React and Node JS Assessment

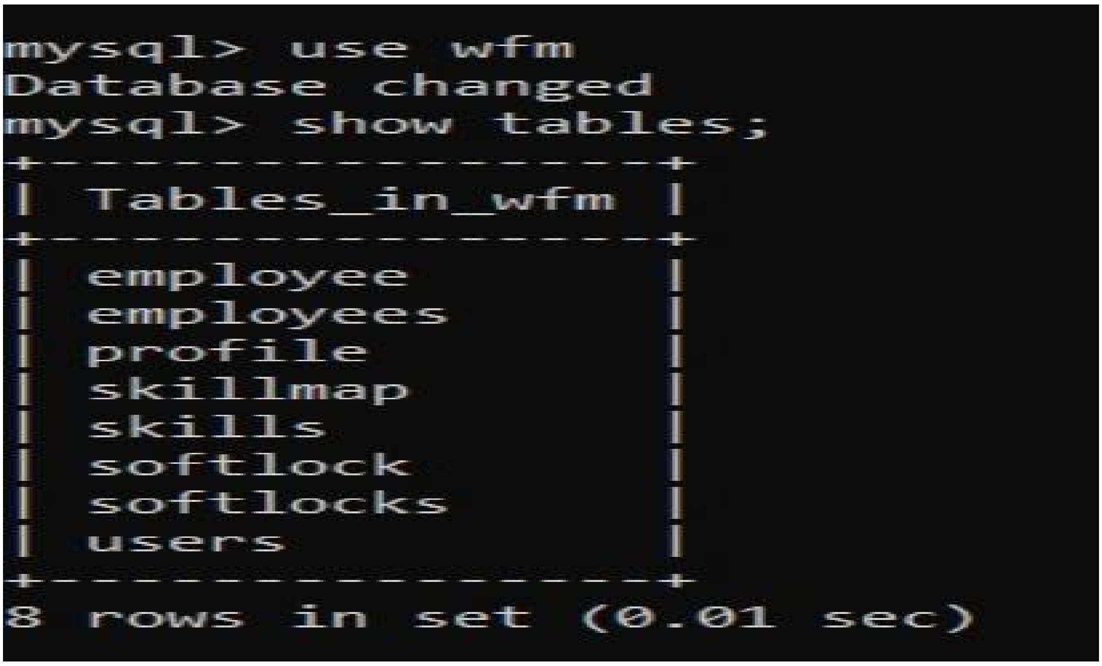
Implementation

Configuring database into the local machine:

Step1: After installation go to command prompt and type the following command “ MySql -u root -p ” which will asks to enter the password.

Step2: After entering password for adding database into the local machine use this command “ Create database wfm ”. It will create a new database. After creating new database use this command “ Mysql -u root -p –database wfm < backup.sql ”

Step3: For checking the data base tables we can use “ Show tables ” command .

