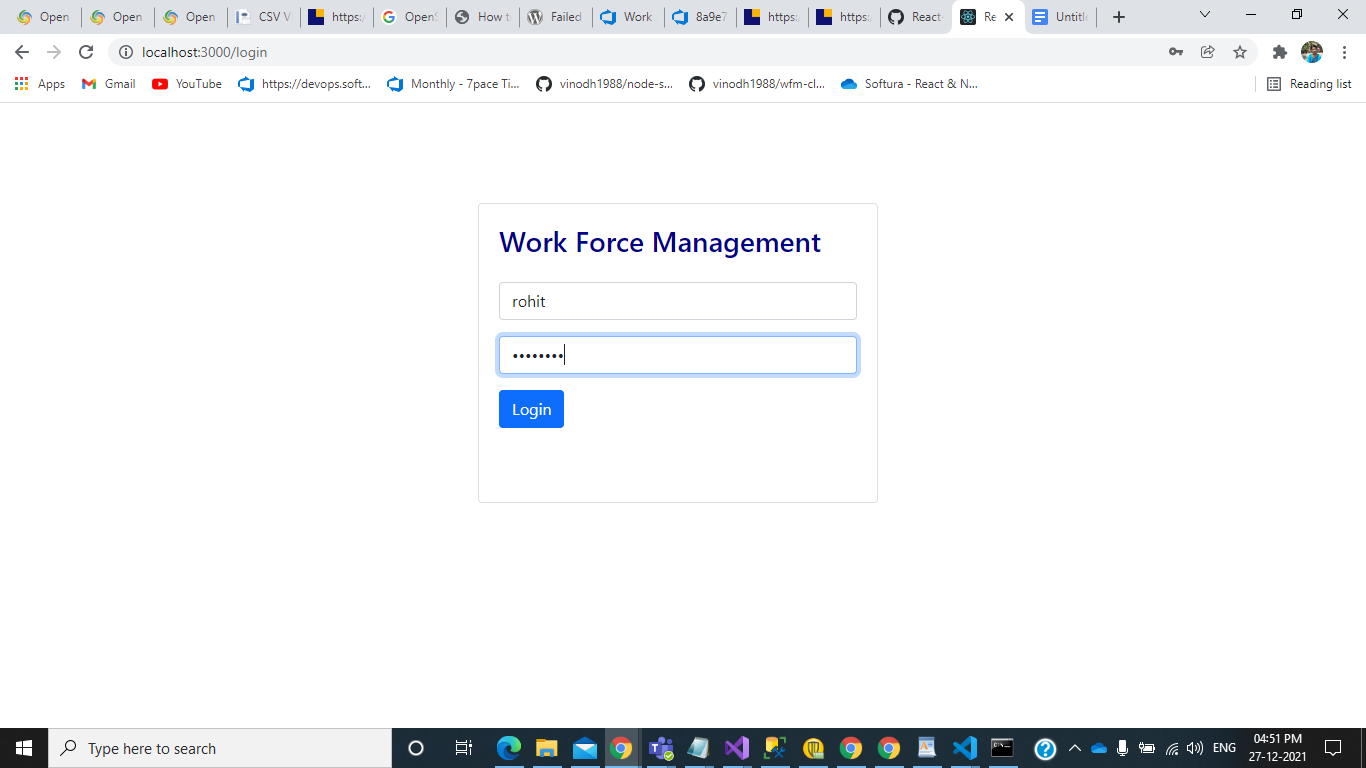
**SOLUTION APPROACH AND SCREENSHOTS**

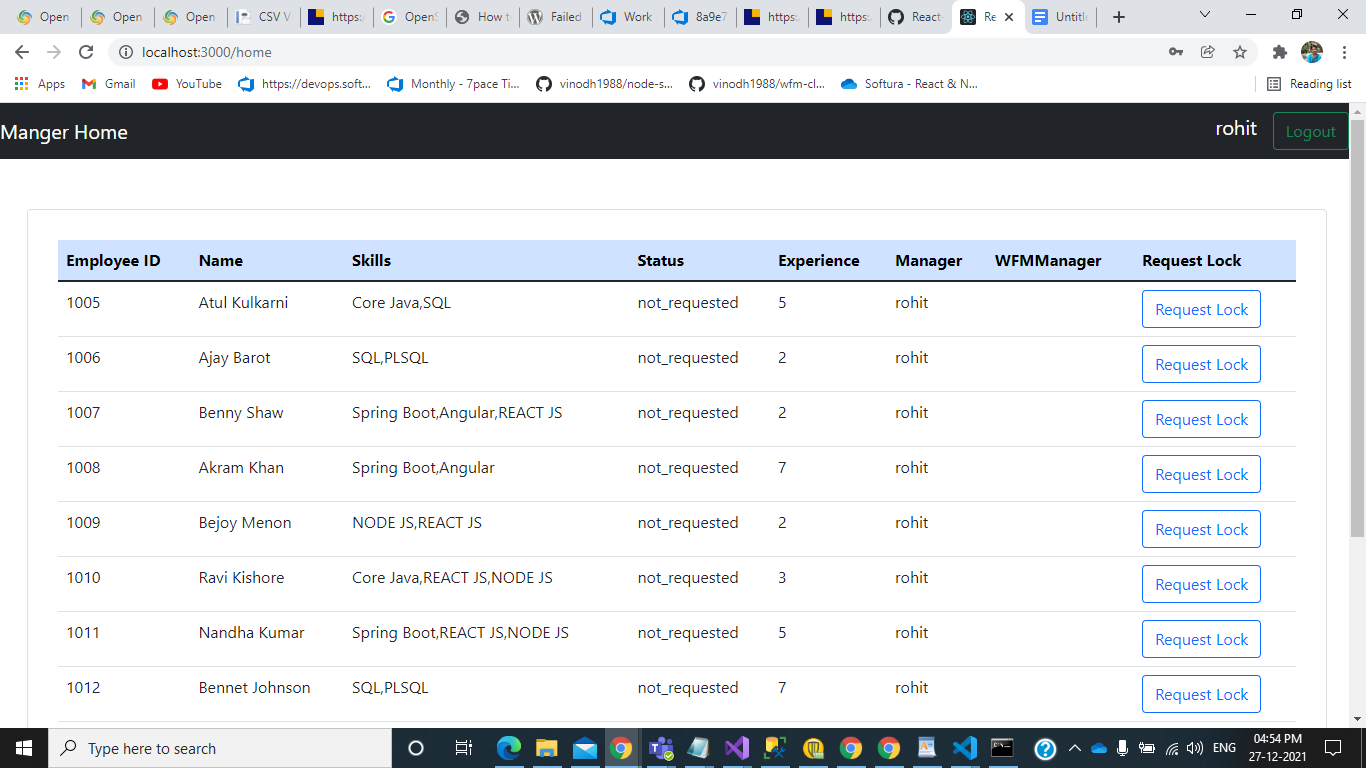
**–Lakshmi priya**

**Screenshots:**

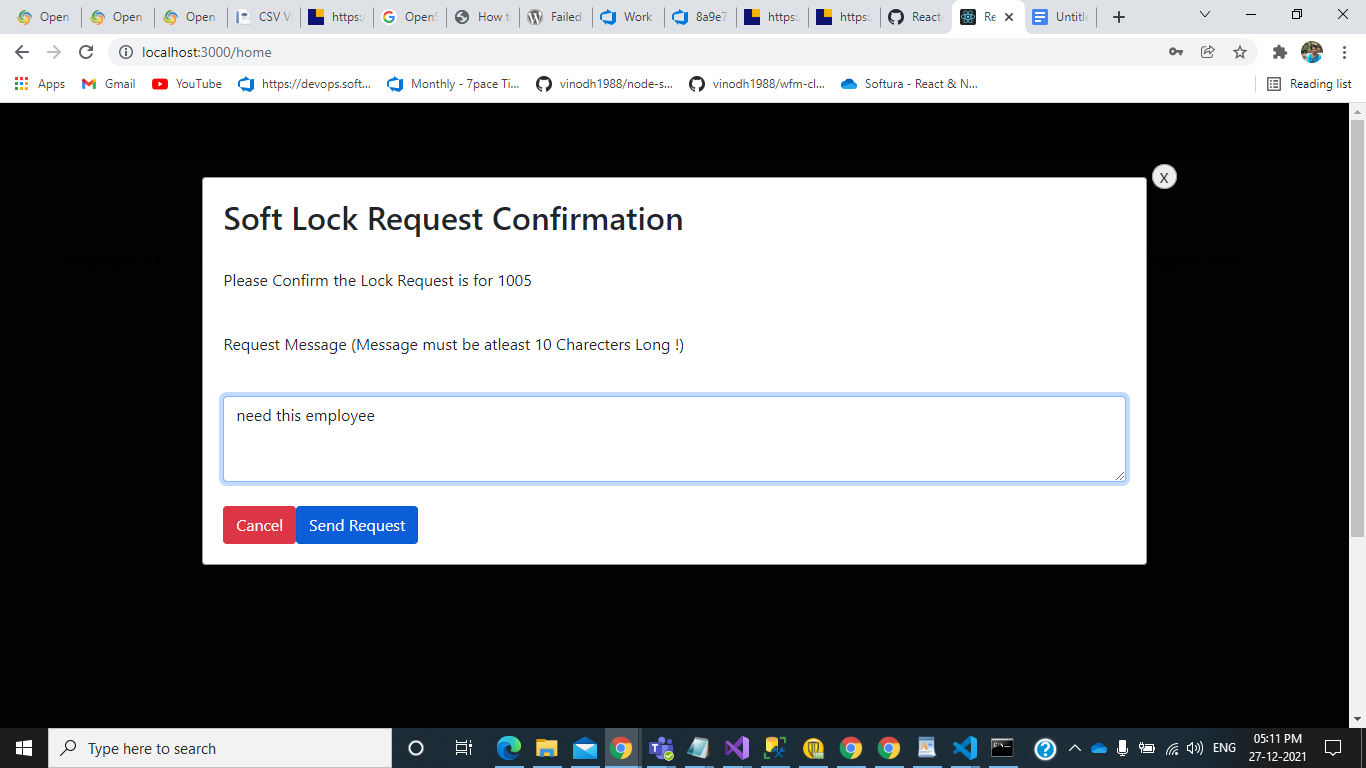
**Login:**



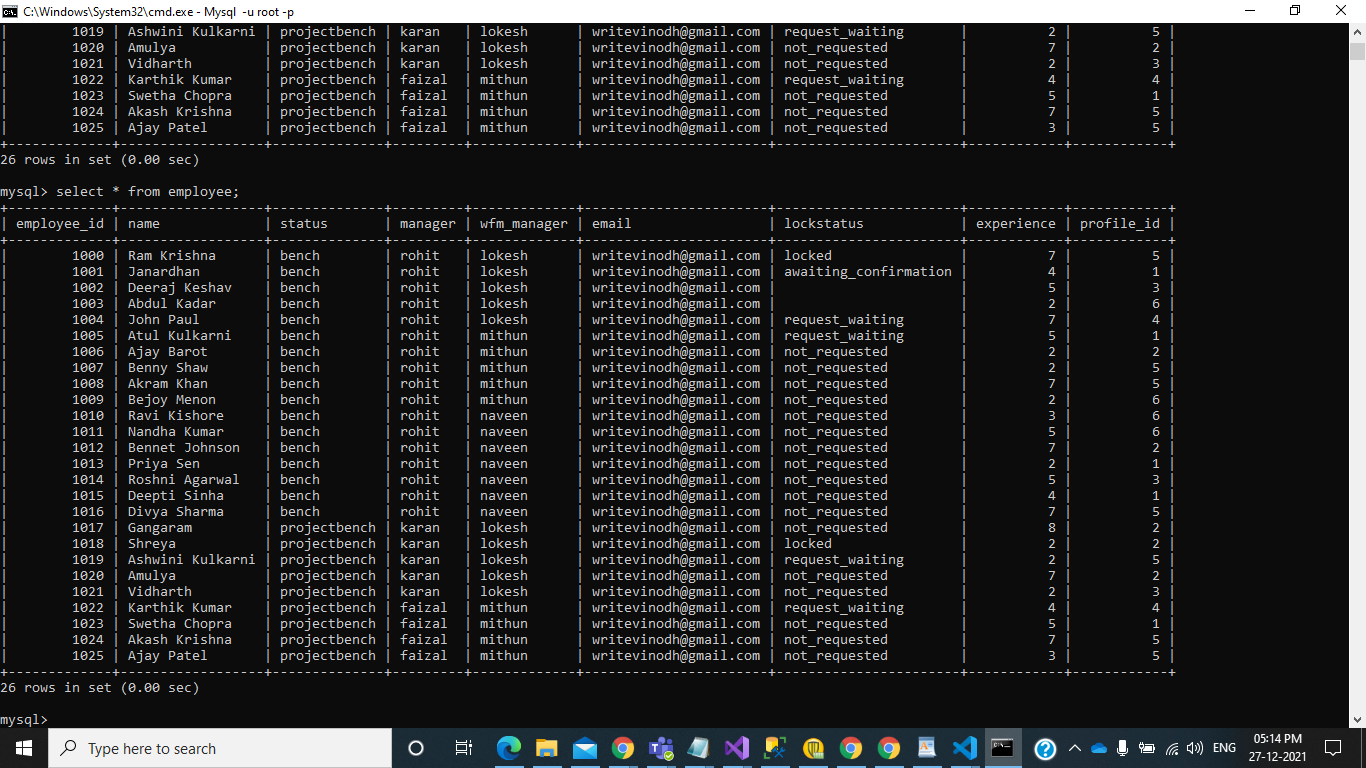
**Manager Home screen:**

****

This is the manager home screen in which it lists the employees under the manager who have logged in. I have written the select query in node api such that it lists only the employees whose manager name is the username who have logged in. Then when you click the request lock button next the corresponding employee a popup comes up in the you can write a request message and send the request to the corresponding wfm manager on clicking the send button.

