**Activities/Solution/Approach**

**Node JS**

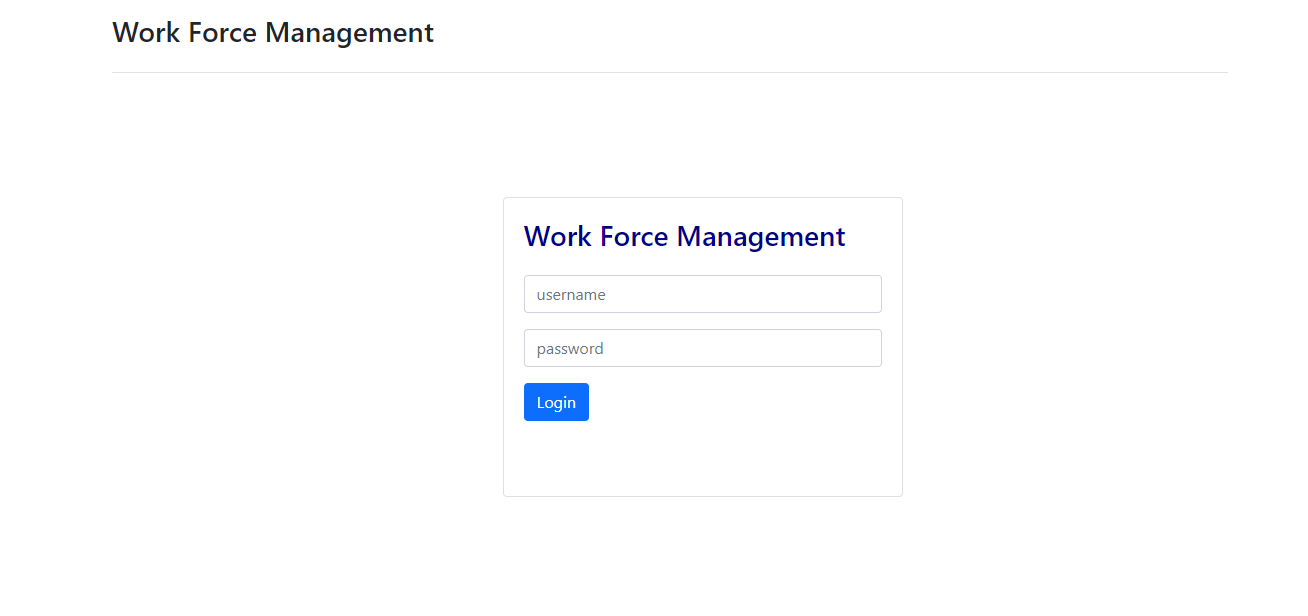
1. Creating Models for Employee, Skill, SkillMap, Softlock tables.
2. Creating the relationship between the models.
3. Creating API for getting the list of employees those are mapped with the ‘Manager’ parameter.
   1. Get ‘manager’ as parameter and use findAll method where condition.
   2. Return result as json.
4. Creating API for updating employee and create new record in the softlock table.
   1. Get ‘employee id’ as parameter and request comment as request body.
   2. Updating the lockstatus of the employee with ‘request\_waiting’.
   3. Create new record with appropriate data in the softlock table for that employee.
   4. On success of updating return employee list those associated with logged In employee.
5. Creating API for getting the list softlocks which is in ‘waiting’ status and those are mapped with the ‘Wfm\_Manager’ parameter.
   1. Get ‘manager’ as parameter and use findAll method where condition.
   2. Return result as json.
6. Creating API for updating softlock and employee based on the parameter.
   1. Get ‘lockid’ as parameter and managerstatus request body.
   2. Updating the lockstatus of the employee based on the user input.
   3. Update softlock based on the user input.
   4. On success of updating return employee list those associated with logged In employee.
7. Creating passport authorization and applied in the all the API.

