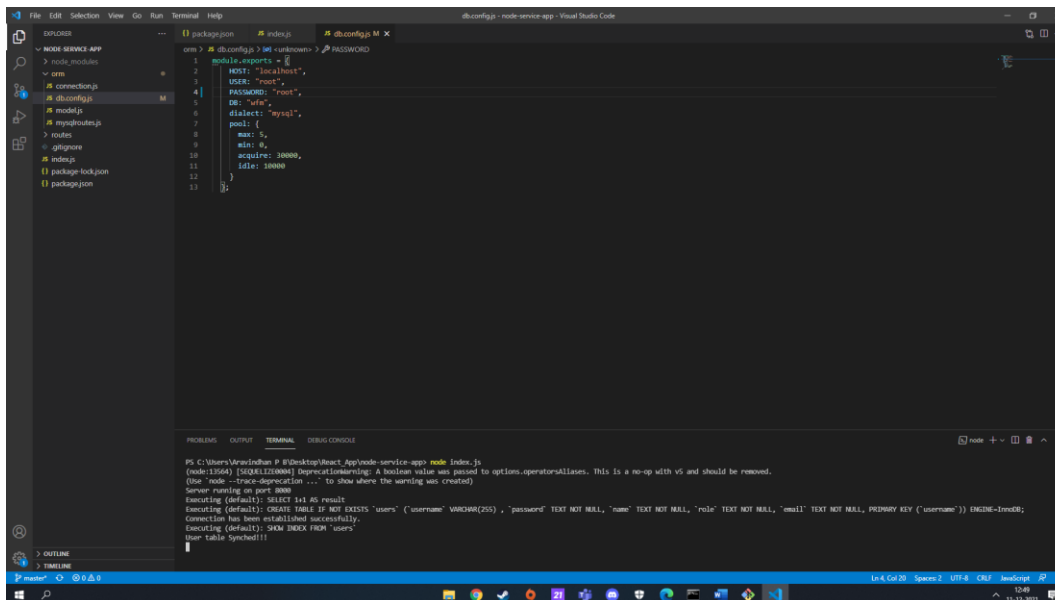
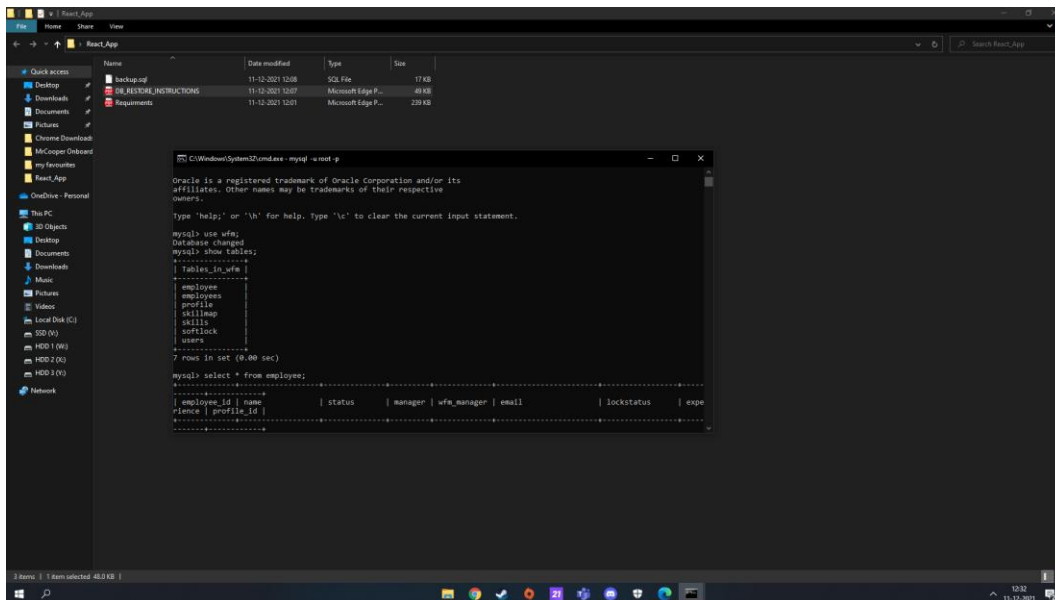


REACT NODE PROJECT - WORKFORCE MANAGEMENT

-Aarthi Balamurugan

Database Setup:

1. Installation of MySQL.
2. Using backup.sql file configured WFM Database



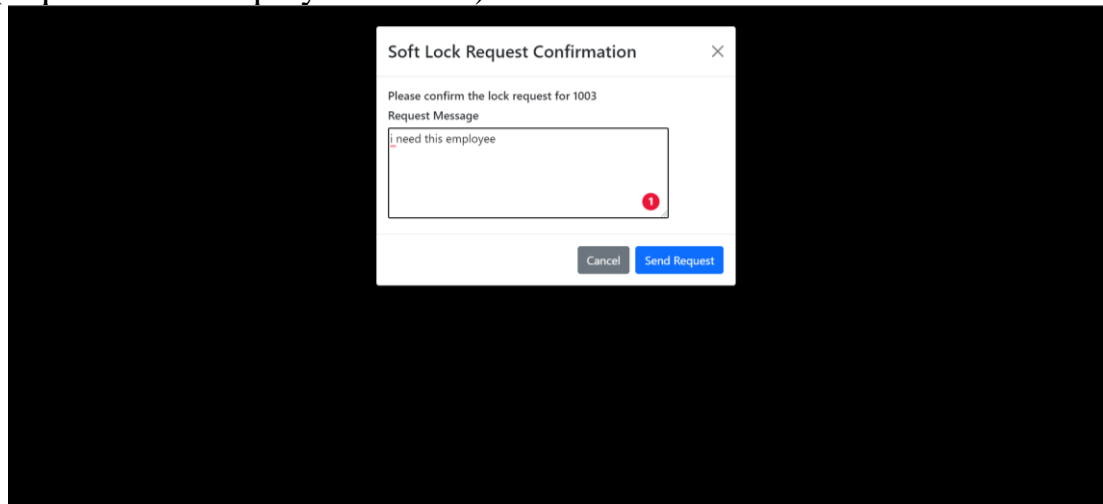
Solution Screenshots:

Manager Home Screen

Manager Home						
						rohit Logout
EmployeeID	Name	Skills	Profile	Experience	Manager	Status
1003	Abdul Kadar	Spring Boot,Angular,NODE JS	Full Stack Javascript Developer	2	rohit	not_requested
1004	John Paul	Python,SQL,PLSQL	Python Developer	7	rohit	not_requested
1005	Atul Kulkarni	Core Java,SQL	Java Developer	5	rohit	not_requested
1006	Ajay Barot	SQL,PLSQL	Backend Developer	2	rohit	not_requested
1007	Benny Shaw	Spring Boot,Angular,REACT JS	Full Stack Java Developer	2	rohit	not_requested
1008	Akram Khan	Spring Boot,Angular	Full Stack Java Developer	7	rohit	not_requested
1009	Bejoy Menon	NODE JS,REACT JS	Full Stack Javascript Developer	2	rohit	not_requested
1010	Ravi Kishore	Core Java,REACT JS,NODE JS	Full Stack Javascript Developer	3	rohit	not_requested
1011	Nandha Kumar	Spring Boot,REACT JS,NODE JS	Full Stack Javascript Developer	5	rohit	not_requested
1012	Bennet Johnson	SQL,PLSQL	Backend Developer	7	rohit	not_requested
1013	Priya Sen	Core Java,SQL	Java Developer	2	rohit	not_requested

SoftLockRequest :

(requested for employee id 1003)



After this send request in **WFM Manager Home** (logged in as lokesh wfm manager

WFM Manager Home				lolesh Logout
EmployeeID	Requestee	Req Date	Req Message	Status
1002	rohit	2022-01-06	i need this employee	waiting
1003	rohit	2022-01-06	i need this employee	waiting

SoftLock Request Confirmation:

Accepted or Rejected status will be send to the employee table

WFM Manager Home
lolesh Logout

EmployeeID	Requestee	Status
1002	rohit	waiting
1003	rohit	waiting

Soft Lock Request Confirmation

Status Update For Request Lock
Employee ID: 1003
Requestee: rohit
Request Description: i need this employee
Status:
Accepted

Cancel
Send Request

Solution approach:

Manager/home.tsx:

```
import Employee from "../Employee";
import Login from "../Login";

const ManagerHome=()=>{
  const username = localStorage.getItem("username")
  return (
    <div>
    <div>
      <nav className="navbar navbar-dark bg-dark justify-content-between">
        <a className="navbar-brand" > Manger Home</a>
        <form className="form-inline my-2 my-lg-0">
          <a className="navbar-brand" >{username}</a>
        </form>
      </nav>
    </div>
    </div>
  )
}
```

```

        <button className="btn btn-outline-success my-2 my-sm-0" onClick={()=>
        {if(username)

            return [localStorage.clear(),Login]}}>Logout</button>
        </form>
    </nav>
</div>
<div>
    <Employee></Employee>
</div>
</div>
)
}