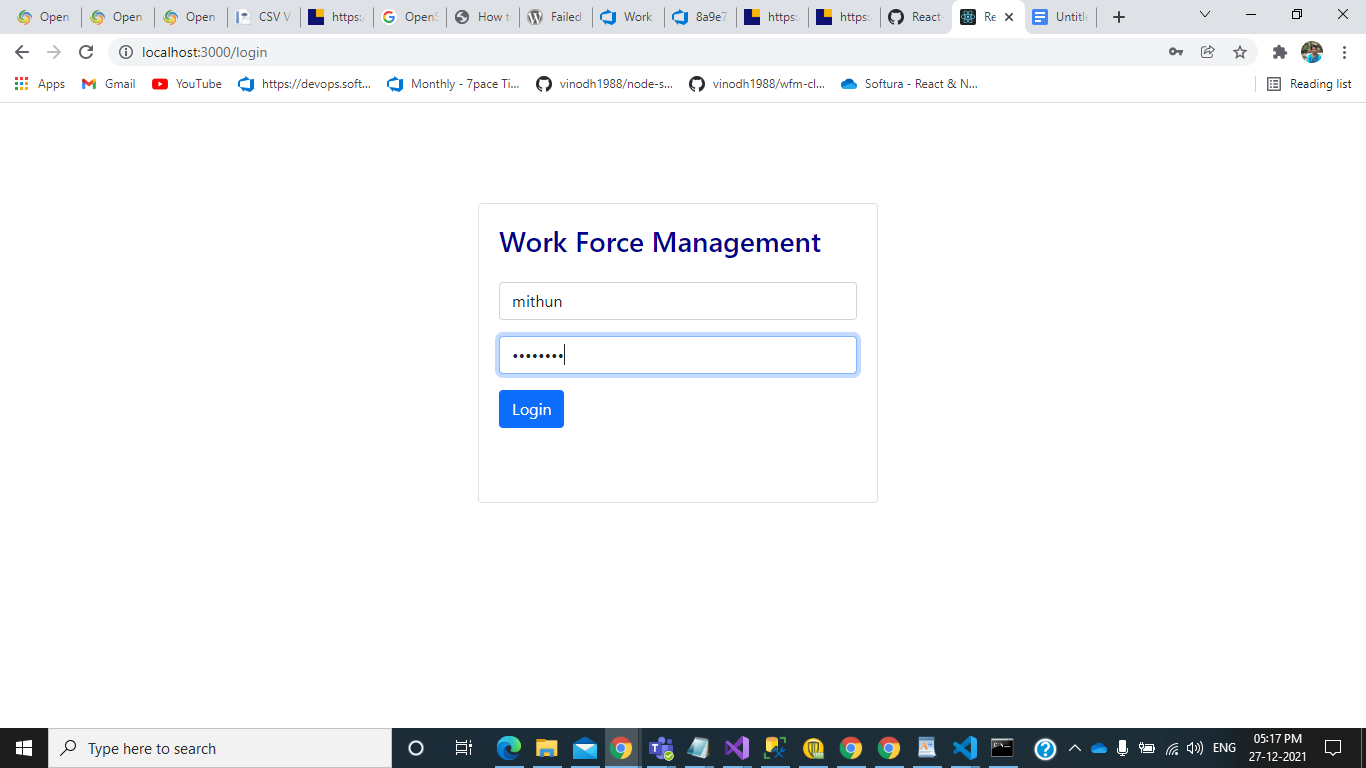


Once you send request for an employee his/her lockstatus will change to request waiting

