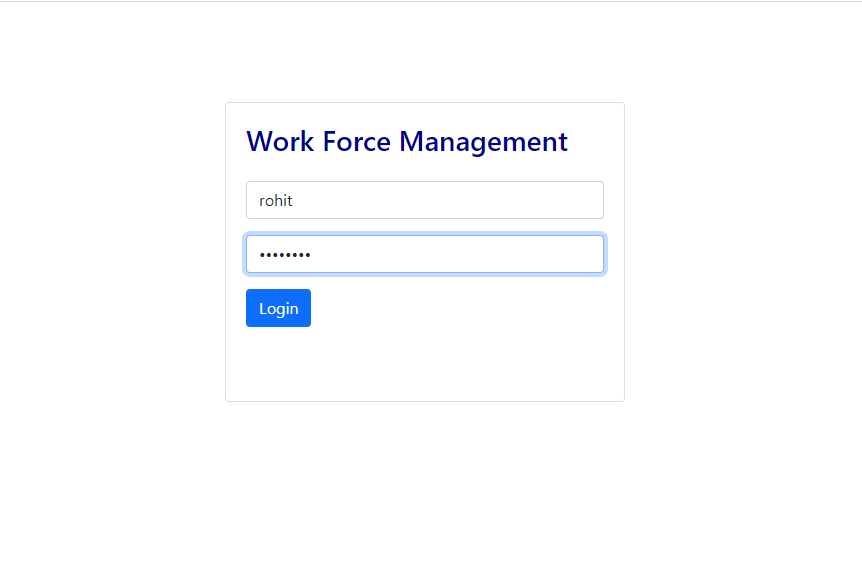
# APPROCH NOTE AND OUTPUT SCREENSHOTS

## MANAGER HOME PAGE:

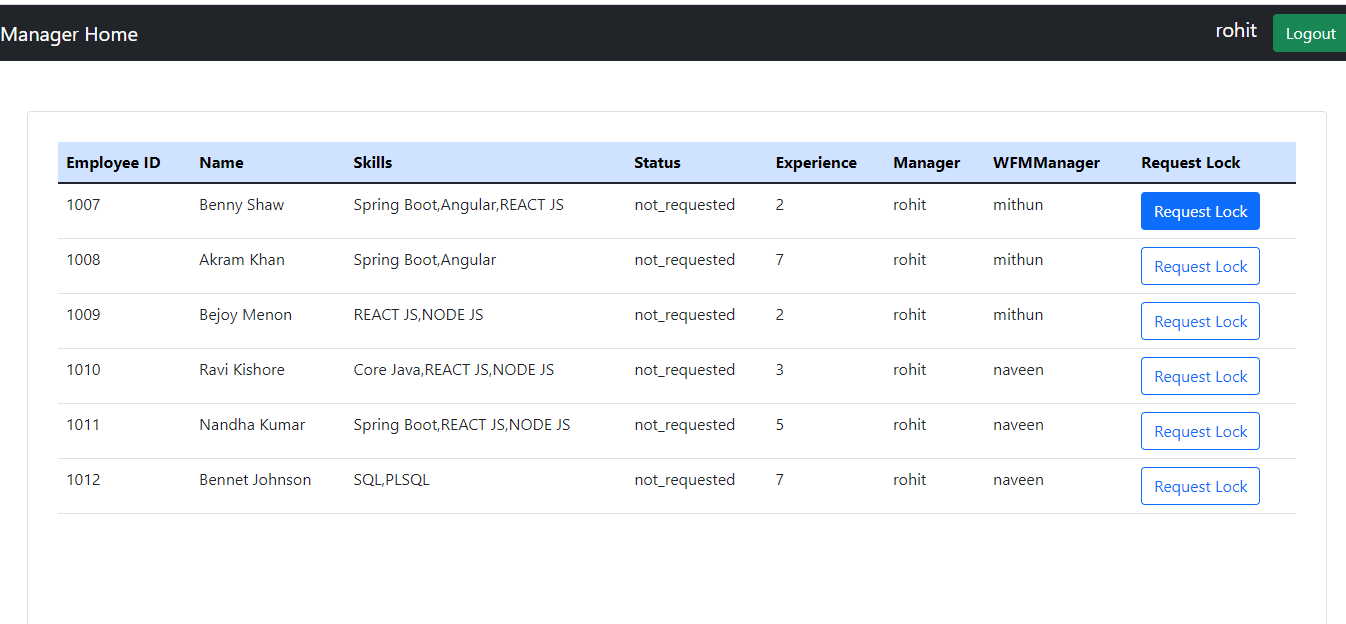
* When the Manager Login to the page, the page will list all the employees under the corresponding selected manager with column values Employee ID, Name, Skill, Status, Experience, WFM manager and Request lock Column. The Request Lock Column will have a button for lock Request for the corresponding WFM manager of the current manager.
* **Manager Login Page:**

In this example I have logged in as ROHIT with corresponding password.



