React and Node JS Assessment

Requirements:

1. Installation of MySQL.

2. Configuring/Adding database: wfm Database(backup.sql).

3. Checking the login functionality for WFM-Manager and Manager.

Manager Login:

4. Printing employee data along with skills in Manager login (Employees and Skills table).

5. A button for softlock need to be added for every list item.

6. A Popup for after clicking the button with entry field to send request**.**

WFM-Manager Login:

7. Printing Softlock data along with previous manager from employee table (Softlock and Employees Table).

8. A button for accepting confirmation which leads to a Modal Popup.

9. Updating values in the softlock if the request is accepted by wfm manager.

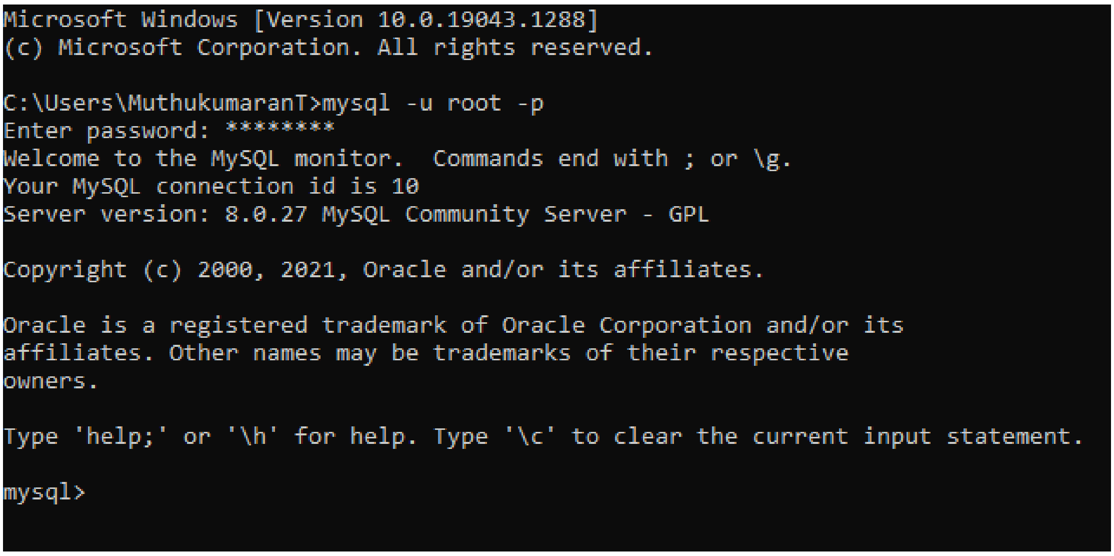
Implementation

As a first step we need to install MySQL into our local machine

Configuring database into the local machine:

Step1: -

After installation go to command prompt and type the following command **“MySQL -u root -p”** which will ask to enter the password**.**



```
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\MuthukumaranT>mysql -u root -p
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.27 MySQL Community Server - GPL

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Step2: -

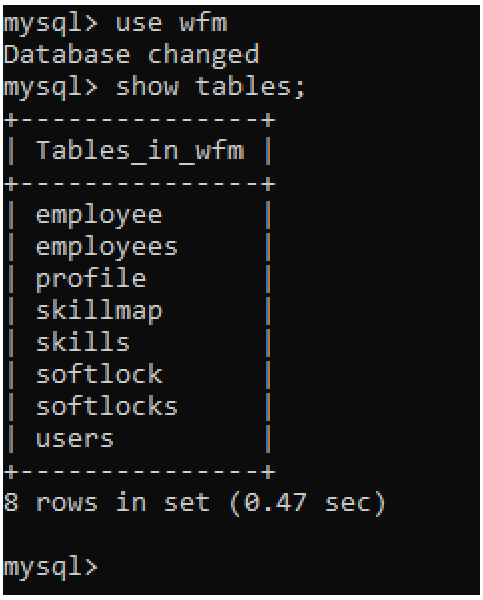
After entering password for adding database into the local machine use this command "**Create database wfm** ". It will create a new database.

Step3: -

After creating new database use this command "**MySQL -u root -p –database wfm < backup.sql"**.

Step4: -

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 (0.47 sec)

mysql>
```