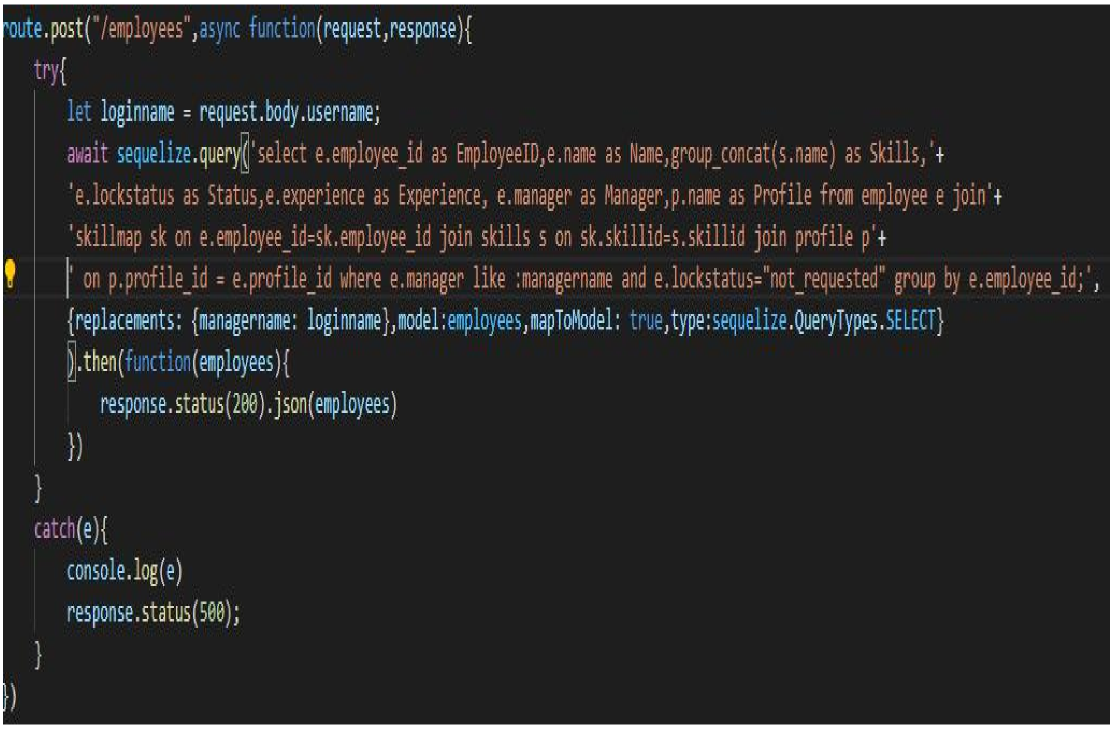


```
mysql> use wfm
Database changed
mysql> show tables;
+-----------------+
| Tables_in_wfm |
+-----------------+
|  employee       |
|  employees      |
|  profile        |
|  skillmap       |
|  skills         |
|  softlock       |
|  softlocks      |
|  users          |
+-----------------+
8 rows in set (0.01 sec)
```

In order to print data in the React app , we need to have API calls for printing employee data in their respective logins.