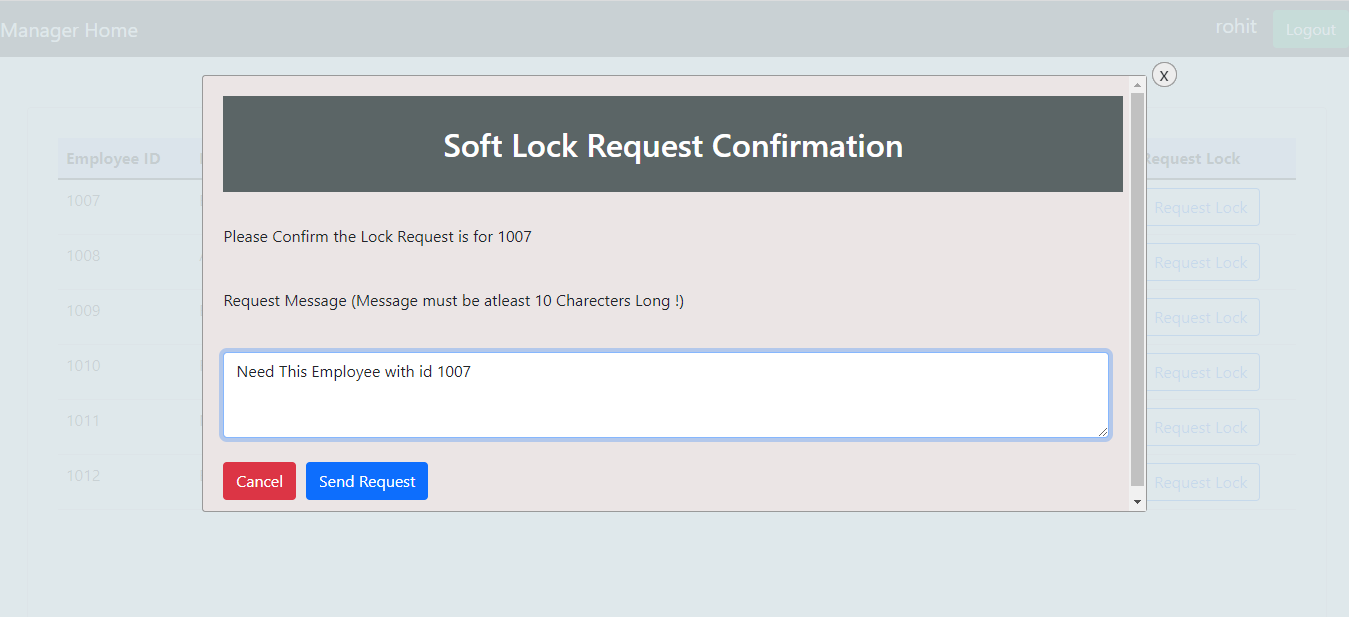
* **Manager Home Page:**

In this example I have selected employee id 1007 and requesting for a lock for the corresponding WFM manager.



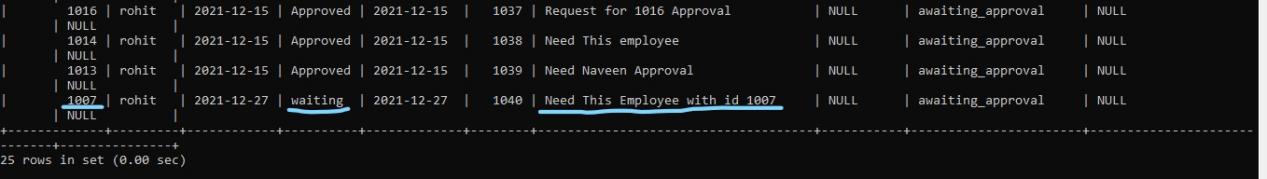
* **Soft Lock Request Confirmation:**

Once the Send Request Button is clicked the employee id manager details and the request message and request date will be stored in the soft lock table.



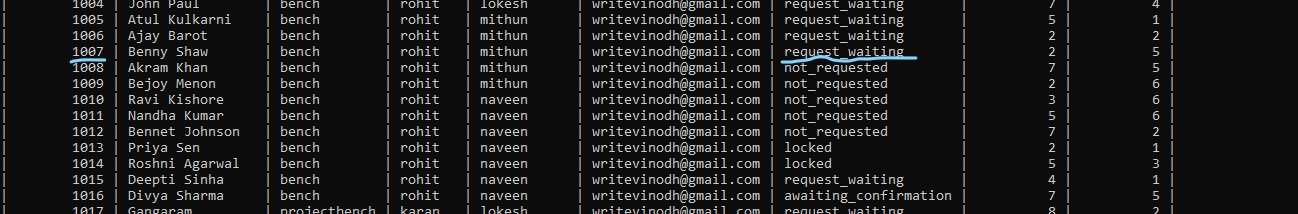
* **Soft lock Table:**

Using WFM database under Soft lock table the request by the manager will be displayed as waiting till the corresponding WFM manager approve or reject the request.