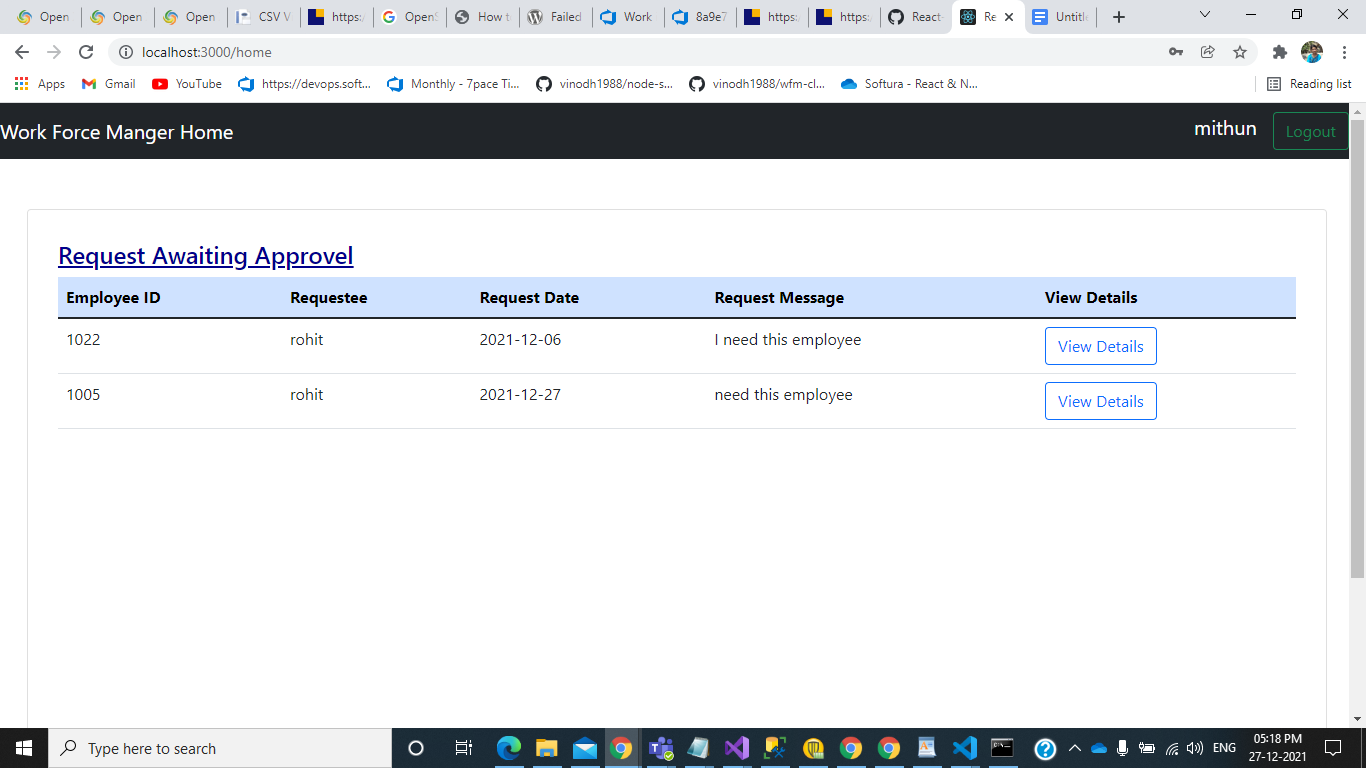


As you can see here i have requested for the employee 1005 and the lockstatus has changed to request\_waiting and also the employees details will get inserted into the softlock table.