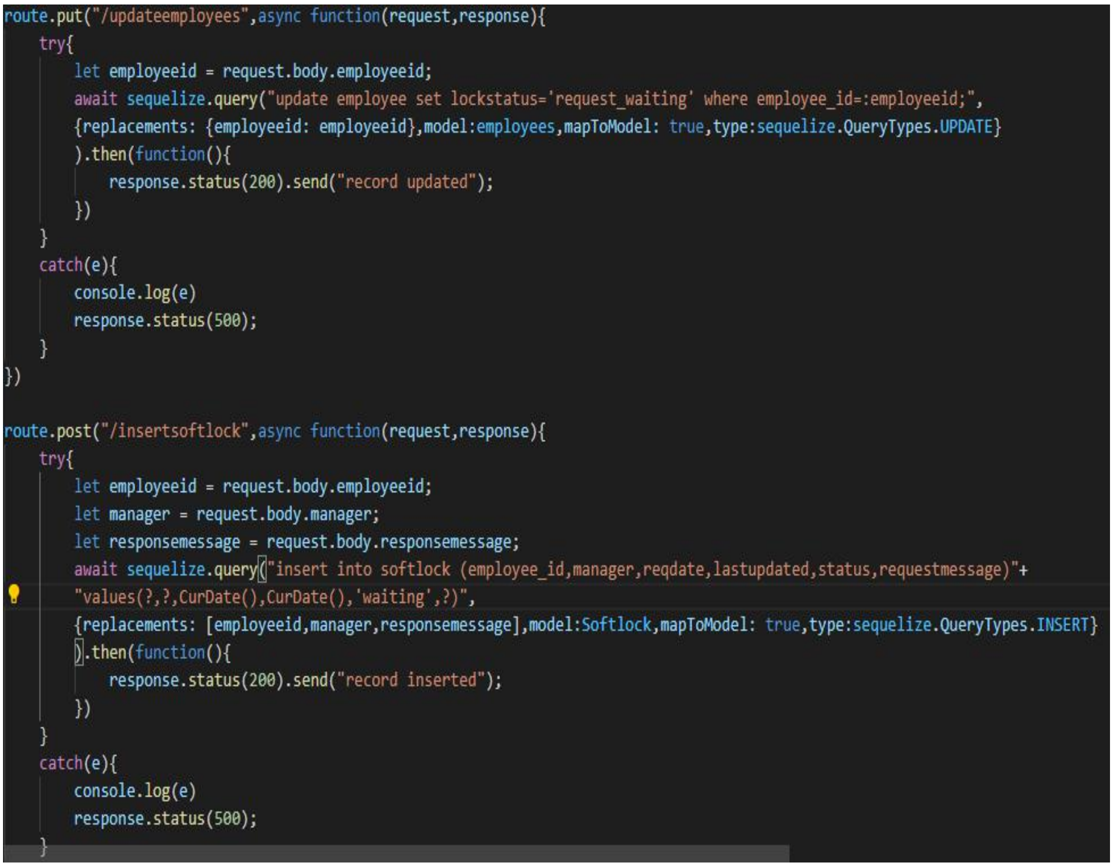
API CALLS

1. API call for printing employees under respective manager.



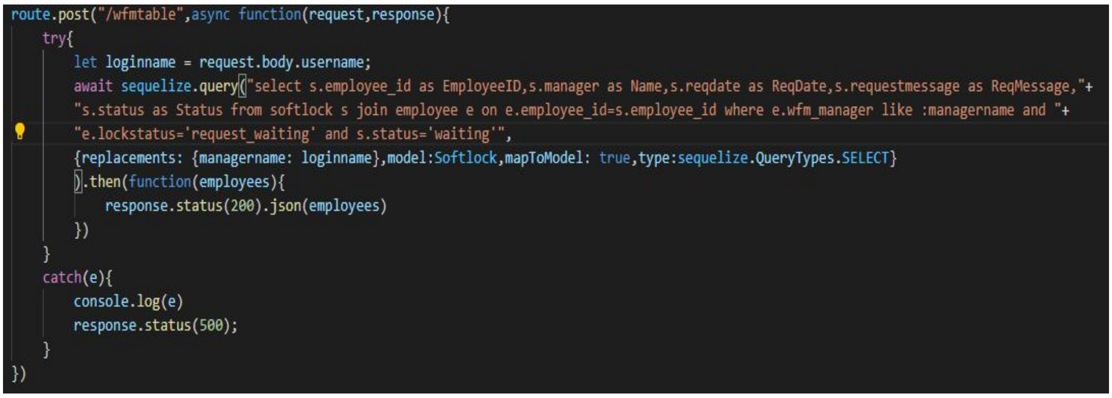
```
route.post("/employees",async function(request,response){
    try{
        let loginname = request.body.username;
        await sequelize.query('select e.employee_id as EmployeeID,e.name as Name,group_concat(s.name) as Skills,'+
        'e.lockstatus as Status,e.experience as Experience, e.manager as Manager,p.name as Profile from employee e join'+
        'skillmap sk on e.employee_id=sk.employee_id join skills s on sk.skillid=s.skillid join profile p'+
        ' on p.profile_id = e.profile_id where e.manager like :managername and e.lockstatus="not_requested" group by e.employee_id;',
        {replacements: {managername: loginname},model:employees,mapToModel: true,type:sequelize.QueryTypes.SELECT}
        ).then(function(employees){
            response.status(200).json(employees)
        })
    }
    catch(e){
        console.log(e)
        response.status(500);
    }
})
```

