**20TS2101S-MEAN STACK WEB DEVELOPMENT**

**REACT JSX**

* JSX stands for JavaScript XML.
* JSX allows us to write HTML in React.
* JSX makes it easier to write and add HTML in React.

**Single Tag Styling**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

const myele1=<h1>Hello World</h1>

ReactDOM.render(myele1,document.getElementById('root'));

**Multiple Tags(Wrapping Elements)**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

const myele1=<div>

<h1>Hello World</h1>

<h2>CSE</h2>

<p>HAI HOW ARE YOU</p>

</div>

ReactDOM.render(myele1,document.getElementById('root'));

**JSX Styling:**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

var mystyle={color:'blue',textAlign:'center'}

const myele1=<div style={mystyle}>

<h1>Hello World</h1>

<h2>CSE</h2>

<p>Addition is: {5+5} </p>

</div>

ReactDOM.render(myele1,document.getElementById('root'));

ReactDOM.render(myele1,document.getElementById('root'));

**JSX Styling(Using External Style Sheet)**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css'

const myele1=<div>

    <h1 className="header">Welcome KL university</h1>

</div>

ReactDOM.render(myele1,document.getElementById('root'));

**Index.css**

.header

{

  color: yellow;

}

**REACT COMPONENTS**

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML via a render() function.

1. Function Components
2. Class Components
3. **Function Components**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

function Header()

{

    return <h1>KL university, Dept of CSE</h1>

}

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

Pass multiple Function Calling:

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

function Header()

{

    return <h1>KL university, Dept of CSE</h1>

}

function Sidebar()

{

    return <h1>KL university, Dept of CSE</h1>

}

function Navbar()

{

    return <h1>KL university, Dept of CSE</h1>

}

function Website()

{

    return <div>

        <Header/>

        <Sidebar/>

        <Navbar/>

    </div>

}

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

1. **Class Components**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

class Website extends React.Component

{

    render()

    {

        return <h1>Welcome to KL university</h1>

    }

}

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

**Multiple Class Components**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

class A extends React.Component

{

    render()

    {

        return <h1>Welcome to KL university</h1>

    }

}

class B extends React.Component

{

    render()

    {

        return <div>

            <A/>

        <h1>Welcome to CSE</h1>

        </div>

    }

}

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

**Multiple Class Components(With External Call)**

**index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import Linkwebsite from './App.js';

class Website extends React.Component

{

    render()

    {

        return <div>

        <Linkwebsite/>

        <h1>Welcome to KL university</h1>

        </div>

    }

}

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

**App.js**

import logo from './logo.svg';

import './App.css';

import React from 'react';

class Linkwebsite extends React.Component

{

  render()

  {

    return <h1>Welcome</h1>

  }

}

export default Linkwebsite;

# REACT PROPS

* Props are arguments passed into React components.
* Props are passed to components via HTML attributes.

Index.js

import React from 'react';

import ReactDOM from 'react-dom';

class Car extends React.Component {

  render() {

    return <h2>I am a {this.props.brand}!</h2>

  }

}

const myelement = <Car brand="Ford" />;

ReactDOM.render(myelement, document.getElementById('root'));

**Pass one component values to another:**

**Index.js**

import React from 'react';

import ReactDOM from 'react-dom';

class Parent extends React.Component

{

  render()

  {

    return <div>

    <h1>Welcome to MSWD S20 Class</h1>

    <h2>Inherited Value is:{this.props.v}</h2>

    </div>

  }

}

class Child extends React.Component

{

  render()

  {

    return <Parent v={"SKILLDEVELOPMENT"}/>

  }

}

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

# REACT STATE (HOOKS)

React components has a built-in state object.

The state object is where you store property values that belongs to the component.

When the state object changes, the component re-renders.