```

export default ManagerHome

Employee.tsx:

```

import { useState,useEffect,useRef } from "react";
import { Redirect } from "react-router";
import axios from "axios";
import Popup from '../popup';
import ReactDOM from "react-dom";

```

```

type Employee={
    EmployeeID:number;
    Name:string;
    Skills:string;
    Status:string;
    Experince:number;
    Manager:string;
    WFManager:string;
    content:string;
    handleClose:string;
    requestmessage:string;
}

```

```

const Employee = (props:any)=>{
    const [isOpen, setIsOpen] = useState(false);
    const[id,Setid] = useState(0)

```

```
const[reqmessage,Setreqmessage] = useState("")
const [peopleData,setPeopleData] =useState([])
const [accepted,setaccepted]=useState("request_waiting");
const manager=localStorage.getItem("username")
```

```
function togglePopup(event:any){
  setIsOpen(!isOpen);
  if(isOpen!==true){
    const id=event.target.id;
    Setid(id);
  }
}
```

```
async function peopleRead(){
  try{
    let response =
      await axios.get("http://localhost:8000/employees/emp")
      setPeopleData(response.data)
  }
  catch(e){
    setPeopleData([])
  }
}
```

```
async function addDetails(){
  try{
    console.log(id,reqmessage,manager)
    const response =
      await axios.put("http://localhost:8000/employees/updateEmployee",{employeeid :
id,status:accepted});
    console.log(id,accepted)
    if(response.status == 200){
      const res = await axios.post("http://localhost:8000/employees/insertsoftlock",
      {
        employeeid:id,manager:localStorage.getItem("username"),responsemessage: reqmessage});
      Setreqmessage("");
      peopleRead();
    }
  }
  catch(e){
    console.log("Error");
  }
}
```

```

useEffect(()=>{
  peopleRead()
},[])

return (
<div className="card"
style={{width: "1300px", minHeight: "700px",position: "relative", top:"50px",left:"2%"
,padding: "30px", color:"darkblue"}}>

<table className="table">
  <thead className="table-primary">
    <tr>

      <th scope="col">Employee ID</th>
      <th scope="col">Name</th>
      <th scope="col">Skills</th>
      <th scope="col">Status</th>
      <th scope="col">Experience</th>
      <th scope="col">Manager</th>
      <th scope="col">WFManager</th>
      <th scope="col"> Request Lock</th>

    </tr>
  </thead>
  <tbody>
    {
      peopleData.map((x:Employee)=>{
        let empid = (x.EmployeeID).toString();
        if(manager == x.Manager){
          return(
            <tr key={x.EmployeeID}>

              <td>{x.EmployeeID}</td>
              <td>{x.Name}</td>
              <td>{x.Skills}</td>
              <td>{x.Status}</td>
              <td>{x.Experince}</td>
              <td>{x.Manager}</td>
              <td>{x.WFManager}</td>
              <td>
                <button type="button" className="btn btn-outline-primary" id={empid}
onClick={togglePopup}>Request Lock</button>
              </td>
            </tr>
          )
        }
      })
    }
  </tbody>
</table>

```

```

      isOpen && <Popup
      content={<>
        <h2 className="highlight">Soft Lock Request Confirmation</h2>
        <br/>
        <p>Please Confirm the Lock Request is for {id}</p>
        <br />
        <p>Request Message (Message must be atleast 10 Charecters Long !)</p>
        <br/>
        <textarea className="form-control" rows={3} placeholder="Enter your request
message here..." value={reqmessage} onChange={(e)=>Setreqmessage(e.target.value)}></textarea>
        <br/>
        <div>
          <button type="button" className="btn btn-danger btn-space"
onClick={togglePopup}>Cancel</button>
          <button type="button" className="btn btn-primary btn-space"
onClick={()=>{addDetails();togglePopup(id);}}>Send Request</button>
        </div>
      </>
      handleClose={togglePopup}
    />
  }
</td>
</tr>
)
}
})
}
</tbody>
</table>
</div>
)}

```

export default Employee

WfmManager/home.tsx:

```

import Login from "../Login";
import Empreq from "../Empreq";

const WFMHome={()=>{
  const username = localStorage.getItem("username")
  return (
    <div>

```

```

<div>
  <nav className="navbar navbar-dark bg-dark justify-content-between">
    <a className="navbar-brand" >Work Force Manger Home</a>
    <form className="form-inline my-2 my-lg-0">
      <a className="navbar-brand" >{username}</a>
      <button className="btn btn-outline-success my-2 my-sm-0" onClick={()=>
        {if(username)
          return [localStorage.clear(),Login]}}>Logout</button>
    </form>
  </nav>
</div>
<div>
  <Empreq></Empreq>
</div>
</div>
)
}export default WFMHome