* **Employee Table:**

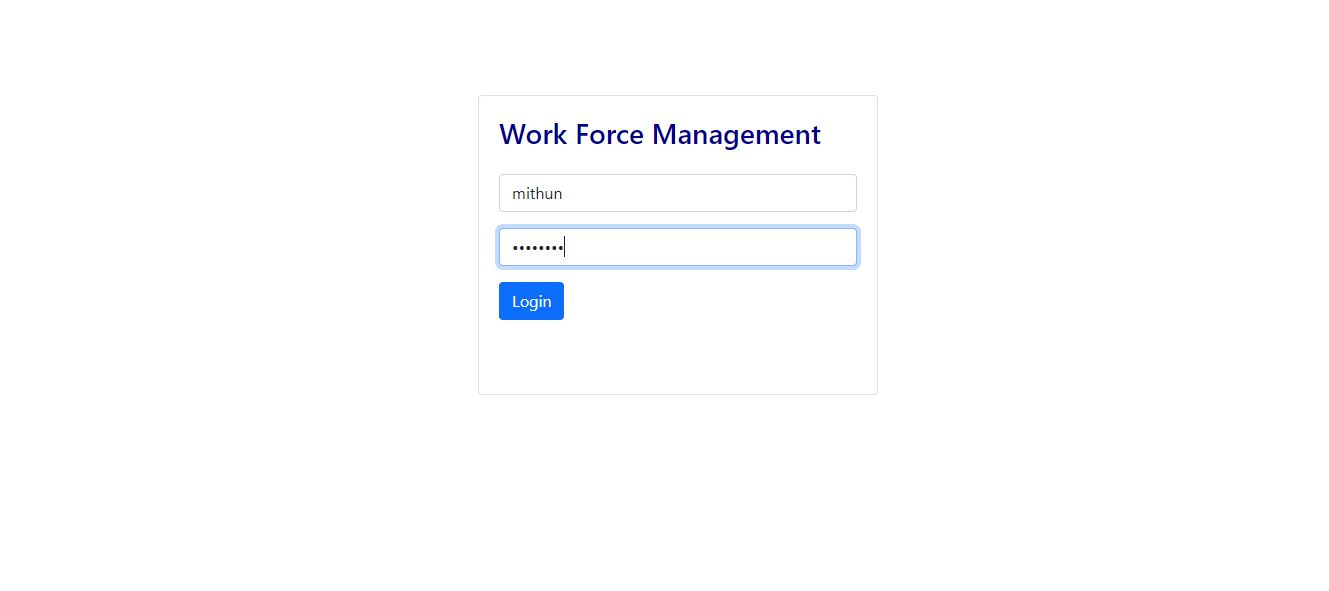
Using WFM database under Soft lock table the request by the manager will be displayed as request\_waiting till the corresponding WFM manager approve or reject the request.



### WORK FORCE MANAGER HOME PAGE:

### WFM Login Page:

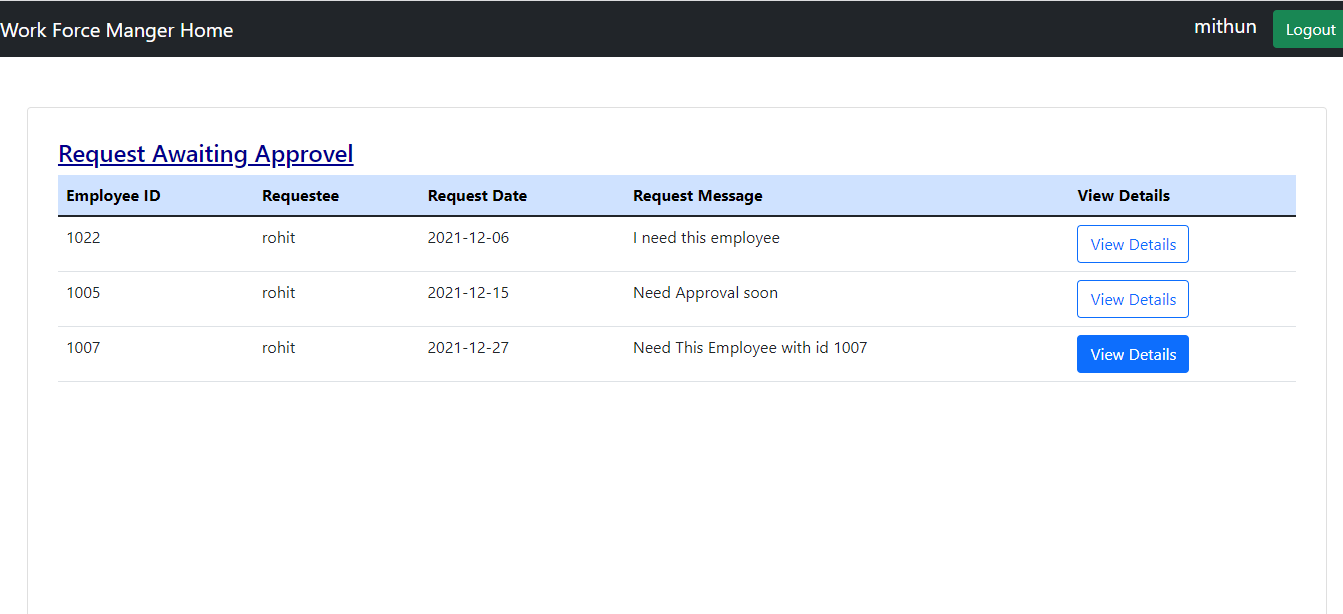
In this example I have logged in as MITHUN with corresponding password.