Index.js

import React from 'react';

import ReactDOM from 'react-dom';

class Access extends React.Component

{

  constructor()

  {

    super();

    this.state={model:"hero",color:"red",year:"2015"}

  }

  render()

  {

    return <div>

    <h1>My bike model is:{this.state.model}</h1>

    <h1>My bike color is:{this.state.color}</h1>

    <h1>My bike year is:{this.state.year}</h1>

    </div>

  }

}

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

Change the value of the state

Index.js

import React from 'react';

import ReactDOM from 'react-dom';

class Access extends React.Component

{

  constructor()

  {

    super();

    this.state={model:"hero",color:"red",year:"2015"}

  }

  Changevalue=()=>

  {

    this.setState({model:"Yamaha"});

  }

  render()

  {

    return <div>

    <h1>My bike model is:{this.state.model}</h1>

    <h1>My bike color is:{this.state.color}</h1>

    <h1>My bike year is:{this.state.year}</h1>

    <button type="button" onClick={this.Changevalue}>Change value</button>

    </div>

  }

}

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

**EVENT HANDLING:**

Event handling using JavaScript:

Event.html

<html>

<body>

<form>

<input type="submit" name="b1" onclick="alert('hello')";>

<input type="submit" name="b1" ondblclick="alert('hello')";>

<input type="submit" name="b1" onmouseover="alert('hello')";>

<input type="submit" name="b1" onmouseout="alert('hello')";>

</form>

</body>

</html>

Event handling using Function:

index.js

import React from 'react';

import ReactDOM from 'react-dom';

function Shoot()

{

  alert("MSWD S20 STUDENTS WELCOME YOU ALL");

}

    const myele=(<button type="button" onClick={Shoot}>Click me</button>);

ReactDOM.render(myele,document.getElementById('root'));

Event handling using React Class Component:

index.js

import React from 'react';

import ReactDOM from 'react-dom';

class Website extends React.Component

{

  Shoot=(a)=>

  {

    alert("MSWD S20 STUDENTS WELCOME YOU ALL"+a)

  }

  render()

    {

        return (<button type="button" onClick={this.Shoot("By Srithar")}>Click me</button>);

    }

}

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

COMPLEX STATE:

Eg1: (Single state)

Method 1:

import ReactDOM from 'react-dom';

import React, { useState } from 'react';

function Example() {

const [count, setCount] = useState(0);

return (

<div>

<p>You clicked {count} times</p>

<button onClick={() => setCount(count + 1)}>Click me </button>

</div>

);

}

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

Method 2:

import ReactDOM from 'react-dom';

import React, { useState } from 'react';

function Example() {

const [count, setCount] = useState(0);

const One=()=>

{

setCount(count+5)

}

return (

<div>

<p>You clicked {count} times</p>

<button onClick={one}>Click me </button>

</div>

);

}

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

Eg2: (Multiple states)

import ReactDOM from 'react-dom';

import React, { useState } from 'react';

const App = () => {

    const [left, setLeft] = useState(0)

    const [right, setRight] = useState(0)

    return (

      <div>

        {left}

        <button onClick={() => setLeft(left + 1)}>

          left

        </button>

        <button onClick={() => setRight(right + 1)}>

          right

        </button>

        {right}

      </div>

    )

  }

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

Handling Arrays:

import ReactDOM from 'react-dom';

import React, {useState} from 'react';

/\* Passing state to child componets    \*/

/\*Multiple state components\*/

function Example()

{

  const [left,setLeft]=useState(0);

  const [right,setRight]=useState(0);

  const [allClicks,setAll]=useState([])

  const One=()=>

  {

    setLeft(left+1)

    setAll(allClicks.concat('L'))

  }

  const Two=()=>

  {

    setRight(right+1)

    setAll(allClicks.concat('R'))

  }

return(<div>

   {left}

  <button onClick={One}>Left</button>

  <button onClick={Two}>Right</button>

   {right}

   {allClicks.join('')}

</div>);

}

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

**Handling Forms in React:**

**import React from 'react';**

**import ReactDOM from 'react-dom';**

**class MyForm extends React.Component {**

**constructor(props) {**

**super(props);**

**this.state = { username: '' };**

**}**

**myChangeHandler = (event) => {**

**this.setState({username: event.target.value});**

**}**

**render() {**

**return (**

**<form>**

**<h1>Hello {this.state.username}</h1>**

**<p>Enter your name:</p>**

**<input**

**type='text'**

**onChange={this.myChangeHandler}**

**/>**

**</form>**

**);**

**}**

**}**

**ReactDOM.render(<MyForm />, document.getElementById('root'));**

REACT ROUTER:

React Router is a standard **library for routing in React**. It enables the navigation among views of various components in a React Application, allows changing the browser URL, and keeps the UI in sync with the URL.

