
updateArray[index] = newItem;
setMyArray(updateArray);
};

// rendering array
return(
```

```
<div>        {myArray.map((item,index)=>{
    return <div key={index}> {item} </div>
    })}
</div>      );   }
export default MyComponent;
```

One more new thing

If we want to called function on any event with Arg we have to use anonymous function

```
<button onClick={()=>{
    calledFunc(2);   }
    }> submit </button>
```

Project

TodoList

[Add list] [+] when we click on + New list will add with value on input like If Apple added to inp and + is clicked

Apple x

if Mango also added

Mango x

If user click x cancel that list will be removed

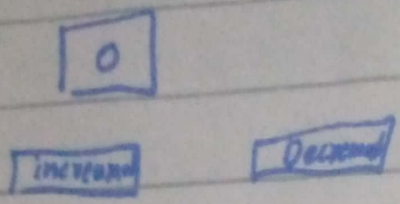
let make App.jsx

```
import React {useState} from 'react';

function Ap(){
[redacted]
const [arr, updateArr] = useState([]);

function change(){
  let value = document.getElementById("inp").value;
  updateArr([...arr, value]);
}

function deleteFunc(i){
  updateArr((e)=>{
  return e.filter((f,v)=>{
      return v!=i; if(v!=i){return e;}
    })
  })
}

return (
  <>
  <div>
```

```
<h1> todo list </h1>          <br/>
<input type="text" id="inp"
placeholder="Add list" >  </input>

<button onClick={change}> + </button>
</br>

{ arr.map((e,index)=>{
    return( <>
<div  d style={{display:"flex"}}>
<div  style={{width:"30px"}}>
  {e}
</div>
<div>
    <button  style={{marginLeft:"40px"}}
onClick={()=>{
      deleteFunc(index)
  }}>  X </button>
</div>
</div>
  <br/>
  </>)          })
</div>  </> );
}
export default App;
```

Lacture 41.

React challenge on incrementing and Decrementing the state variable on Button Clicked




```
import React, {useState} from 'react';

function App(){
const [data, setData] = useState(0);

let inc = ()=>{
    setData((e)=>{
        return e+1; });
    }

let dec = ()=>{
    setData((e)=>{
    alert("u cannot decrement from 0");
        return 0;
    }
    return e-1; });
}
```

```
return (
  <>
    <h1> {data} </h1>
    <button onClick={inc}> Increament </button>
    <button onClick={dec}> Decreament </button>
  </>
);
}

export default App;
```

Lecture 42

How to use Material UI icons in react App

In fontawsome to we icon we had to attached link to index.html

But Material icon ko easily hum as an component import karky use karsktty hain

pahly hum npm install @material-ui/icons then do npm install @material-ui/core

Now Material icon ke website pay jaky inside of hum koi be icon ky component ko

Lacture 43 - import

The Best UI Componend for React Js.

Getting started be Meterial UI

hum just icon he import nai kar sakty bally dosry built in component like header footer etc ek Material UI sy import kar sakty hain.

Lacture 44.

NPM (Node Pakage manager) in 10 minutes → it will Make our life

(1) NPM is the worlds largest Software Registry

(2) The registry contain over 80,000 code packages.

(3) Open source developer use NPM to share software

it like code repository where all code is placed already we just Need to call it.

Npm is installed with Node.js
All npm packages are defined in files called package.json
The content of package.json must be in JSON.

In window we used Command prompt in linux CLI (command line client) Npm can manage dependies

lets Make mini project of Digital Clock

ham directly npm repository mn do code already hai usko use karky easily kar skty hain. lets search on google npm react digital clock. Open first link waha ek install link mily ga npm install react-digital-clock and paste that on command prompt in your app folder Now check dependies on. Package.json it will show "react-digital-clock": "^0.12",

Download Extension
IntelliSense for Css class name in Html

Create Accordion using React Js

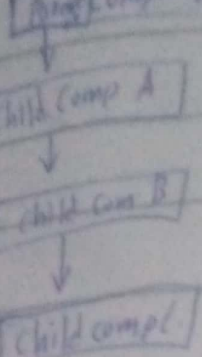
Lacture 49)
Project Mini Create website in 1 minute

Lacture 50)
React project of google keep

Lacture 51)
Context Api in React Js

React hooks provides a concept call Context.

React Context ApI allows you to easily access data at differed levels of the component tree without passing prop to every level



Pahly ye hota tha hum direct parent component ey kisi aur mn nai bahj skty thy means agr hamara Parent ey B mn data bahna hai to hum props ky through pahly A mn bahjaty phir B mn But Context Api ke waja ey hum direct bahj skty hai

We wed

createContext() to create Data

Provider to pass Data

Consumer to access Data

In App.jsx