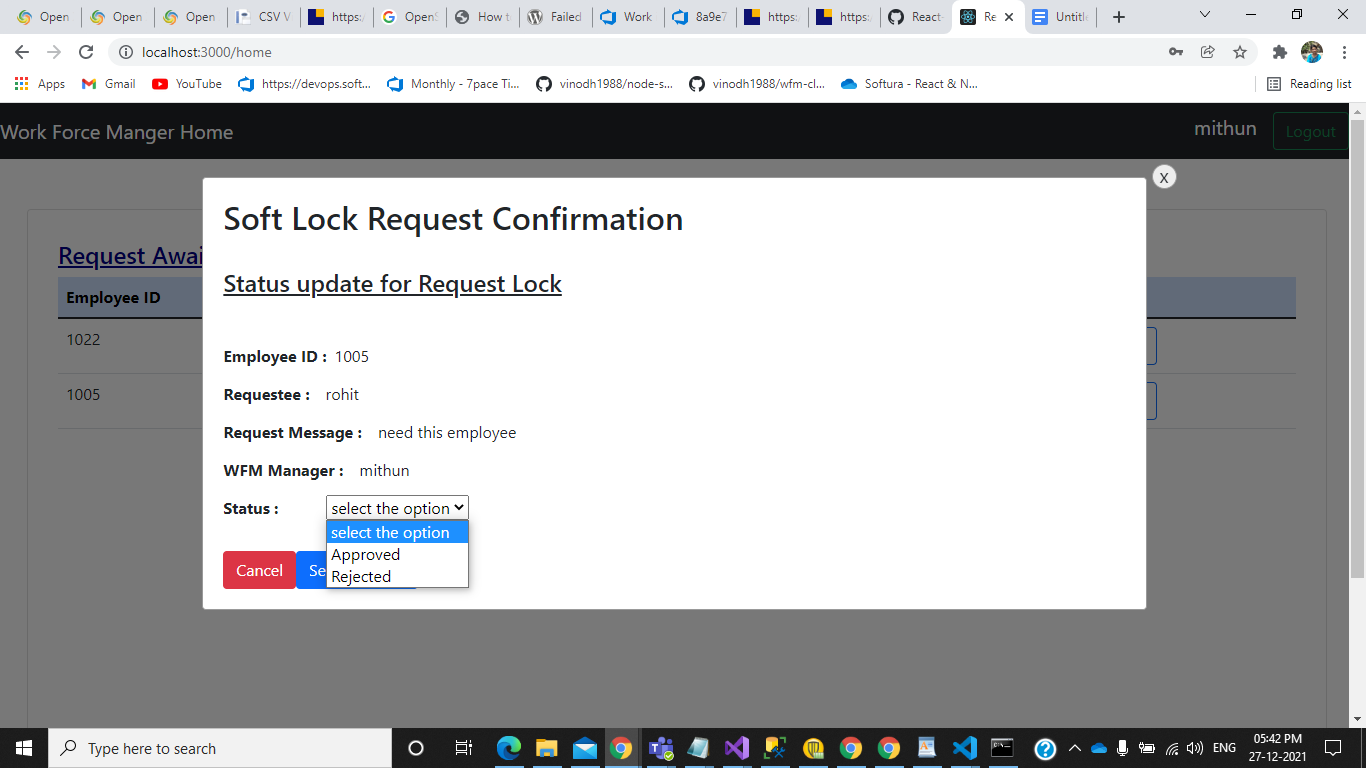
**WFM\_Manager Login:**

****

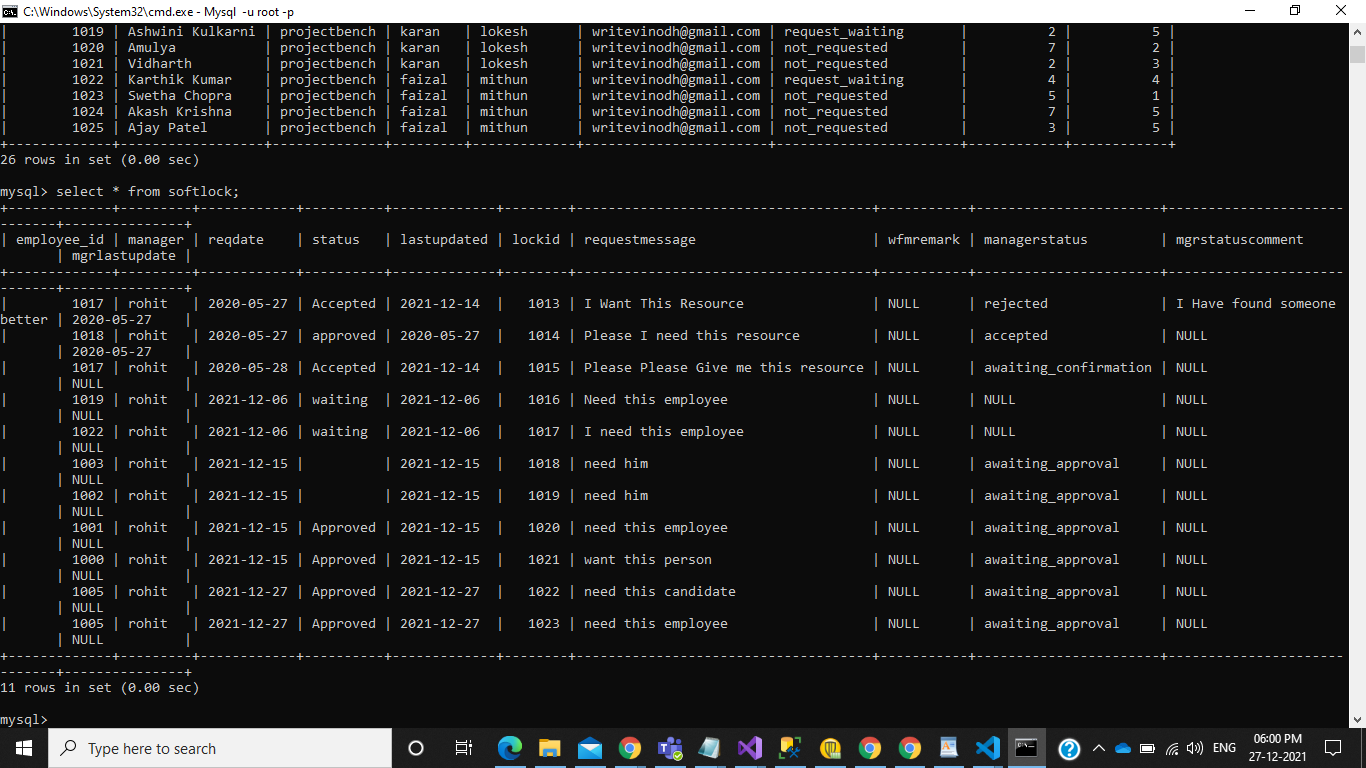
**WFM\_Manager Home screen:**

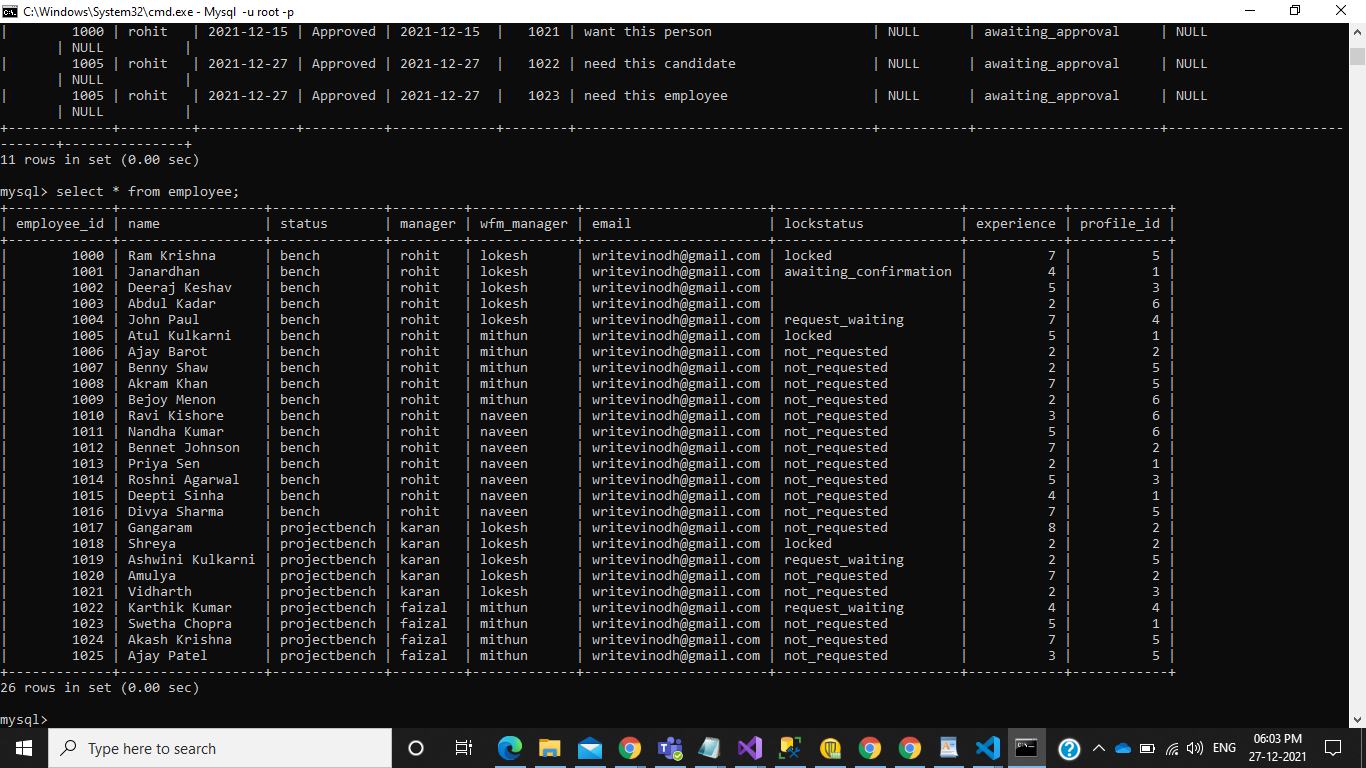


As the same case in manager login here also only the employees with corresponding wfm manager are listed from the softlock table. And there is a button against each employee view details when