Install: **npm install react-router-dom --save**

Index.js

import ReactDOM from 'react-dom';

import React from 'react';

import {

  BrowserRouter as Router,

  Switch,

  Route,

  Link,

  NavLink

} from "react-router-dom";

import Home from './Home';

import About from './About';

import Notfound from './Notfound';

const route=(

  <Router>

  <div>

  <h1>Welcome to Hyper link creation</h1>

  <ul>

    <li>

      <NavLink to="/" exact activeStyle={{color:"red"}}>Home</NavLink>

    </li>

    <li>

    <NavLink to="/a" exact activeStyle={{color:"red"}}>About</NavLink>

    </li>

  </ul>

  <Switch>

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

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

  <Route component={Notfound}/>

  </Switch>

  </div>

  </Router>

)

ReactDOM.render(route,document.getElementById('root'));

Home.js

import ReactDOM from 'react-dom';

import React from 'react';

class Home extends React.Component

{

    render()

    {

        return (<div>

            <h1>Hello guys</h1>

            </div>

            )

    }

}

export default Home;

about.js

import ReactDOM from 'react-dom';

import React from 'react';

class About extends React.Component

{

    render()

    {

        return (

            <div>

            <h1>About Me</h1>

            </div>)

    }

}

export default About;

**Notfound.js**import ReactDOM from 'react-dom';

import React from 'react';

const Notfound=()=><h1>Not found</h1>

export default Notfound

**MATERIAL UI:**

* Material-UI is simply a library that allows us to import and use different components to create a user interface in our React applications.
* Install command: **npm install @material-ui/core @material-ui/icons**

**Sidebar making**

**Index.js**

import ReactDOM from 'react-dom';

import React from 'react';

import App from './App';

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

**App.js**

import React from 'react';

import {makeStyles} from "@material-ui/core";

const useStyles=makeStyles({

  sideMenu:{

    display:'flex',

    flexDirection:'column',

    position:'absolute',

    left:'0px',

    width:'300px',

    height:'100%',

    backgroundColor:'#234'

  }

})

const App=()=>{

  const Classes=useStyles();

  return(

    <div className={Classes.sideMenu}>

    </div>

  );

}

export default App;

**­­APP BAR WITH CARD:**

**Index.js**

import ReactDOM from 'react-dom';

import React from 'react';

import App from "./App";

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

**App.js**

import React from 'react';

import {Grid} from "@material-ui/core";

import Header from './Header.jsx';

import Content from './Content';

const App=()=>{

  return(

    <Grid container direction="column">

      <Grid item>

       <Header/>

      </Grid>

      <Grid item container>

        <Grid item sm={2}/>

        <Grid item sm={8}>

          <Content/>

        </Grid>

        <Grid item sm={2}/>

      </Grid>

    </Grid>

  );

};

export default App;

**Header.jsx**

import React from "react";

import { AppBar, Toolbar, Typography } from "@material-ui/core";

import AcUnitIcon from '@material-ui/icons/AcUnit';

const Header=()=>{

return(

    <AppBar position="static">

        <Toolbar>

        <AcUnitIcon/>

            <Typography>This is header</Typography>

        </Toolbar>

    </AppBar>

);

};

export default Header;