**React JS**

1. Creating Logout component to implement the Logout functionality.
2. Creating LogoutHOC to connect the state - event and Logout component.
3. Creating ManagerGrid component to display the employees in the tabular columns.
4. Creating ManagerGridHOC to connect the state - event and ManagerGrid component.
5. Creating ManagerModel component to display the modal dialog.
6. Creating ManagerModelHOC to connect the state - event and ManagerModel component.
7. Creating appropriate reducers to update the state of the ManagerGrid and ManagerModel component.
8. Creating appropriate handler generator function to get and update employees.
9. Creating WfmGrid component to display the employees in the tabular columns.
10. Creating WfmGridHOC to connect the state - event and WfmGrid component.
11. Creating WfmModel component to display the modal dialog.
12. Creating WfmModelHOC to connect the state - event and WfmModel component.
13. Creating appropriate reducers to update the state of the WfmGrid and WfmModel component.
14. Creating appropriate handler generator function to get and update softlocks.
15. Creating some reusable component to construct the grid, skill and modal.

**OUTPUT**

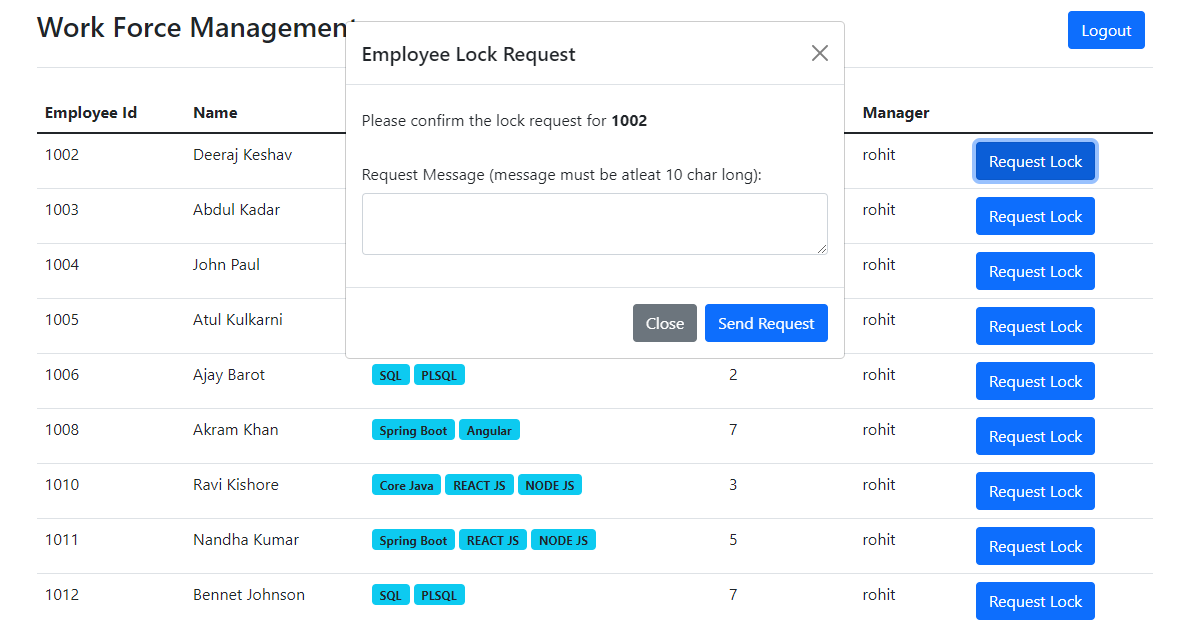
**Login Screen:**



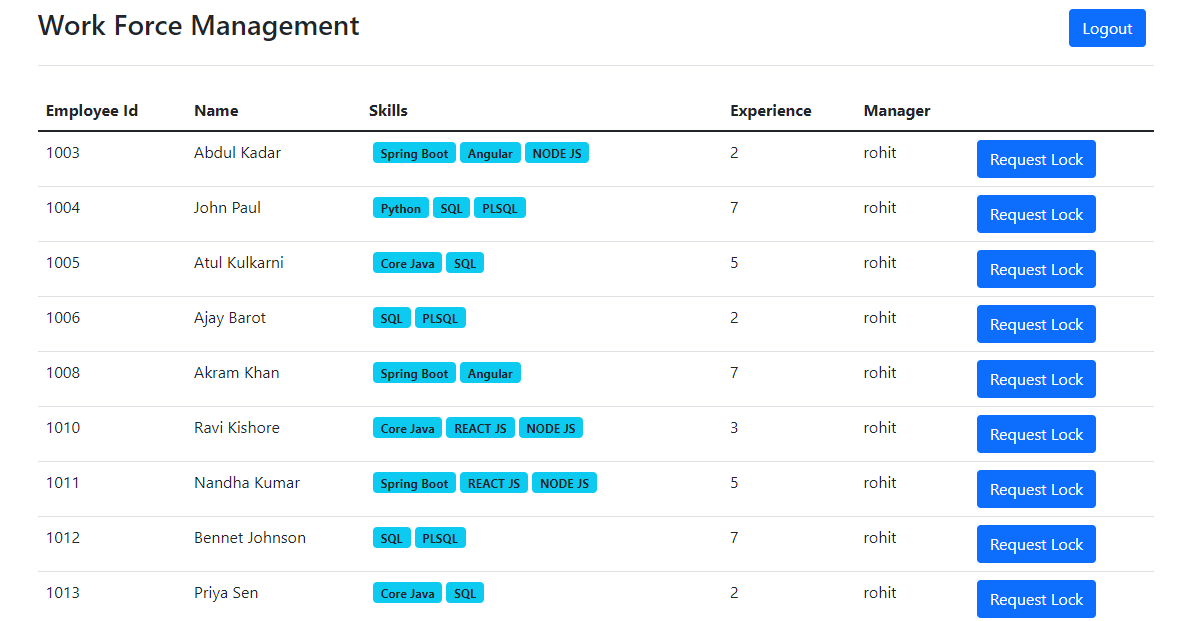
**Login as Manger user (Rohit)**



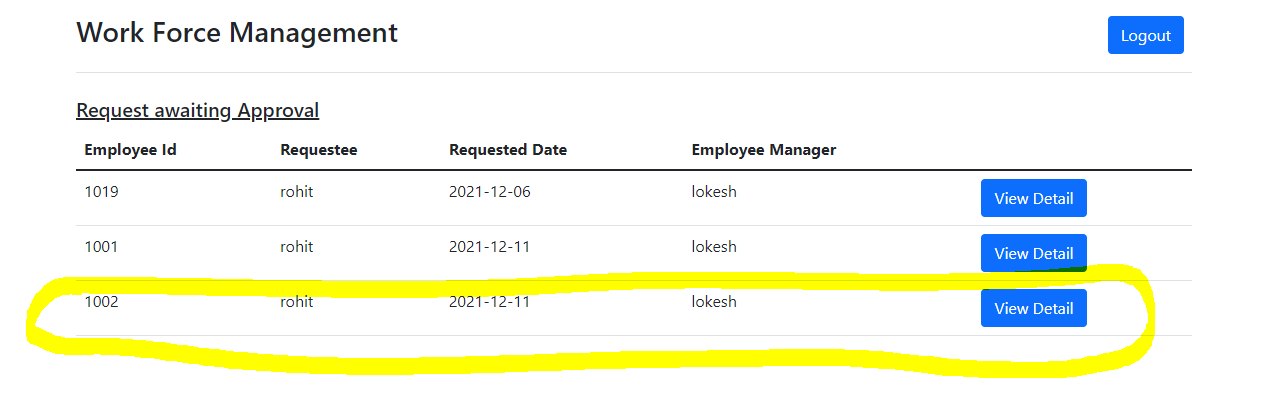
**Requesting the lock for employee ‘1002’**