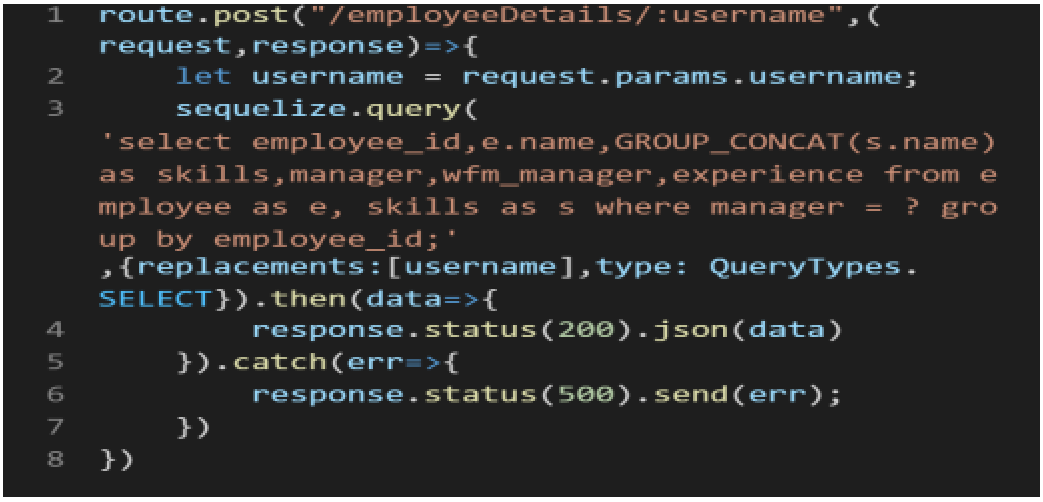
In order to print data in the React app, we need to have API calls for printing employee data in their respective logins.

API CALLS

As a first step I have tried to do the api calls using sequelize calls. Because of facing some issues while running the sequelize queries I have switched to write api calls based on raw query.

API call for printing employees

1. Api call for printing the employees from the employees table and skills of the particular employee.



```
route.post("/employeeDetails/:username",(request,response)=>{
    let username = request.params.username;
    sequelize.query('select employee_id,e.name,GROUP_CONCAT(s.name) as skills,manager,wfm_manager,experience from employee as e, skills as s where manager = ? group by employee_id;',{replacements:[username],type: QueryTypes.SELECT}).then(data=>{
        response.status(200).json(data)
    }).catch(err=>{
        response.status(500).send(err);
    })
})
```

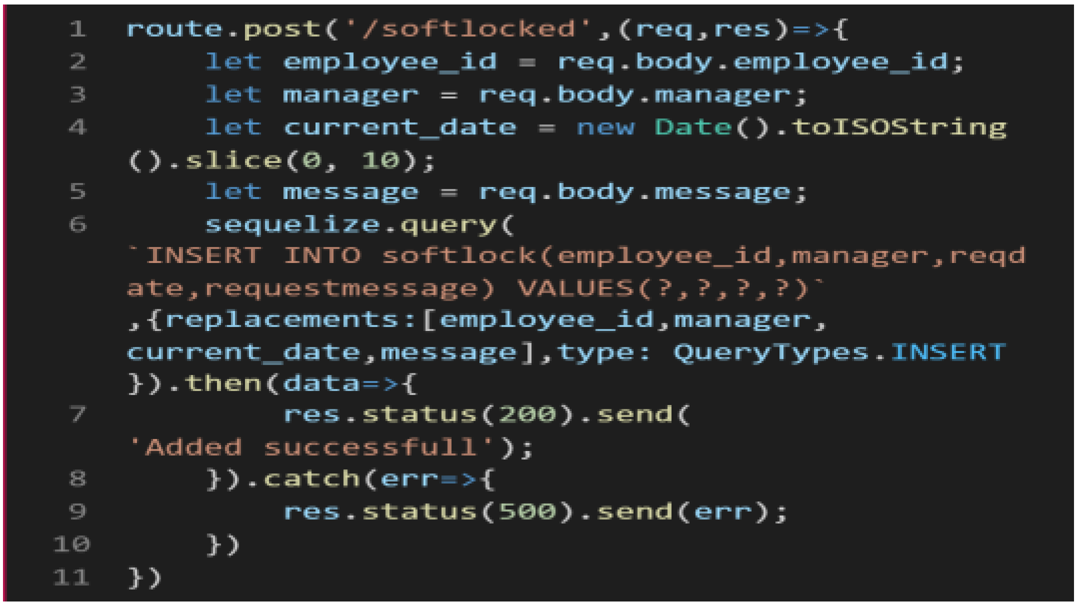
2. This api call is a post method because in sql query the employees list has to be printed based on the Manager login.