```
import {createContext} React from 'react';
import ComA from "./ComA";

const FirstName= createContext();
const LastName = createContext();

const App = () => {
  return (
    <>
```

```
<firstName.Provider value={"Azad"}>
  <lastName.provider value={"imtiaz"}>
    <ComA />
  </lastName.provider>
</firstName.Provider>
</>
);
};
export default App;
export {firstName, lastName};
```

In ComA.jsx

```
import React from 'react';
import CardB from '../cardB';
function cardA() {
    return <CardB />;
    }
export default cardA;
```

In compB.jsx

```
import React from 'react';
import {firstName, lastName} from '../App';
```

```
const ComB = () => {
return (
  <>
    <FirstName.Consumer>
    {(fname) => {
      return (
        <LastName.Consumer>
        {(lnam) => {
return ( <h1> My nam is {fname} {lnam}
        </h1> );
        } }
        </LastName.Consumer>
    );
    }}
  </FirstName.Consumer>
);
  }
export default ComB;
```

Lacture 52)

useContext Hook in React

useContext is a built in React hook that provide a way to consume the value from a context created by context Api.

It allows functional component to to access context data without using a Consumer component.

useContext simplifies the code by avoiding the need for nested Consumers and provide a more straightforward and cleand way to access the context data in functional component

In prev Consumer code we can short it by

ComB.jsx

```
import React {useContext} from 'react';
import {firstName, lastName} from './App';
```

```
const ComB = () => {
  const fname = useContext(FirstName);
  const lname = useContext(LastName);

  return ( <h1> My name is
    {fname} {lname} </h1>
    );
}
```

Wao its so short

Lecture 5 (3)

UseEffect Hook

By using this hook, you tell React that your component needs to do something after render. React will remember the function you passed (we will refer to it as your effect) and call it later after performing the DOM updates. We set a document title but we could also perform data fetching or call some other

Hooks always work in functional component you need to define it at top of your component. It does not work in class component

Jab bhi page render ho ya update ho wahan pe us time pe useEffect bhot kam ata hai

UseEffect always / Accept a function

It take two Arg. first the function and second the optional dependencis array that control when the useEffect should executed. By default every time it is executed when render function is called due to refresh or update in Code. By providing dependencies array you can specify condition that dictate when effect should run

The dependencies array contains variables [illegible] when these variable update the effect will be executed. If the dependencies array is [] empty, the effect will run only one when first time page renders

(1) useEffect(() => {
// only run when first time render happ
}, []);

(2) No dependencies array (no second arg)
The effect run every time render function happened
useEffect(() => { });

(3) Dependencies array with variable
The effect run is whenever any of variable in the array changes
const someValue = 10;
useEffect(() => {
// This effect run when someValue changes
}, [someValue]);

Lacture "54)

Challenge of useEffect

In this challenge new thing was document.title

Lacture 55)

React Api Call to get Pokemon JSON Data using Axios and useEffect in React Js

You will learn to fetch Api in this project

So

Lacture 56) — If forgot must watch video

React Router in React Js

React Router is a popular library for handling routing in React .Js It allows you to create single page application(SPAs) with multiple view or pages without the need

for a full page refresh when navigating b/w different Components

To get started with React Router you will need to install it as a dependency in your project.
Assuming you have a React project set up you can install React router using npm or Yarn.
for npm npm install react-router-dom
for Yarn yarn add react-router-dom

You have to import and used it we will study

import {BrowserRouter, Route, Switch} from 'react-router-dom';

Every thing that gets rendered will need need to go inside <BrowserRouter> element, so wrap your App component inside it. Because all the other components mostly are in App component

Remember in latest version instead of Switch <Routes> is used

In index.ds

```
import App from "./App";
import { BrowserRouter } from "react-dom";

ReactDOM.render(
    <BrowserRouter>
        <App />
    </BrowserRouter>,
    document.getElementById('root')
);
```

In App.dsx