you click it a pop appears with the employees details employeeid, requestee request date and the request message and also status for which there will be a dropdown the wfm manager can select the accept or reject option from it.

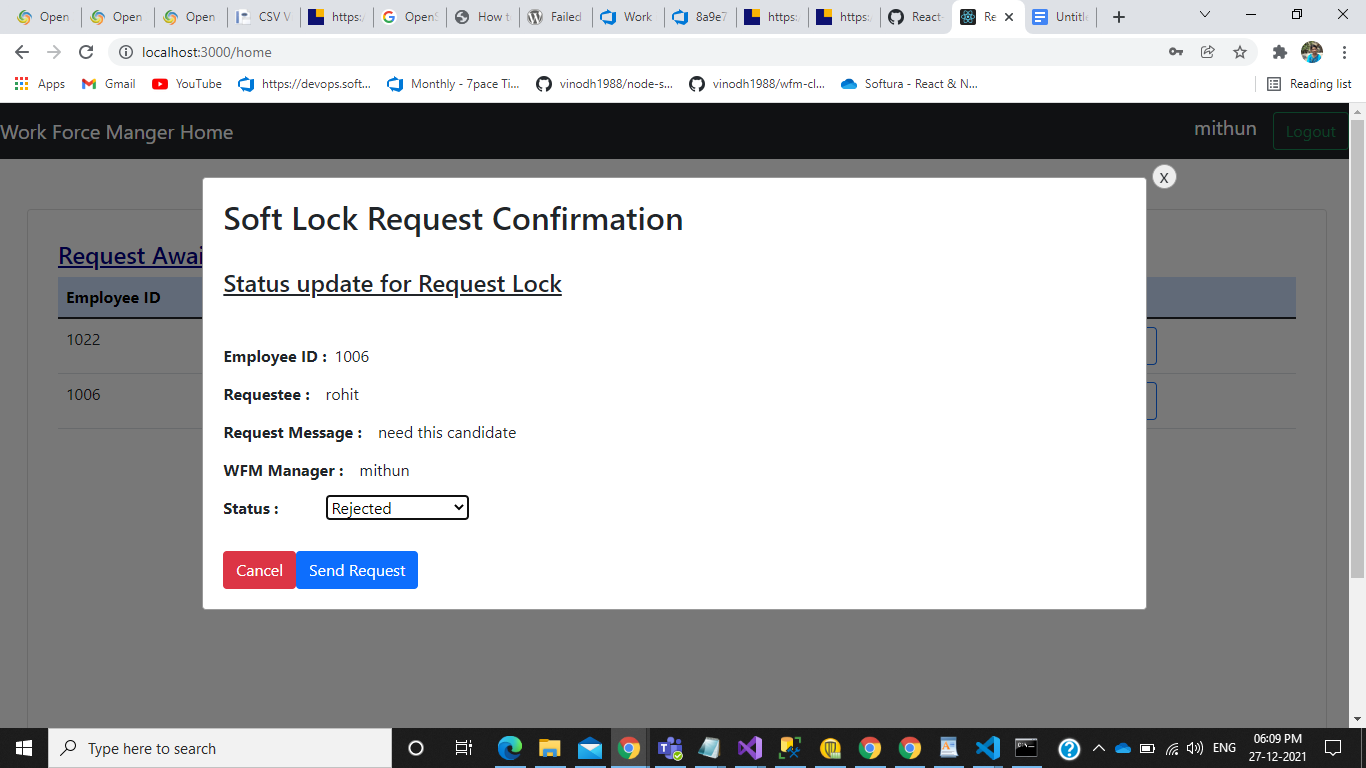


When the wfm manager approves the employee the status in the softlock table changes from waiting to approved.