3. So, using a local variable assigned with username of logged in manager will be passed into the sql query.

4. Based on that list of employees will be printed.

API call for passing request to WFM-Manager

1. Api call for sending request to the wfm manager by updating values into softlock table.



```
route.post('/softlocked',(req,res)=>{
    let employee_id = req.body.employee_id;
    let manager = req.body.manager;
    let current_date = new Date().toISOString
().slice(0, 10);
    let message = req.body.message;
    sequelize.query(
`INSERT INTO softlock(employee_id,manager,reqd
ate,requestmessage) VALUES(?,?,?,?)`
,{replacements:[employee_id,manager,
current_date,message],type: QueryTypes.INSERT
}).then(data=>{
        res.status(200).send(
'Added successfull');
    }).catch(err=>{
        res.status(500).send(err);
    })
})
```

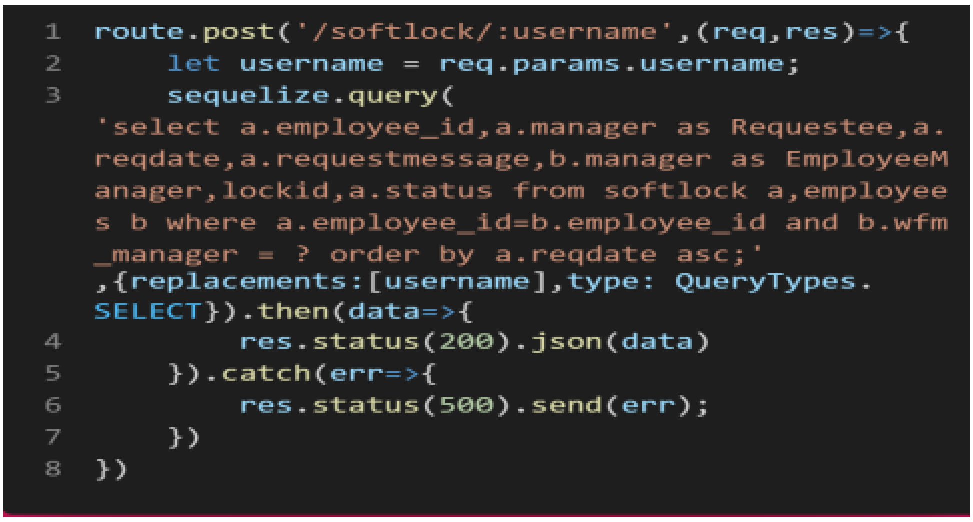
2. This API call is post method because we are inserting values into the table softlock as sending request to WFM-Manager.

3. Hence, using local variables for inserting values into the table from the manager home.

4. This will create a record in softlock as request sent by the manager to the WFM-Manager (which will be retrieved in the WFM-Manager side).