```
import {Route, Switch} from "react-router
-dom";
import About from "./About";
import Contact from "./Contact";

const App = () => {
  return (
    <>
      <Switch>
//switch use hoga tab he ek component
```

Sorry after
that go to Blue
page any like
dea the.

So when our website is open this component will display localhost3000

// display home

showroot

<Route path="/" component={Home} />

<Route path="/contact" component={Contact} />

//At localhost3000/contact when we go will get
// contact component

<Route path="/" component={About} />

<Route path="/contact" component={Contact} />

//At localhost3000/contact when we go will get
// contact component

//Esmy ek issue hoga ~~switch~~ ky jab
hum localhost300/contact py jayn gy
tabe be humara contact ky sth
sath wala About be show hoga
so esko door karny ky liya
ham exact ka use karang like
<Route exact path="/" component=
{About} />

Now localhost300/contact py just
contact display hoga.

If we want ky agr koi User
ghalat path enter karay to error
ah jay tbh ek error page bana lo
and

<Route component={Error} />

Escy cy hum just URI ky
through page ko show karwo sahky.
But if we want to use menu
and go to any component without reload

Lecture 57)

Lecture 57)

Create Read/Menu/Navbars using React Router

In React application instead of using <a> anchor tag to handle navigation between pages.. it is recommended to use the link component provided by React Router. when you used <a> tag click on the link trigger full page refresh, which is the default behaviour of a browser when clicking on link

In SPAs we avoid full page refreshes because they can be slow and disrupt the user experience

In <link> instead of href we use to

Now in latest version instead of component= [Home] used

Now element = {<Home />} is

```
import {link} from "react-router-dom";

<link exact to="/" > About us </link>
<link to="/contact"> Contact us </link>
```

import {link} from "react-router-dom";

<link exact to="/" > About us </link>
<link to="/contact"> contact us </link>

Listen- it will work But one issue abi active konsa link hai URL mn to nzr aay ga but page py dekh ky nai pata chaly ga- so Now according to latest update instead of <link> we used <navLink> so we can used it activeClassName attribut -

import {NavLink} from 'react-route-dom';

<NavLink exact activeClassName="active_class"
to="/"> About us </NavLink>

<NavLink activeClassName="active-class"
to="/contact"> contact us
</NavLink>

In Css Make active-class

Now

According to above work we can say Single page Application contain a single page on which we rendered different components

Lecture 58)

React Route Render Method
Diff b/w Render and
Component prop on React
Router

its Mean Route render Method

(1) <Route component> - Ya humny pooraha
(2) <Route render>
(3) <Route children> function // we
will not study it today

So previously we study :
<Route component>
But if we wants to pass

props inside it we have to use function.

```
<Route exact path="/contact" compentent={
  ()=> { <Contact name="Azan" /> }
} />
```

If we don't want to pass props we already study how to used it

```
<Route exact path="/contact" component=
  { <Contact /> } />
```

Now Same like we used componyt instead we can used render it will peaform same functionallity.

Different btw both

When we use component (instead of render or children, ~~below~~) the Router uses React.createElement to create a new React Element for the given component every time That mean like we know in inline function first time compiler

it second time he does not compile it again Here it we see that logic same component is created always.

So If we want inline rendering we used render we hmy wo inline function jesi wo functionality provide karta hai.

When to use which one.

No rule But most comonly it is prefered when no props use component when props exist use render

Lacture 59)

Use useParams Hooks in React Router

React Routes ky be apny hooks hain

(1) useHistory (2) usePaarams (3) useLocation

Today we will learn ~~use~~ ~~history~~ usePaarms hook

usePaarms

In React Router usePaarams is a tool that help you read and use information from the web address in your react componeut

imagine a website with different pages for each user Instead of creating separate pages for each user you can use dynamic URLs to represent user profiles For example instead of making

- example.com/users/1
- example.com/users/2
- example.com/user/3

You can have a single page with dynamic link like example.com/users/userId

With UseParaus we can access

I m blue page i m Not at right Place

So listen when we want to do
<Route path="/contact/Name" component={Name}

Ab jab jayn gy localhost3000/contact
/Name To contact component
display hoga Name Nai.
Because humny study kea tha
(switch) ek he display kerwata
hai jo be / esky agy nidy
to wy pahly contact mila wyy
korega To esko avoid kerny ky
liya be exact use hoga
ky exact contact hmesha localhost3000
/contact mn he dhkw
<Route exact path="/contact" component
="contact" />

userId value from the URL and use it in component to display or perform other function

for instance if you visit example localhost:12 'useParams' will help you retrieve 2 and you can use it to perform different function

App.js

```
import React from 'react';
import {[redacted] Route, Switch, Link }
  from 'react-[redacted]router-dom';
import UsersData from "./usersDetails";

function App(){
    return(
      [redacted]
        <div>
          <nav>
            <ul>
              <li>
<link to="/"> Home </link>
```

```
</li>
<li>
    <Link to="/users/1">User 1</Link>
</li>
<li>
    <Link to="/users/2">User 2</Link>
</li>
    </ul>
</nav>

<Switch>
    <Route exact path="/" component={Home} />
    <Route path="/users/:userId"
    component={UserDetails} />
</Switch>
    </div>
  );