2. API call after the request from Manager - updating employees table and insert softlock Table.



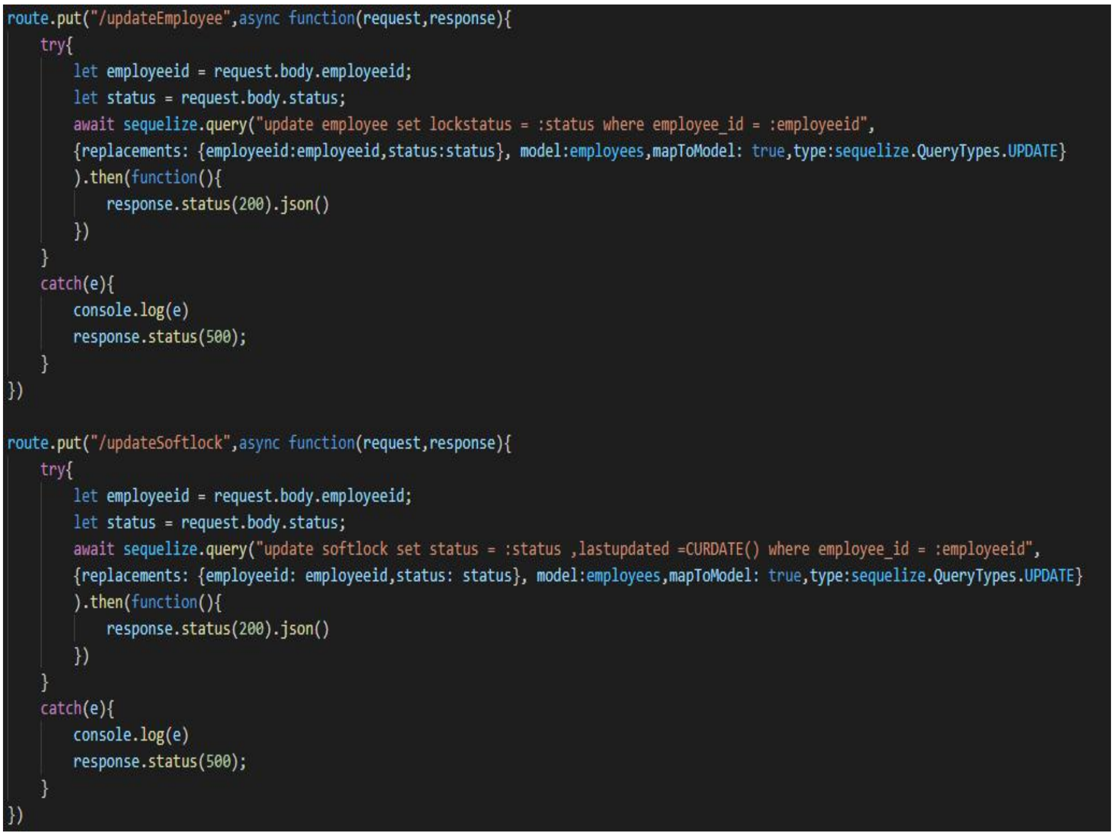
```
route.put("/updateemployees",async function(request,response){
    try{
        let employeeid = request.body.employeeid;
        await sequelize.query("update employee set lockstatus='request_waiting' where employee_id=:employeeid;",
        {replacements: {employeeid: employeeid},model:employees,mapToModel: true,type:sequelize.QueryTypes.UPDATE}
        ).then(function(){
            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);
    }
```

3. API call for printing WFM-manager table.



```
route.post("/wfmtable",async function(request,response){
    try{
        let loginname = request.body.username;
        await sequelize.query("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'",
        {replacements: {managername: loginname},model:Softlock,mapToModel: true,type:sequelize.QueryTypes.SELECT}
        ).then(function(employees){
            response.status(200).json(employees)
        })
    }
    catch(e){
        console.log(e)
        response.status(500);
    }
})
```