API call for Printing Request to WFM-Manager

1. API call for printing list of requests from Softlock and employees table.



```
route.post('/softlock/:username',(req,res)=>{
    let username = req.params.username;
    sequelize.query(
'select a.employee_id,a.manager as Requestee,a.
reqdate,a.requestmessage,b.manager as EmployeeM
anager,lockid,a.status from softlock a,employee
s b where a.employee_id=b.employee_id and b.wfm
_manager = ? order by a.reqdate asc;'
,{replacements:[username],type: QueryTypes.
SELECT}).then(data=>{
        res.status(200).json(data)
    }).catch(err=>{
        res.status(500).send(err);
    })
})
```

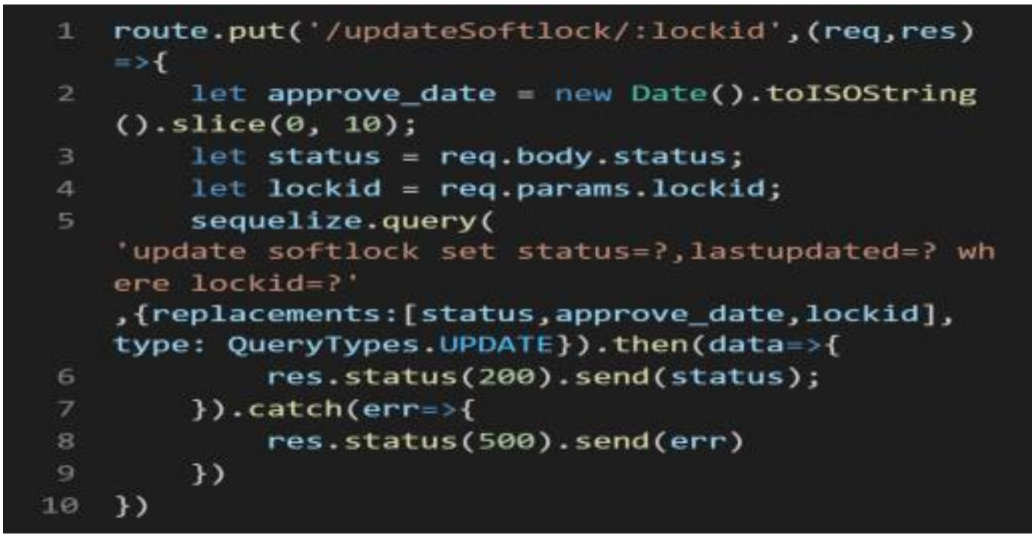
2. This API call is post method because in SQL query the softlock list has to be printed based on the Employee to Manager Relation and Manager to WFM-Manager Relation.

3. Here the username of the logged in manager will be passed as the parameter into the SQL query for populating the request.

4. As a result, requests for particular manager will be printed in the WFM-Manager home.

API call for updating the status from WFM-Manager

1. Api call for sending status(Accept/Reject) back to the manager by updating values into softlock table.



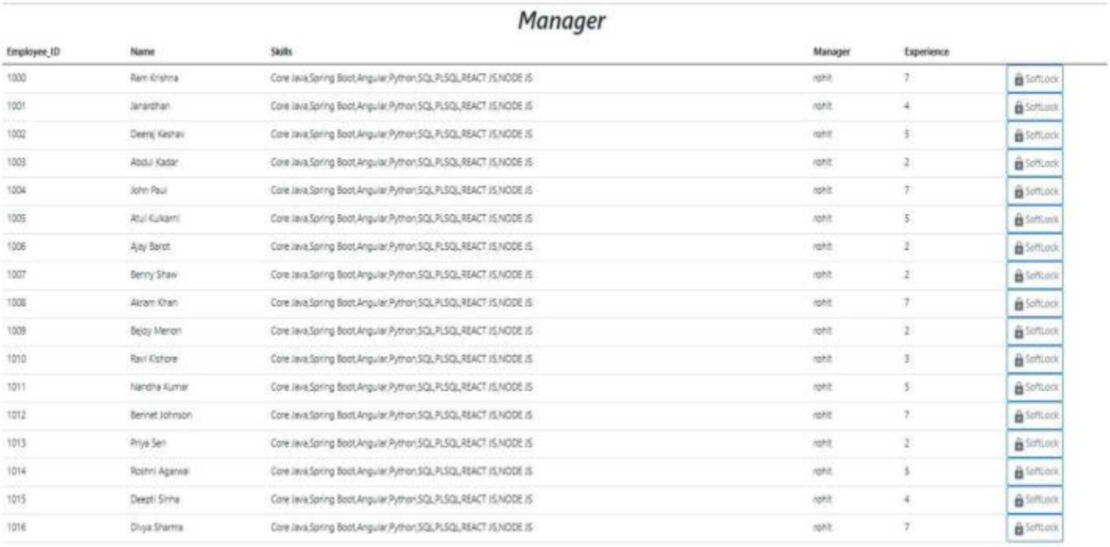
```
route.put('/updateSoftlock/:lockid',(req,res)
=>{
    let approve_date = new Date().toISOString
().slice(0, 10);
    let status = req.body.status;
    let lockid = req.params.lockid;
    sequelize.query(
'update softlock set status=?,lastupdated=? wh
ere lockid=?'
,{replacements:[status,approve_date,lockid],
type: QueryTypes.UPDATE}).then(data=>{
        res.status(200).send(status);
    }).catch(err=>{
        res.status(500).send(err)
    })
})
```