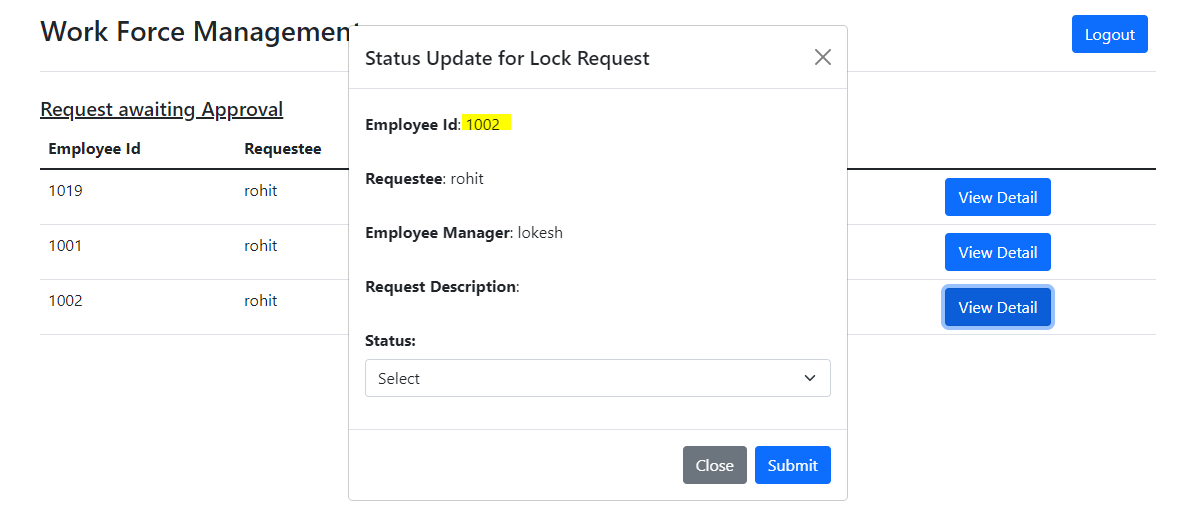
**After request lock for ‘1002’**



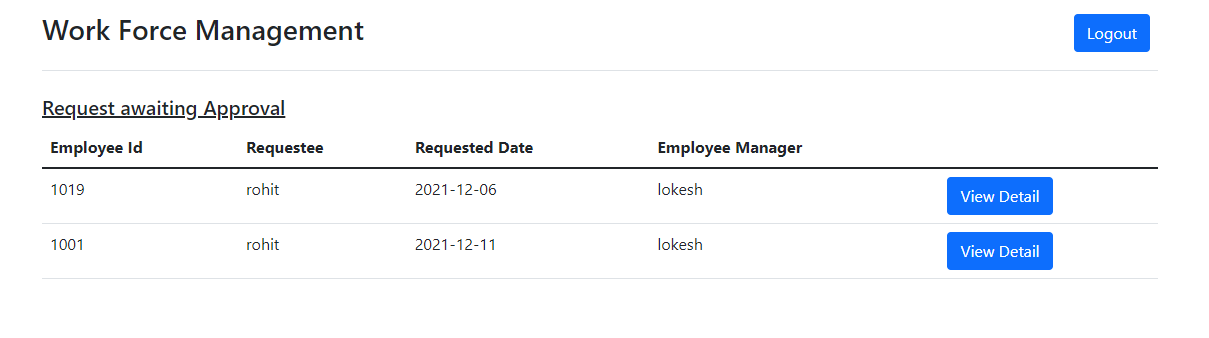
**Logout & Login as Wfm Manager (Lokesh)**