```

Empreq.tsx:

```

import { useState,useEffect,useRef } from "react";
import { Redirect } from "react-router";
import axios from "axios";
//import Popup from 'reactjs-popup';
import Popup from '../popup';
import ReactDOM from "react-dom";
import Employee from "../Managers/Employee";

```

```

type EmpRequest={
  EmployeeID:number;
  Requestee:string;
  RequestDate:string;
  RequestMessage:string;
  WFMMManager:string;
  content:string;
  handleClose:string;
}

```

```

const EmpRequest = (props:any)=>{
  const [isOpen, setIsOpen] = useState(false);
  const[id,Setid] = useState("");
  const[req,setReq]=useState("");
  const[reqMessage,setReqmes]=useState("");
  const[wfmMan,setwfmManager]=useState("");
  const [employeeData,setemployeeData] =useState([])

```



```

const [accepted,setaccepted]=useState("Accepted");
const WFMmanager=localStorage.getItem("username")
const[status,Setstatus] = useState("");
function togglePopup(event:any){
  setIsOpen(!isOpen);
  if(isOpen!==true){
    const empid=event.target.id;
    Setid(empid);

  }
}

async function employeeRead(){
  try{
    let response =
      await axios.get("http://localhost:8000/employees/softlocktabel")
    setemployeeData(response.data)
  }
  catch(e){
    setemployeeData([])
  }
}

async function addEmployee() {
  try {
    const res = await axios.put("http://localhost:8000/employees/updateSoftlock",{ employeeid:
parseInt(Array[0]),status:status});
    employeeRead()
  }
  catch (e) {
    console.log("Error");
  }
}

useEffect(()=>{
  employeeRead()
},[])

let Array = (id).split(",");

return (
<div className="card"
style={{width: "1300px", minHeight: "700px",position: "relative", top:"50px",left:"2%"
,padding: "30px", color:"darkblue"}}>

```

```

<h4><u>Request Awaiting Approval</u></h4>
<table className="table">
  <thead className="table-primary">
    <tr>

      <th scope="col">Employee ID</th>
      <th scope="col">Requestee</th>
      <th scope="col">Request Date</th>
      <th scope="col">Request Message</th>
      <th scope="col">View Details</th>
    </tr>
  </thead>
  <tbody>
    {
      employeeData.map((x:EmpRequest)=>{
        console.log(EmpRequest)
        let empid=(x.EmployeeID)+','+(x.Requestee)+','+(x.RequestMessage)

        if(WFMmanager == x.WFMManager){
          return(
            <tr key={x.EmployeeID}>
              <td>{x.EmployeeID}</td>
              <td>{x.Requestee}</td>
              <td>{x.RequestDate}</td>
              <td>{x.RequestMessage}</td>
              <td>
                <button type="button" className="btn btn-outline-primary" id={empid}
onClick={togglePopup}>View Details</button>
                {
                  isOpen && <Popup
content={<>
  <h2 className="highlight">Soft Lock Request Confirmation</h2>
  <br/>
  <h4><p><u>Status update for Request Lock</u></p></h4>
  <br />
  <form>
    <div className="form-group">
      <label className="col-sm-2 col-form-label" ><b>Employee
ID :&ensp;</b>{Array[0]}</label>
      </div>
      <div className="form-group">
        <label className="col-sm-2 col-form-label" >
<b>Requestee :&ensp;</b>{Array[1]}</label>

```

```

        </div>
        <div className="form-group ">
            <label className="col-form-label" > <b>Request
Message :&emsp;</b>{[Array[2]]}</label>
        </div>
        <div className="form-group">
            <label className="col-form-label"><b>WFM
Manager :&emsp;</b>{WFMmanager}</label>
        </div>
        <div className="dropdown">
            <label className="col-form-
label"><b>Status :&emsp;&emsp;&emsp;</b></label>
            <select className="input-class" required value={status}
onChange={(e)=>Setstatus(e.target.value)}>
                <option value="selectvalue">select the option </option>
                <option value="Approved">Approved</option>
                <option value="Rejected">Rejected</option>
            </select>
        </div>
    </form>
    <br/>
    <div>
        <button type="button" className="btn btn-danger btn-space"
onClick={togglePopup}>Cancel</button>
        <button type="button" className="btn btn-primary btn-space"
onClick={()=>{addEmployee();togglePopup(id);}}>Send Request</button>
    </div>
    </>
    handleClose={togglePopup}
  />
}
</td>
</tr>
</tbody>
</table>
</div>
)}

```

export default EmpRequest