* **Work force manager Home page:**

This page will display a list of details of the manager and the employee under current WFM manager with column values Employee ID, Requestee, Request Date, Request Message and View Details.

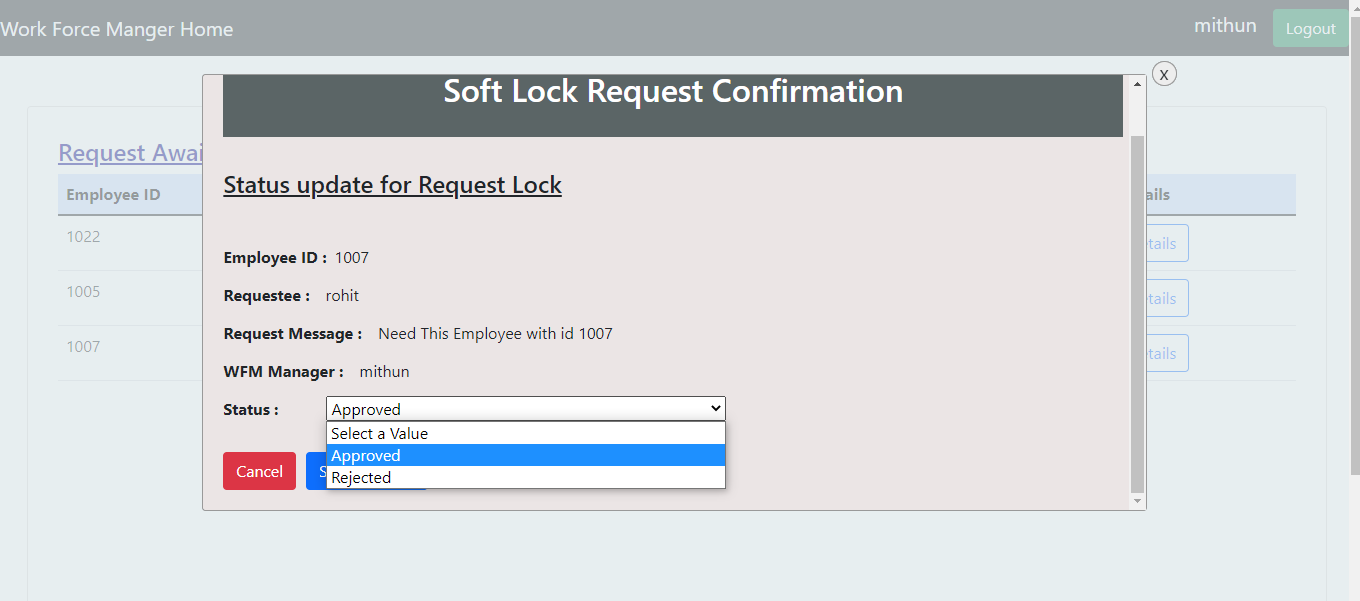
* The Request send will now be displayed in this list.
* Once a particular row button is clicked( here as a example I have clicked Employee ID=1007) it will pop up a new screen.