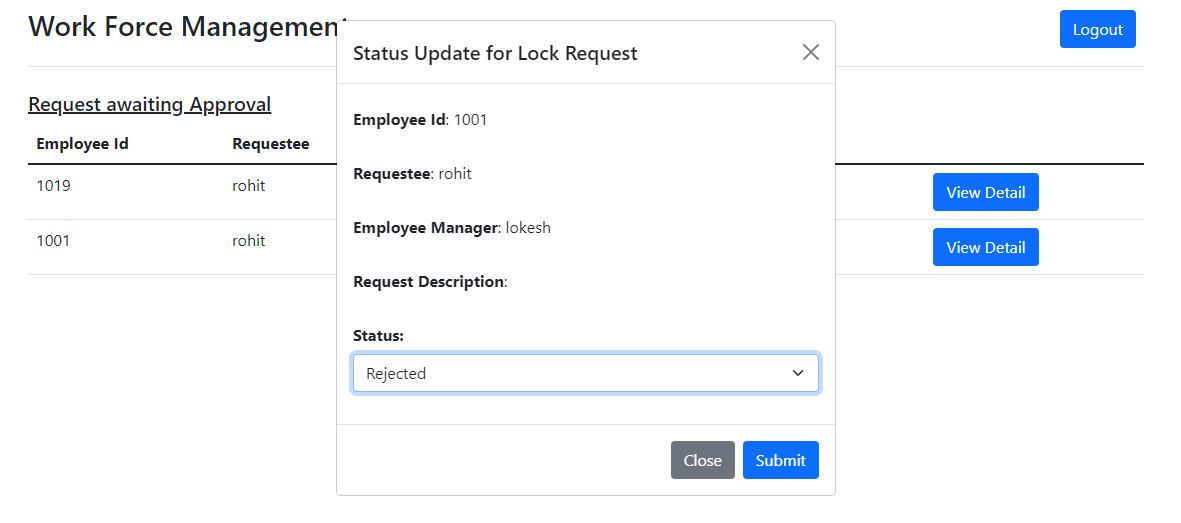
**Viewing the Request Details of employee ‘1002’**



**After Accepting the Request for employee ‘1002’**

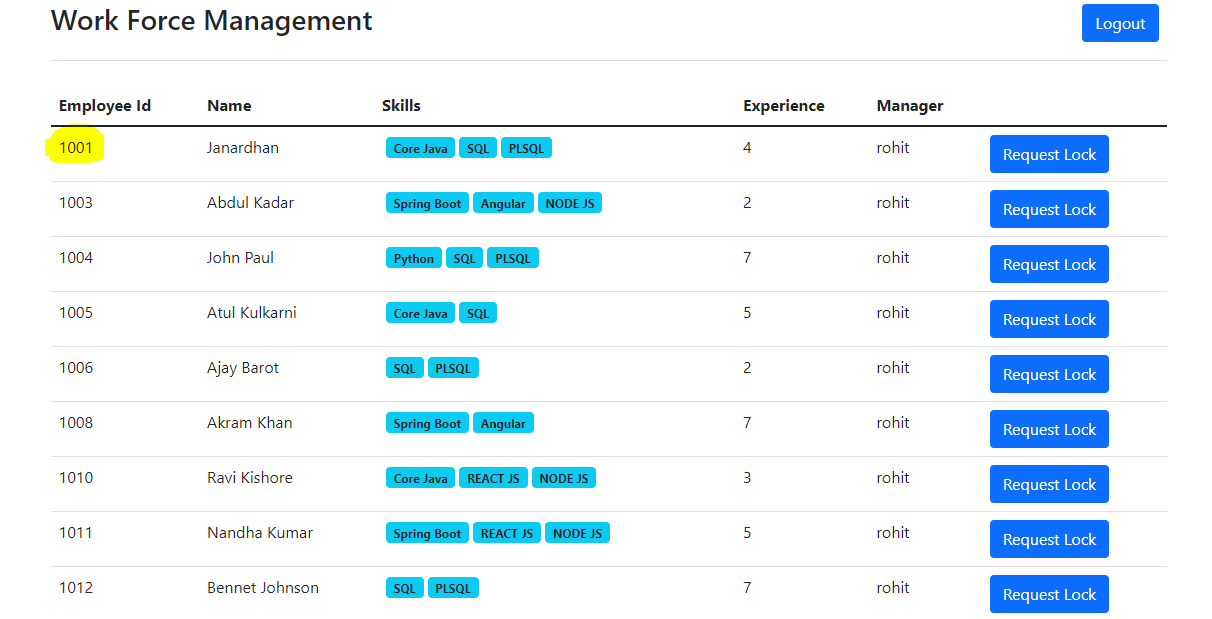


**Rejecting the request for the employee ‘1001’**



**After the request for the employee ‘1001’**

**Manager Screen (Rohit)**



**Wfm Manager screen (Lokesh)**