4. API call for after WFM-manager accepts or rejects value will be updated to the DB.

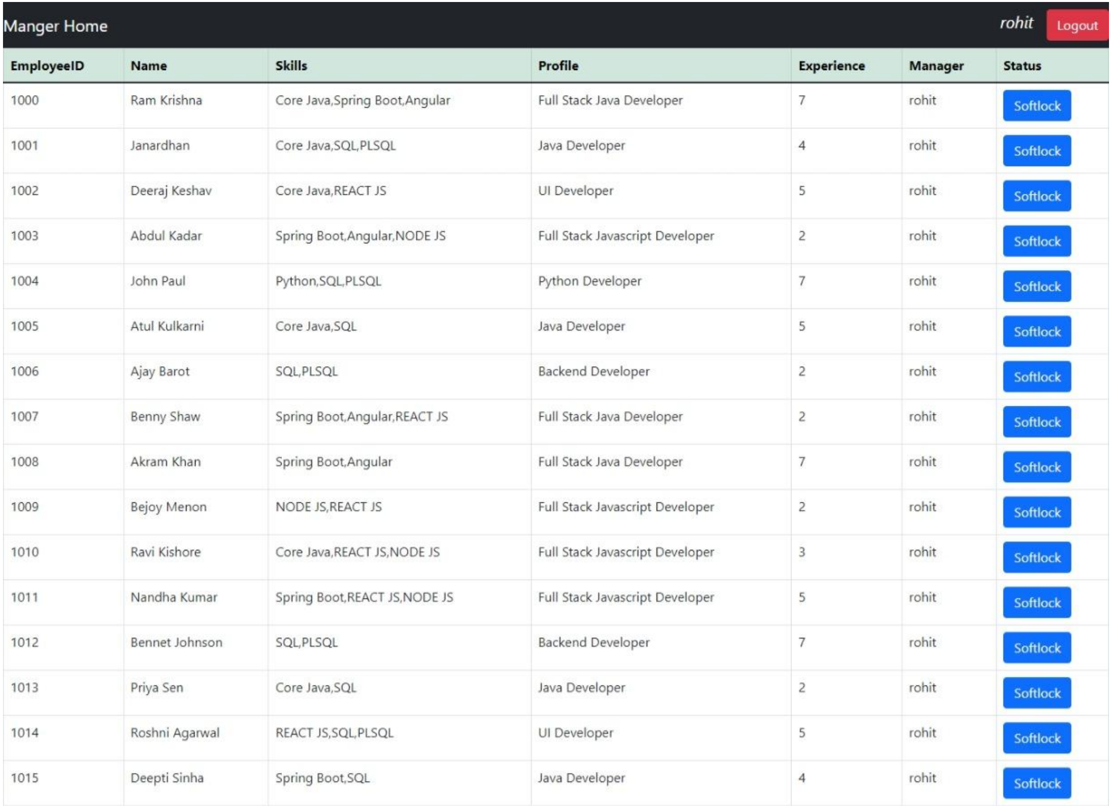


```
route.put("/updateEmployee",async function(request,response){
    try{
        let employeeid = request.body.employeeid;
        let status = request.body.status;
        await sequelize.query("update employee set lockstatus = :status where employee_id = :employeeid",
        {replacements: {employeeid:employeeid,status:status}, model:employees,mapToModel: true,type:sequelize.QueryTypes.UPDATE}
        ).then(function(){
            response.status(200).json()
        })
    }
    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()
        })
    }
    catch(e){
        console.log(e)
        response.status(500);
    }
})
```

USER INTERFACE

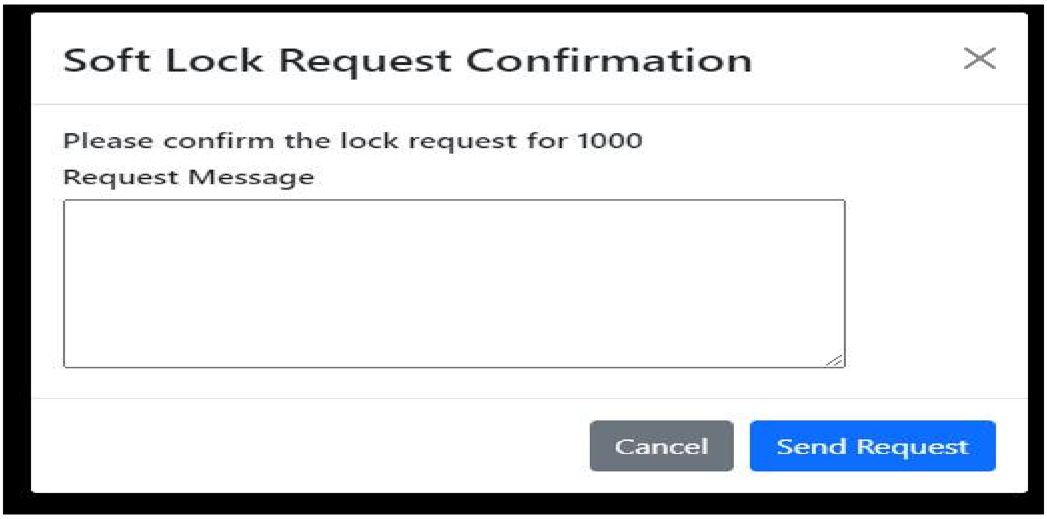
1. Initially the code has the functionality to detect the logged in person whether he is a manager or WFM-manager.