2. This is a Put method as we are updating the record of values in the softlock table.
3. Here, using local variables for getting data from the wfm-manager to update the record in softlock table.
4. So, this call will update the values of selected record in which values will be taken from the wfm-manager home.

USER INTERFACE

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

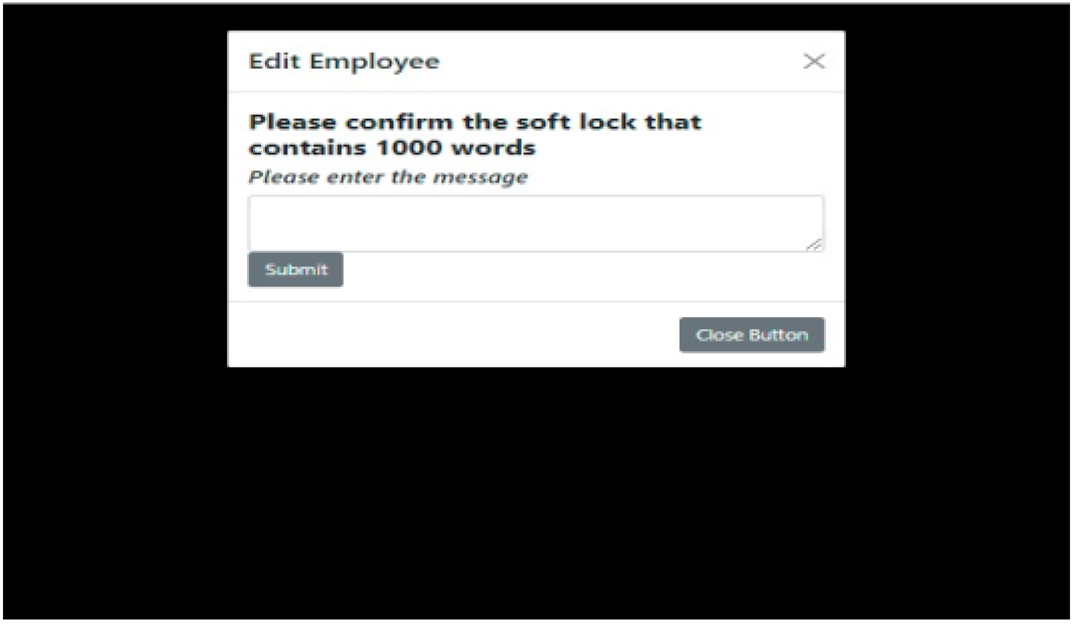
2. If a manager logs in then few will be able to see the employees with their respective status, skill, experience provided with softlock button which will give a modal popup.