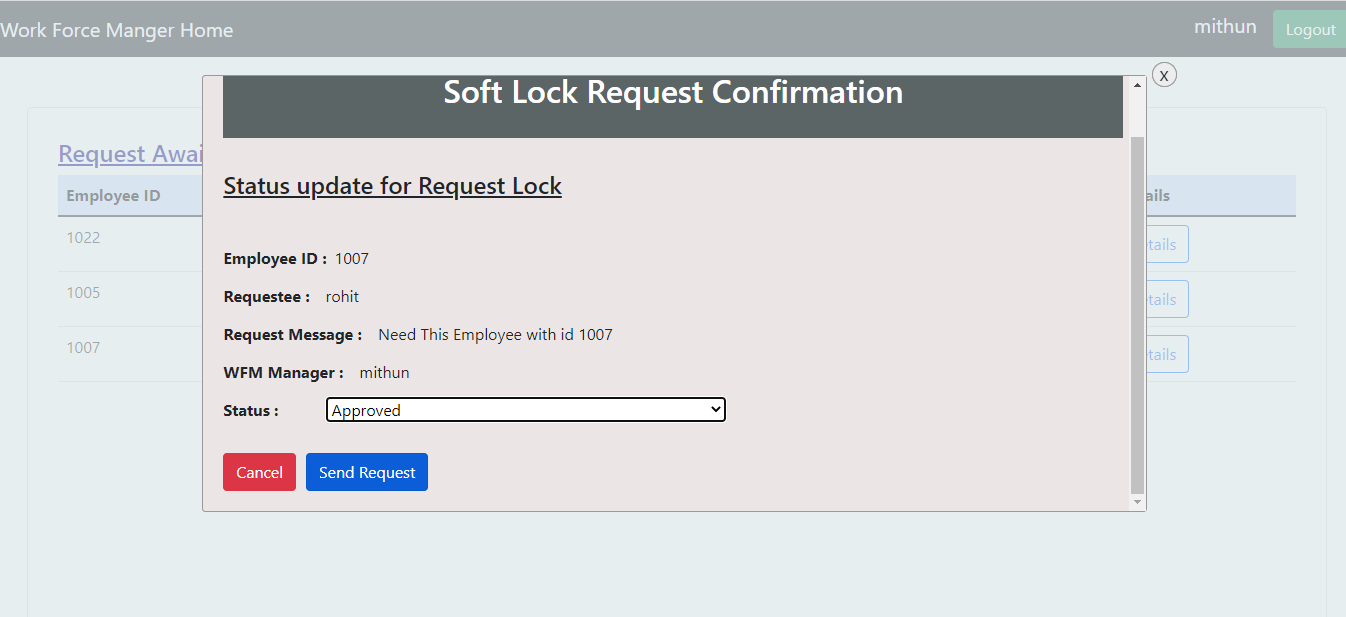


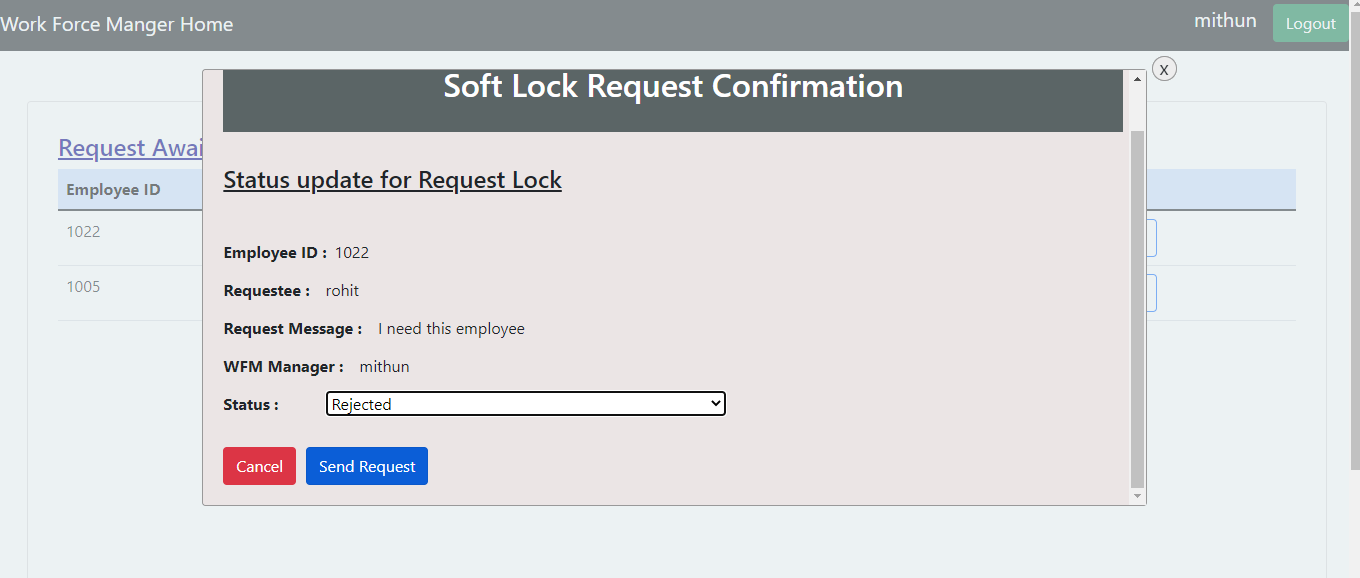
* **Soft Lock Request Confirmation Popup:**

This will display the details of the Request Awaiting for Approval row details which will not be able to be edited by the WFM manager.