2. If a manager logs in then he will be able to see the employees with their respective status, skill, experience provided with soft lock button which will give a modal popup.



EmployeeID	Name	Skills	Profile	Experience	Manager	Status
1000	Ram Krishna	Core Java,Spring Boot,Angular	Full Stack Java Developer	7	rohit	Softlock
1001	Janardhan	Core Java,SQL,PLSQL	Java Developer	4	rohit	Softlock
1002	Deeraj Keshav	Core Java,REACT JS	UI Developer	5	rohit	Softlock
1003	Abdul Kadar	Spring Boot,Angular,NODE JS	Full Stack Javascript Developer	2	rohit	Softlock
1004	John Paul	Python,SQL,PLSQL	Python Developer	7	rohit	Softlock
1005	Atul Kulkarni	Core Java,SQL	Java Developer	5	rohit	Softlock
1006	Ajay Barot	SQL,PLSQL	Backend Developer	2	rohit	Softlock
1007	Benny Shaw	Spring Boot,Angular,REACT JS	Full Stack Java Developer	2	rohit	Softlock
1008	Akram Khan	Spring Boot,Angular	Full Stack Java Developer	7	rohit	Softlock
1009	Bejoy Menon	NODE JS,REACT JS	Full Stack Javascript Developer	2	rohit	Softlock
1010	Ravi Kishore	Core Java,REACT JS,NODE JS	Full Stack Javascript Developer	3	rohit	Softlock
1011	Nandha Kumar	Spring Boot,REACT JS,NODE JS	Full Stack Javascript Developer	5	rohit	Softlock
1012	Bennet Johnson	SQL,PLSQL	Backend Developer	7	rohit	Softlock
1013	Priya Sen	Core Java,SQL	Java Developer	2	rohit	Softlock
1014	Roshni Agarwal	REACT JS,SQL,PLSQL	UI Developer	5	rohit	Softlock
1015	Deepti Sinha	Spring Boot,SQL	Java Developer	4	rohit	Softlock