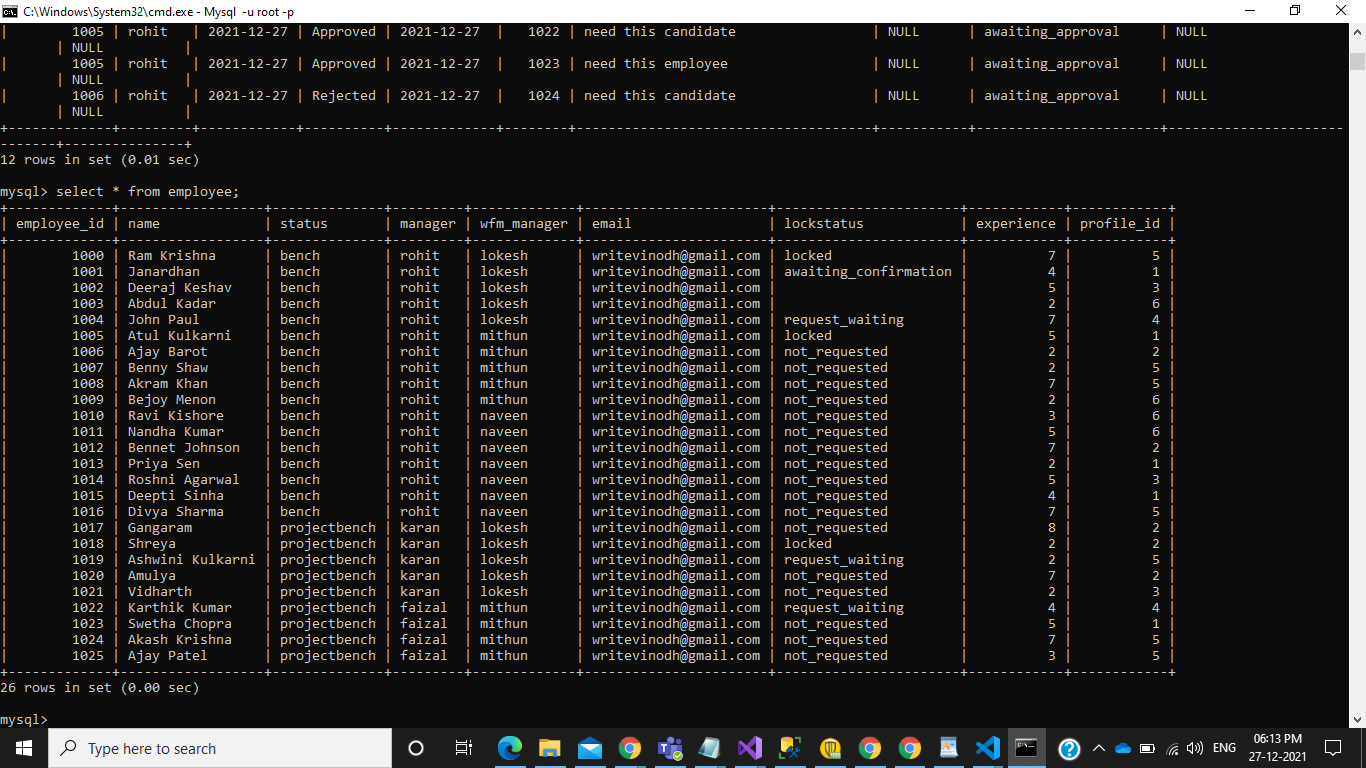




As you can see lockstatus in the employee table has also changed to locked for the employee 1005. In case the wfm manager rejects the employee



The employee status will be updated in the softlock table and also in employee table the status will be set back to not\_requested.



Here you can see that lockstatus for 1006 is set back to not\_requested

Also i have given logout buttons in both wfm manager and manager home pages.

**Solution approach:**

In case of solution approach in Manager’s home.tsx i have called the employee.tsx in which the api is called and the table is displayed. Same as that of in manager in wfm manager’s home.tsx I have called Empreq.tsx in which the api call occurs and the table is displayed. And the logout has been implemented in the home.tsx once the logout button is clicked it empties the localstorage and redirects to login page.

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>

<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;

WFMManager: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">WFMManager</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.WFMManager}</td>

<td>

<button type="button" className="btn btn-outline-primary" id={empid} onClick={togglePopup}>Request Lock</button>

{

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;

WFMManager: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 Approvel</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 :&emsp;</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