* The WFM manager can Approve or Reject the Request from the manager.
* Based on the input given by the WFM manager the status will be updated both in the employee and the soft lock table as Approved and locked (if Approved) or rejected and not Requested( if Rejected).

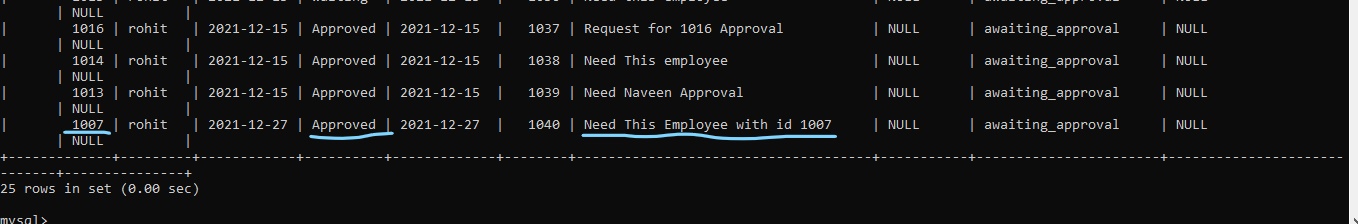






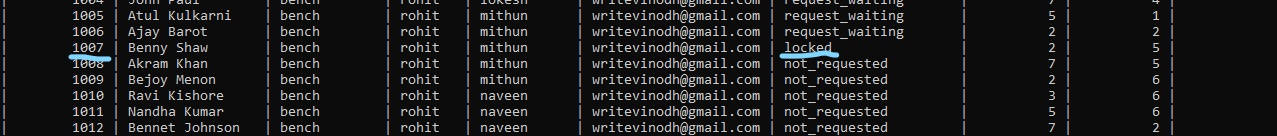
* **Soft lock Table: (If Approved (ID:1007))**

Using WFM database under Soft lock table the request by the WFM manager will be displayed as Approved .



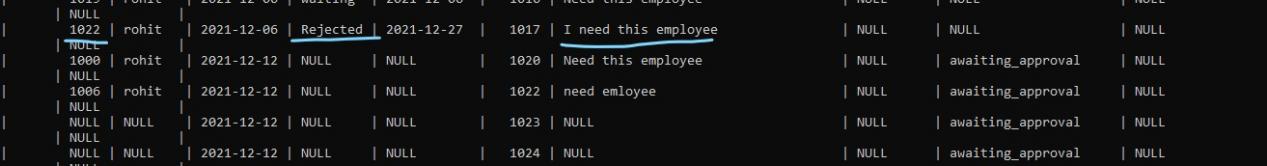
* **Employee Table:(If Approved(ID:1007))**

Using WFM database under Employee table the request by the WFM manager will be displayed as Locked.



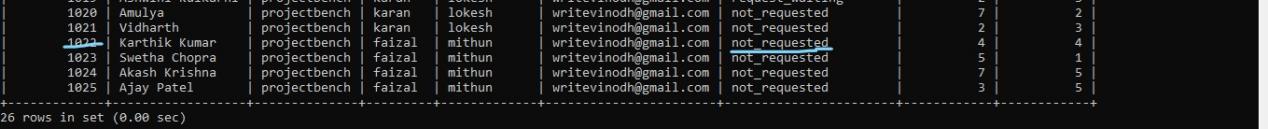
* **Soft lock Table: (If Rejected(ID:1022))**

Using WFM database under Soft lock table the request by the WFM manager will be displayed as Rejected .



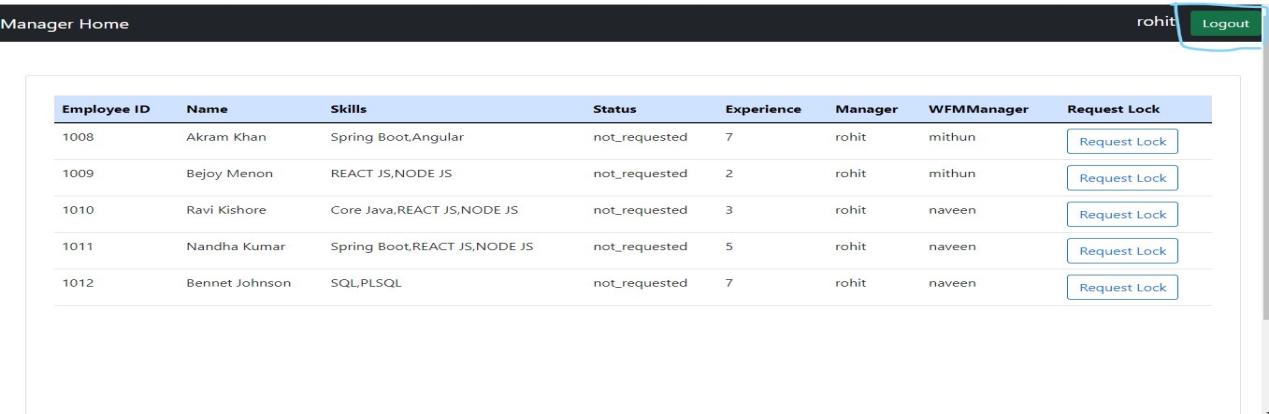
* **Employee Table:(If Rejected(ID:1022))**