**Content.jsx**

import React from "react";

import Coffeecard from './Coffeecard';

import {Grid} from "@material-ui/core";

const Content=()=>{

    return(

        <Grid container>

            <Grid item xs={4}>

            <Coffeecard/>

            </Grid>

            <Grid item xs={4}>

            <Coffeecard/>

            </Grid>

            <Grid item xs={4}>

            <Coffeecard/>

            </Grid>

            <Grid item xs={4}>

            <Coffeecard/>

            </Grid>

            <Grid item xs={4}>

            <Coffeecard/>

            </Grid>

            <Grid item xs={4}>

            <Coffeecard/>

            </Grid>

        </Grid>

    );

};

export default Content;

**Conffeecard.jsx**

import \* as React from 'react';

import ImageList from '@material-ui/core/ImageList';

import ImageListItem from '@material-ui/core/ImageListItem';

const itemData = [

    {

      img: 'https://images.unsplash.com/photo-1551963831-b3b1ca40c98e',

      title: 'Breakfast',

    },

    {

      img: 'https://images.unsplash.com/photo-1551782450-a2132b4ba21d',

      title: 'Burger',

    },

    {

      img: 'https://images.unsplash.com/photo-1522770179533-24471fcdba45',

      title: 'Camera',

    },

    {

      img: 'https://images.unsplash.com/photo-1444418776041-9c7e33cc5a9c',

      title: 'Coffee',

    },

    {

      img: 'https://images.unsplash.com/photo-1533827432537-70133748f5c8',

      title: 'Hats',

    },

    {

      img: 'https://images.unsplash.com/photo-1558642452-9d2a7deb7f62',

      title: 'Honey',

    },

    {

      img: 'https://images.unsplash.com/photo-1516802273409-68526ee1bdd6',

      title: 'Basketball',

    },

    {

      img: 'https://images.unsplash.com/photo-1518756131217-31eb79b20e8f',

      title: 'Fern',

    },

    {

      img: 'https://images.unsplash.com/photo-1597645587822-e99fa5d45d25',

      title: 'Mushrooms',

    },

    {

      img: 'https://images.unsplash.com/photo-1567306301408-9b74779a11af',

      title: 'Tomato basil',

    },

    {

      img: 'https://images.unsplash.com/photo-1471357674240-e1a485acb3e1',

      title: 'Sea star',

    },

    {

      img: 'https://images.unsplash.com/photo-1589118949245-7d38baf380d6',

      title: 'Bike',

    },

  ];

const Coffeecard=()=>{

  return (

    <ImageList sx={{ width: 500, height: 450 }} cols={3} rowHeight={164}>

      {itemData.map((item) => (

        <ImageListItem key={item.img}>

          <img

            src={`${item.img}?w=164&h=164&fit=crop&auto=format`}

            srcSet={`${item.img}?w=164&h=164&fit=crop&auto=format&dpr=2 2x`}

            alt={item.title}

            loading="lazy"

          />

        </ImageListItem>

      ))}

    </ImageList>

  );

};

export default Coffeecard;

**MONGO DB**

**-It is a NoSQL database**

**-Relational DB->Stored in rows & columns**

**-NoSQL->Key-Value pair/document**

**Difference B/W MySQL and MongoDB**

**1.table=collection**

**2.insert=insertOne/insertMany**

**3.record=document**

**4.select=findOne/findMany**

**5.order=sort**

**6.delete=deleteOne/deleteMany**

**7.update=updateOne/updateMany**

**1. Start server**

**open cmd,**

**>cd C:\Program Files\MongoDB\Server\5.0\bin**

**>mongod**

**2.Start Client**

**Open cmd,**

**>mongo**

**>show dbs**

**admin**

**config**

**local**

**>use admin**

**>show collections**

**students**

**staff**

**>db.students.find();**

**Create DB:**

**App.js**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/demodb";

MongoClient.connect(url,function(err,db){

  if (err) throw err;

  console.log("DB created success");

  db.close();

});

**Create collection:**

**App.js**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){

  if (err) throw err;

  var dbname=db.db("demodb");

  dbname.createCollection("students", function(err, result){

    if(err) throw err;

    console.log("Collection success");

    db.close();