Manager

Employee_ID	Name	Skills	Manager	Experience	
1000	Ram Krishna	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	7	SoftLock
1001	Janardhan	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	4	SoftLock
1002	Deeraj Keshav	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	5	SoftLock
1003	Abdul Kadar	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	2	SoftLock
1004	John Paul	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	7	SoftLock
1005	Atul Kulkarni	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	5	SoftLock
1006	Ajay Barot	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	2	SoftLock
1007	Benny Shaw	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	2	SoftLock
1008	Akram Khan	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	7	SoftLock
1009	Bejoy Menon	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	2	SoftLock
1010	Ravi Kishore	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	3	SoftLock
1011	Nandha Kumar	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	5	SoftLock
1012	Bennet Johnson	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	7	SoftLock
1013	Priya Sen	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	2	SoftLock
1014	Roshni Agarwal	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	5	SoftLock
1015	Deepti Sinha	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	4	SoftLock
1016	Divya Sharma	Core Java,Spring Boot,Angular,Python,SQL,PLSQL,REACT JS,NODE JS	rohit	7	SoftLock

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

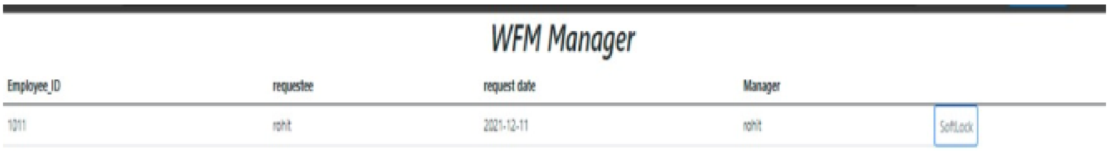


Edit Employee

Please confirm the soft lock that contains 1000 words

Please enter the message

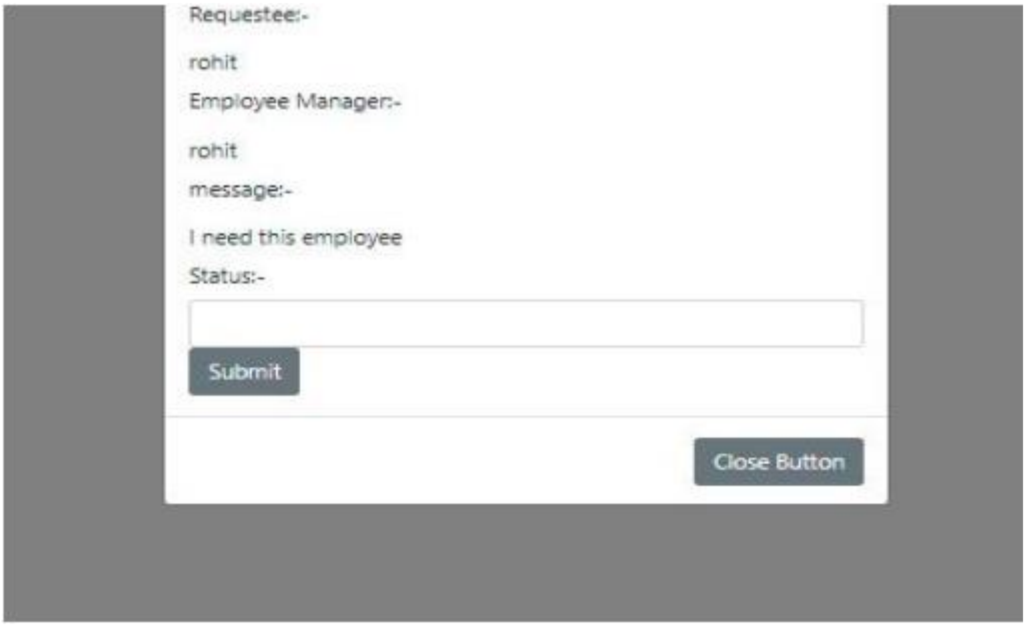
4. If WFM-Manager logs in we can observe the request sent by manager as a table in addition with a button.



WFM Manager

Employee_ID	requestee	request date	Manager	
1011	rohit	2021-12-11	rohit	SoftLock

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



Requestee:-

rohit

Employee Manager:-

rohit

message:-

I need this employee

Status:-

6. Data base will be updated using the API call.