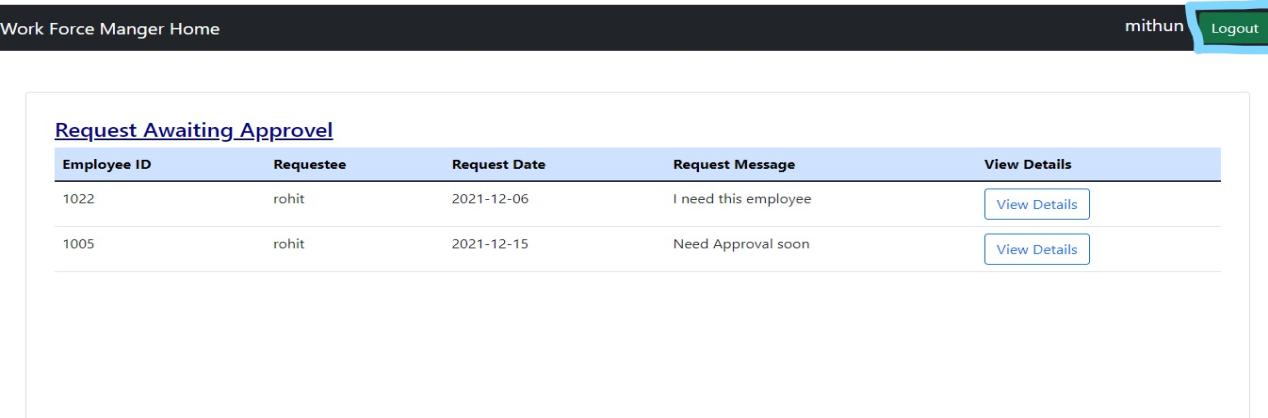
Using WFM database under Employee table the request by the WFM manager will be displayed as not\_requested.



### LOGOUT:

* The Logout button at the top right corner will allow the WFM manager/manager to Logout from the Page.





**SOURCE CODE**

**WFM-Client-React:**

**MANAGER PAGE:**

**Home.tsx:**

import Login from "../Login";

import Employee from "./Employee"

const ManagerHome=()=>{

const username = localStorage.getItem("username")

const name = localStorage.getItem("name")

const employeeid = localStorage.getItem("employeeid")

console.log(employeeid)

console.log(name)

return (

<div>

<div>

<nav className="navbar navbar-dark bg-dark justify-content-between">

<a className="navbar-brand" > Manager Home </a>

<form className="form-inline my-2 my-lg-0">

<a className="navbar-brand" >{username}</a>

<button className="btn btn-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 '../pop';

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 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/employee/employees")

setPeopleData(response.data)

}

catch(e){

setPeopleData([])

}

}

async function addDetails(){

try{

console.log(id)

const response =

await axios.put("http://localhost:8000/employee/updateemployee",{employeeid : id});

if(response.status == 200){

console.log(id,reqmessage)

const res = await axios.post("http://localhost:8000/employee/insertsoftlock",

{

employeeid:id,manager:localStorage.getItem("username"),responsemessage: reqmessage});

}

peopleRead();

Setreqmessage('');

}

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

**WFM Manager:**

**Home.tsx:**

import Login from "../Login";

import EmpRequest from "./EmpRequest"

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-success my-2 my-sm-0" onClick={()=>

{if(username)

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

</form>

</nav>

</div>

<div>

<EmpRequest></EmpRequest>

</div>

</div>

)

}

export default WFMHome

**EmpRequest.tsx:**

import { useState,useEffect,useRef } from "react";

import { Redirect } from "react-router";

import axios from "axios";

//import Popup from 'reactjs-popup';

import Popup from '../pop';

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 [peopleData,setPeopleData] =useState([])

//const [accepted,setaccepted]=useState("Accepted");

const[status,setStatus]=useState('')

const WFMmanager=localStorage.getItem("username")

function togglePopup(event:any){

setIsOpen(!isOpen);

if(isOpen!==true){

const empid=event.target.id;

Setid(empid);

}

}

async function peopleRead(){

try{

let response =

await axios.get("http://localhost:8000/employee/softlocktable")

setPeopleData(response.data)

}

catch(e){

setPeopleData([])