}
export default App;
```

In UserDetail.jsx file

```
import React from 'react';
import {useParams} from 'react-router-dom';

function UserDetails(){                    // to return object
  const {userId} = useParams();

  return (<>
    <div>
      <h2> {userId} </h2>
    </div>
  </>
  );
}

default UserDetails
```

Lecture 60)

useLocation Hook in React Router

The useLocation return the location object that represent the current URL. You can think about it like a useState that returns a new location whenever the URL changes.

U can access pathname, search parameters, hash and more

import {useLocation} from 'react-Router-dom'

let c = useLocation();

We can print it to see what we can access
console.log(c);

c.pathname
c.key
c.state
etc

Lacture 61)

useHistory Hook in React Router

History key and location both multi hai but always use location for location Because in history location is mutable.

It allows you to access the history object such as

navigating to different routes, pushing new entries to the history stack or going back and forth between the visited page

import {useHistory} from 'react-router-dom';

let obj = useHistory();
//console it and see what what you can do it with
console.log(obj); // you can see it method etc in console

we can used
obj.push("/about"); method to navigate the user to "/about" route programmatically

history.goBack() → navigate back to previous page

etc

So our Three Hooks of react-router-dom is Done. There were useParams, useLocation, useHistory

Lacture 62) Live Filter Search using Hooks and React Js in Hindi

Lacture 63) missing in playlist

Lacture 64)

Great React 404 Error Page

```
import React from 'react';
import {Switch, Route} from 'react-router-dom';

function Error(){

  <Switch>
    <Route exact path="/" component={About}
    } />
    <Route exact path="/contact" component
    ={<Contact/>} />

    <Route component={Error} />
  </Switch>
}
```

Make an error component

Lacture 65)

Redirect in React Router

Same concept koi ghalat address dy mean ki page py jana chay jo exist he na karta hoto Error page py jany ka jisa kisi aur root page py bhej dena

Lacture 66)

How to install and use Bootstrap 5 in React js

Lacture 67)

Responsive website

Lacture 68)

How to host React js website live for free using Github

Lacture 69)

SASS

SASS stands for Syntactically awsome Stylesheets. It is a preprocessor scripting language that is used to

extend and enhance the capabilities of Css. Sass provides a more efficient and organized way to write Css code by adding features that are not available in Css

npm install --save-dev node-sass

it will come into devDependencies in package.JSON

The extension you will used is .scss Not .css which we used from css

We can do Nesting in it and making variables and many more features

One more use Variable different file we create karlo and import karlo is main styling file main

For learning SASS fully watch tapa technical video on it

Extra learning

```
import React,{useEffect} from 'react';
const App=()=>{
const [news,updateNews]= useState({ fname:'',
lname:'', File:null });

function handleChange(e){
    const {name, value, type} = e.target;

IF(type===='file'){
    updateNews(()=>{
    return {...news, [name]: e.target.File[0].
    });
}
else{
    updateNews(()=>{
return {...news, [name]: value, };
    })
}
function print(){
console.log(news.fname);
```

```
console.log(news.lname);   console.log(news.file);
}

return (  <>
<input type="text" value={news.fname} onChange=
{handleChange} name="fname" placeholder="Fir
name" />
<input type="text" value={news.lname} name=
"lname" onChange={handleChange}
placeholder="Last Name" />

<input type="file" name="file" onChange={
{handleChange} />

<div style={{ height:'300px', border:'2px
solid red' }} >
    { news.file &&
  ( <img
      src={URL.createObjectURL(news.file)}
      alt="image"
      style={{ maxWidth:'100%' }}
    />
  ) }
</div>
```

<button onClick={print}> Submit </button>

</>

); };

export default App;

// When the file is Selected using the file input 'URL.createObjectURL() method is used to create a temporary URL for the selected file. This URL is used to display image preview

Full Stack Mern App

Harsh Pathak

Search, Pagination, filter, Sort, Export to Csv

Front End - React Backend Express, Mongodb

- Front end
- My Mern App Folder
  - client // To create reactApp npx create-react-app client
    - Public - for images
    - src
      - component
        - Header
        - Pagination

- Pages
  - Edit Page
  - Home Page
  - Profile
  - Register
- services
  - ~~Apis~~
  - Apis

File App.js index.js

For Styling we used React Bootstrap

React select

npm i react-toastify for stylish alerts etc

Spinner react bootstrap component for loading

use Navigat Hook

```
import { useNavigate } from "react-router-dom";
const addUser = () => {
    navigate("/register");
}
<button onClick={addUser}>Click</button>
```

Backend

- My Mean App
  - Client - frontend
  - Server - Backend
    - app.js - file .env
    - controllers .db .models
    - multiaccounts .public
    - Routes .uploads