  });

 });

**Insert one value:**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){

  var dbname=db.db("demodb");

  dbname.collection("students").insertOne({name:"srithar",city:"coimbatore"},function(err, result){

    if(err) throw err;

    console.log("Insert success");

    db.close();

  });

 });

**Insert Multiple Values:**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){

  var dbname=db.db("demodb");

  var muldata=[{name:"vijay",city:"chennai"},

  {name:"ram",city:"tirupur"},

];

  dbname.collection("students").insertMany(muldata,function(err, result){

    if(err) throw err;

    console.log("Insert success");

    db.close();

  });

 });

**Find one:**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){``

  if(err) throw err;

  var dbname=db.db("srithar");

  dbname.collection("students").findOne({},function(err,result){

    if(err) throw err;

    console.log(result.name);

    db.close();

});

});

**Find Many:**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){``

  if(err) throw err;

  var dbname=db.db("srithar");

  dbname.collection("students").find({}).toArray(function(err,result){

    if(err) throw err;

    console.log(result);

    db.close();

});

});

**Find Specific:**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){``

  if(err) throw err;

  var dbname=db.db("srithar");

  dbname.collection("students").find({name:'Ramesh'}).toArray(function(err,result){

    if(err) throw err;

    console.log(result);

    db.close();

});

});

UpdateOne:

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){``

  if(err) throw err;

  var dbname=db.db("srithar");

  var myfield={city:"Chennai"};

  var newvalues={$set:{city:"Sathankulam"}};

  dbname.collection("students").updateOne(myfield,newvalues,function(err,result){

    if(err) throw err;

    console.log("update success");

    db.close();

});

});

**Sort:**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){``

  if(err) throw err;

  var dbname=db.db("srithar");

var sortorder={name:1};

  dbname.collection("students").find().sort(sortorder).toArray(function(err,result){

    if(err) throw err;

    console.log(result);

    db.close();

});

});

**Limit:**

var MongoClient=require('mongodb').MongoClient;

var url="mongodb://localhost:27017/";

MongoClient.connect(url,function(err,db){

  if (err) throw err;

  var dbname=db.db("srithar");

  dbname.collection("employee").find().limit(2).toArray(function(err, result){

    if(err) throw err;

    console.log(result);

    db.close();

  });

});

**Join Collections**

MongoDB is not a relational database, but you can perform a left outer join by using the $lookup stage.

The $lookup stage lets you specify which collection you want to join with the current collection, and which fields that should match.

**Find Some:**

var MongoClient = require('mongodb').MongoClient;

var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

  if (err) throw err;

  var dbo = db.db("srithar");

  dbo.collection("students").find({}, { projection: { \_id: 0, name: 1 } }).toArray(function(err, result) {

    if (err) throw err;

    console.log(result);

    db.close();

  });

});

**Aggregate:**

**db.students.aggregate([{$group : {\_id : "$name", city : {$sum : 1}}}])**

**Getting Data from server:**

**Index.js**

import ReactDOM from 'react-dom';

import React from 'react';

import axios from 'axios';

function App(){

    const getJoke=()=>{

    axios.get("https://jsonplaceholder.typicode.com/posts").then(

        (response)=>{

            console.log(response);

        }

    );

}

return(

    <div>

        <h1>This is a Axios example</h1>

        <button onClick={getJoke}>Click me</button>

    </div>

);

}

export default App;

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

**Index.js**

import ReactDOM from 'react-dom';

import React from 'react';

import axios from 'axios';

import { useState } from 'react';

function App(){

const[Joke,setJoke]=useState("");

    const getJoke=()=>{

    axios.get("https://jsonplaceholder.typicode.com/posts/1").then(

        (response)=>{

            console.log(response);

            setJoke(response.data.title+"...."+response.data.body);

        }

    );

};

return(

    <div>

        <h1>This is a Axios example</h1>

        <button onClick={getJoke}>Click me</button>

        {Joke}

    </div>

);

}

export default App;

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