}

}

async function addEmployee() {

try {

const res = await axios.put("http://localhost:8000/employee/updateSoftlock",

{ employeeid: parseInt(Array[0]),status:status});

peopleRead();

}

catch (e) {

console.log("Error");

}

}

useEffect(()=>{

peopleRead()

},[])

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>

{

peopleData.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" value={status} onChange={(e)=>setStatus(e.target.value)}required>

<option value="Select a Value" >Select a Value</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

**Node-service-app:**

**index.js:**

var express=require("express");

var passport = require('passport');

var users=require('./routes/userroutes')

var employee=require('./orm/mysqlroutes')

var app=express();

const cors=require("cors")

app.use(passport.initialize());

app.use(cors())

app.use(express.urlencoded({extended:true}))

app.use(express.json())

app.use("/users",users)

app.use("/employee",employee)

app.listen("8000",function(){

console.log("Server running on port 8000")

})

**mysqlroutes.js:**

const model=require('../orm/model')

const route=require("express").Router();

const sequelize = require('./connection');

const { QueryTypes } = require('sequelize');

const employees=model.employees;

const softlock=model.softlock;

const passport=require('passport')

route.get("/employees",async function(request,response){

try{

sequelize.query('select e.employee\_id as EmployeeID,e.name as Name,group\_concat(s.name) as Skills,e.lockstatus as Status,e.experience as Experince, e.manager as Manager, e.wfm\_manager as WFMManager from employee e join skillmap sk on e.employee\_id=sk.employee\_id join skills s on sk.skillid=s.skillid where e.lockstatus="not\_requested" group by e.employee\_id;',

{model:employees ,mapToModel:true,type: sequelize.QueryTypes.SELECT}

).then(function(employees){

response.json(employees)

})

}

catch(e){

console.log(e)

response.status(500)

}

})

//select s.employee\_id as EmployeeID,s.manager as Name,s.reqdate as ReqDate,s.requestmessage as ReqMessage,s.status as Status from softlock s join employee e on e.employee\_id=s.employee\_id where e.wfm\_manager like :managername and e.lockstatus='request\_waiting' and s.status='waiting'

route.get("/softlocktable",async function(request,response){

try{

sequelize.query("select s.employee\_id as EmployeeID, s.manager as Requestee,s.reqdate as RequestDate,s.requestmessage as RequestMessage,s.status as Status,e.wfm\_manager as WFMManager from softlock s join employee e on s.employee\_id=e.employee\_id where e.lockstatus='request\_waiting' and s.status='waiting';",

{model:softlock ,mapToModel:true,type: sequelize.QueryTypes.SELECT}

).then(function(softlock){

response.json(softlock)

})

}

catch(e){

console.log(e)

response.status(500)

}

})

route.put("/updateemployee", async function (request, response) {

try{

let employeeid = request.body.employeeid;

await sequelize.query("update employee set lockstatus='request\_waiting' where employee\_id=:employeeid;",

{replacements: {employeeid: employeeid},model:employees,mapToModel: true,type:sequelize.QueryTypes.UPDATE}

).then(function(employees){

response.status(200).send("record updated");

})

}

catch(e){

console.log(e)

response.status(500)

}

})

route.post("/insertsoftlock",async function(request,response){

try{

let employeeid = request.body.employeeid;

let manager = request.body.manager;

let responsemessage = request.body.responsemessage;

await sequelize.query("insert into softlock (employee\_id,manager,reqdate,lastupdated,status,requestmessage) values(?,?,CurDate(),CurDate(),'waiting',?)",

{replacements: [employeeid,manager,responsemessage],model:softlock,mapToModel: true,type:sequelize.QueryTypes.INSERT}

).then(function(){

response.status(200).send("record inserted");

})

}

catch(e){

console.log(e)

response.status(500);

}

})

route.put("/updateSoftlock",async function(request,response){

try{

let employeeid = request.body.employeeid;

let status=request.body.status;

await sequelize.query("update softlock set status=:status ,lastupdated =CURDATE() where employee\_id = :employeeid",

{replacements: {employeeid: employeeid,status:status}, model:employees,mapToModel: true,type:sequelize.QueryTypes.UPDATE}

).then(function(){

response.status(200).json()

})

if(status=="Approved")

{

await sequelize.query("update employee set lockstatus = 'locked' where employee\_id = :employeeid",

{replacements: {employeeid: employeeid}, model:employees,mapToModel: true,type:sequelize.QueryTypes.UPDATE}

).then(function(){

response.status(200).json()

})

}

else

{

await sequelize.query("update employee set lockstatus = 'not\_requested' where employee\_id = :employeeid",

{replacements: {employeeid: employeeid}, model:employees,mapToModel: true,type:sequelize.QueryTypes.UPDATE}

).then(function(){

response.status(200).json()

})

}

}

catch(e){

console.log(e)

response.status(500);

}

})

module.exports=route;