For the employees page:

leaveRequestsList - FInd many on the employees collection to get all requested leaves

Filter : {"leaves.status": "requested"}

Approve/reject leave request: Update one on the employees collection

Filter: {"first\_name":{(components.text4.value}},"leaves.status":"requested"}

Update: { $set: {"leaves":{"status":"approved"/”rejected”}}}

addEmployee: To add employee from the form in our front end

Operation : insert one

Document : { first\_name: "{{components.textinput1.value}}", last\_name: "{{components.textinput2.value}}", email: "{{components.textinput4.value}}", phone: "{{components.textinput3.value}}" }

allEmployees: To get all employees

Operation: FInd many

Collection: employees

**For the dashboard**

**upcomingPaidLeaves :** To show upcoming approved paid leaves

Operation : aggregate

Query : [{$match: {"leaves": { "$elemMatch" : { "status":"approved" }}}},{"$project": {"first\_name":1, "last\_name":1, "email":1, "leaves": {"$filter": {"input": "$leaves", "as": "leaves", "cond": { $and: [{"$gt":["$$leaves.start\_date","{{moment().toISOString(true)}}"]},{ "$eq":["$$leaves.status","approved"]}] } }}}}]

To format the data we get from the query we used this following js script:

approvedLeavesArray = queries.upcomingPaidLeaves.data;

var arrayObjects = approvedLeavesArray.reduce((acc, cv) => {

return acc.concat(cv.leaves.reduce((accLeave, leave) => {

accLeave.push({

"first\_name": cv["first\_name"],

"email": cv["email"],

"date": moment(leave["start\_date"]).format("DD-MM-YYYY"),

"leave\_days": leave["leave\_days"],

});

return accLeave;

}, []));

}, []);

return arrayObjects;

Chart: To plot the bar graph for the leaves data

Operation : Find many

Query: { "leaves.leave\_days": { $gt: 0 }} //to get leave days greater than 0

monthChart: to format the data we got from the chart query

Js:

currentMonth = moment().format("M");

let leaves = queries.chart.data;

let months = leaves.reduce((acc, cv) => {

cv.leaves.forEach(obj => {

if (moment(obj.start\_date).format("M") <= currentMonth && (moment(obj.start\_date).format("M") >= (currentMonth - 3))) {

if (acc.hasOwnProperty(moment(obj.start\_date).format("MMMM"))) {

acc[moment(obj.start\_date).format("MMMM")] = acc[moment(obj.start\_date).format("MMMM")] + obj.leave\_days

} else {

acc[moment(obj.start\_date).format("MMMM")] = obj.leave\_days

}

}

})

return acc;

}, {});

return months;

leaveDaysTotal : to get total leave days

Find many on the employees collection

Then we format it using convertLeaveDaysTotal script :

var leaves = queries.leaveDaystotal.data;

var count = leaves.reduce((acc,cv)=> {cv.leaves.forEach(obj => { acc += obj.leave\_days})

return acc;},0)

let result = {value : count}

return result;

pendingApproval : we use find many to find the employee data with filter

{"leaves.status": "requested"}

**For the calender page**

companyHolidays: FInd many on the collection company\_holidays to get all holidays

addHoliday : insert one on the company holidays collection

Document: {title: "{{components.textinput1.value}}", start: "{{moment(components.datepicker1.value).toISOString(true)}}" }

convertDateFormat : to convert the data collection from the companyHolidays query

let holidaysData = queries.companyHolidays.data;

let formattedData = holidaysData.reduce((acc, cv) => {

acc.push({

...cv,

start: moment(cv.start).format("MM-DD-YYYY HH:mm:ss A Z"),

end: moment(cv.start).format("MM-DD-YYYY HH:mm:ss A Z")

})

return acc;

}, [])

return formattedData;

fetchApprovedLeaves : To fetch the approved leaves we use aggregate operation

[{$match: {"leaves": { "$elemMatch" : { "status":"approved" }}}},{"$project": { "first\_name": 1, "email": 1, "leaves": {"$filter": {"input": "$leaves", "as": "leaves", "cond": {"$eq":["$$leaves.status","approved"]}}}}}]

formattedApprovedLeaves :

var approvedLeavesArray = queries.fetchApprovedLeaves.data;

var arrayObjects = approvedLeavesArray.reduce((acc, cv) => {

return acc.concat(cv.leaves.reduce((accLeave, leave) => {

accLeave.push({

"title": cv["first\_name"],

"email": cv["email"],

"start": moment(leave["start\_date"]).format("MM-DD-YYYY HH:mm:ss A Z"),

"end": moment(leave["start\_date"]).add(leave["leave\_days"], "days").format("MM-DD-YYYY HH:mm:ss A Z"),

});

return accLeave;

}, []));

}, []);

return arrayObjects;

mergeData : to toggle between show leaves or not

return components.toggleswitch2.value ? queries.convertDateFormat.data.concat(queries.formattedApprovedLeaves.data) : queries.convertDateFormat.data;

For the request page:

approvedCount : to count the number of approved leaves

**Filter : {"email": "{{globals.currentUser.email}}", "leaves.status":"approved"}**

leaveRequest : to request a leave

Operation : update one

Filter: {"email": "{{globals.currentUser.email}}"}

Updation : {$push:{"leaves":{"start\_date": "{{moment(components.datepicker1.value).toISOString(true)}}", "leave\_days":{{components.numberinput1.value}}, "status": "requested"}}}

listRequests : find many

{email: "{{globals.currentUser.email}}"} with the filter