3. After clicking on the soft lock button on a particular employee ,it will show a popup screen.

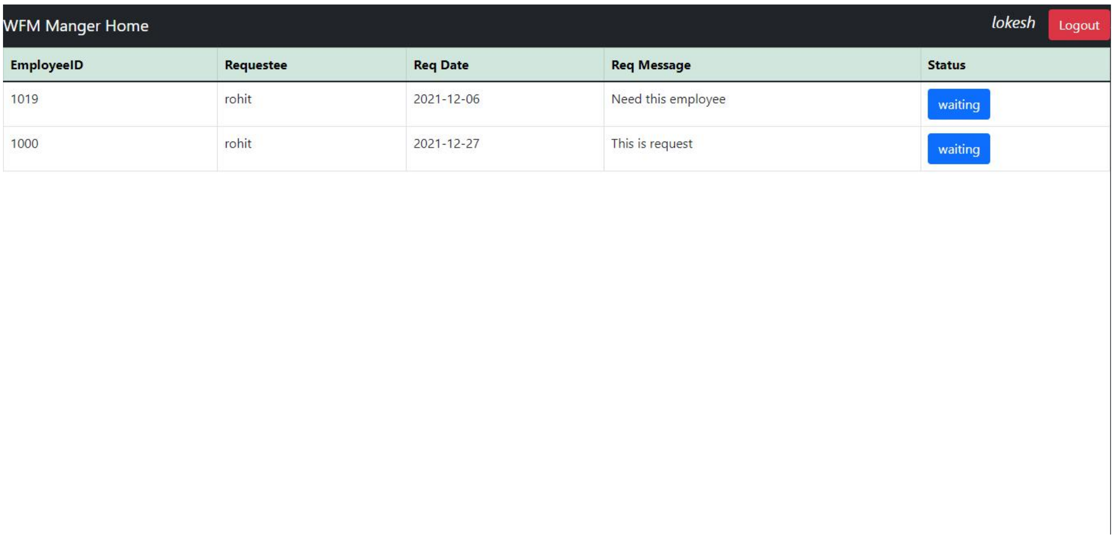


Soft Lock Request Confirmation

Please confirm the lock request for 1000

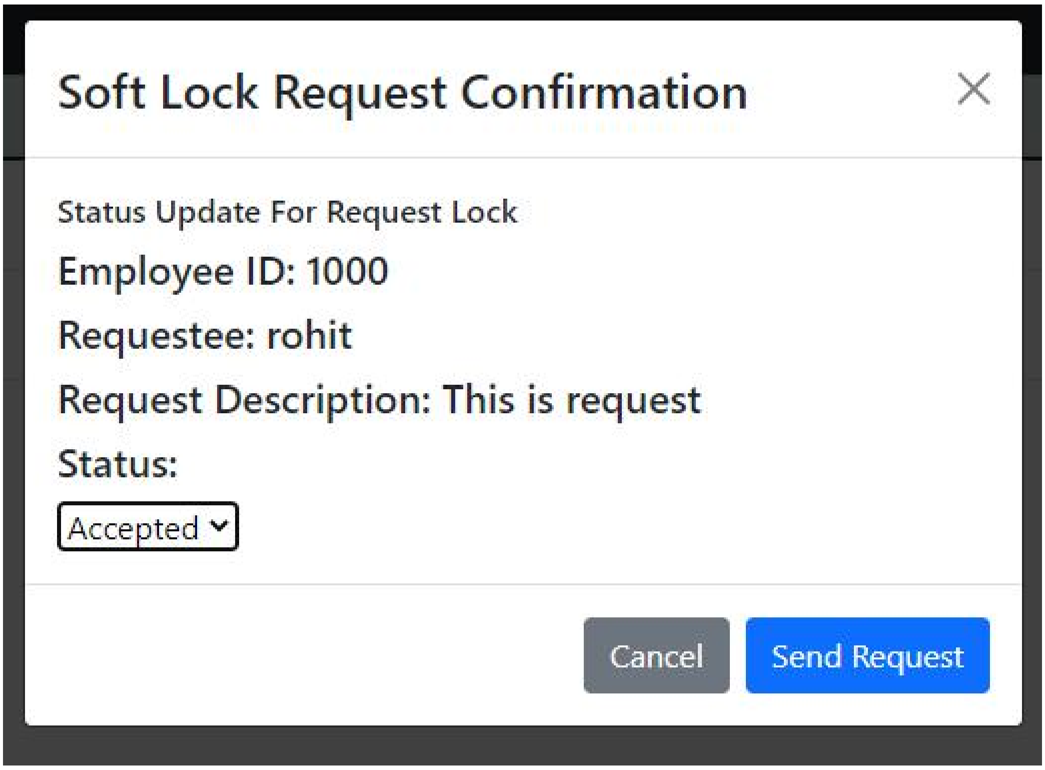
Request Message

4. If WFM-Manager logs in we can observe the request sent by manager as a table in addition with a button



EmployeeID	Requestee	Req Date	Req Message	Status
1019	rohit	2021-12-06	Need this employee	waiting
1000	rohit	2021-12-27	This is request	waiting

5. If WFM-Manager clicks on the button for a particular request it will raise a popup for confirmation.


Soft Lock Request Confirmation

Status Update For Request Lock

Employee ID: 1000

Requestee: rohit

Request Description: This is request

Status:

Accepted

6. After clicking Send Request Db got updated on both employee and soft lock table.

THANK YOU