Assignment 12 || Employee Management

assignment12_category_0005

A well-renowned company wants to monitor the workload of the employees and keep records of salary, contracts etc. They need a website where the employees can post the updates of their workflow, the HR (Human Resources) Executive needs to monitor the employee workflow, be able to pay them etc. Please, read the requirements thoroughly.

🟢 **Important**

Read the whole instruction at once. Then take a deep breath. Sort out which tasks should be done first. Make your own notes on how to approach a solution or on which section you may need help. Try to split a task into multiple smaller tasks, then solve them individually.

Key Rules:

- Include a minimum of 20 notable GitHub commits on the client side
- Include a minimum of 12 notable GitHub commits on the server side
- Add a meaningful readme.md file with the name of your website, Admin username, password and live site URL. Include a minimum of 10 bullet points to feature your website.
- Make it responsive for all devices. You need to make it responsive for mobile, tablet and desktop views. Make the dashboard responsive as well.
- After reloading the page of a private route, the user should not be redirected to the login page.
- Use the Environment variable to hide the Firebase config keys and Mongodb credentials.
- Don’t use any Lorem ipsum text in your website.
- Show sweet alert/toast/notification for all the crud operations, successful authentication login, and sign-up. Don’t use the default browser alert.
- Implement tanstack query in all the data fetching functionality (For GET method only)
- Provide the Admin User’s Email/Password in your README.md file.

Main Requirements

- Make sure your design and website idea is unique. First, finalize your idea (what type of website you want to build). Then google the site design or visit theme forest to get your website idea. [You can visit this blog to collect free resources for your website]. However, your website can not be related to your previous assignments or any demo project displayed in the course or in our conceptual sessions

🏡Home Page

In this web page or I should say, web app, the Home page will contain a

- Banner / Slider (Displaying the success of the company || relevant idea you can come up with)
- Services (Services your company provide)
- Testimonials (Could be a slider showing off appreciation from different people about the company)
- Two More relevant sections you can think of.

In your layout, please include **Navbar** and **Footer.** So that these could be seen anywhere throughout the web app.

🗺️ Navbar

Navbar will have a logo, that represents the company. Conditional Register, Login Button (if the visitor is not logged in). Conditional User Photo, and clicking that photo reveals a functioning Logout button. (if a user is logged in)

The navbar will have these nav items.

- Dashboard (Private)
- Contact us (Public)

👥 Users and Authentication

You Must implement Email and Password-based Authentication. This means you will have to implement the Registration and the login page. Users should be able to toggle between Login and Registration view.

> ❗ **Note**: Do not enforce the forget or reset password feature and the email verification method, as it will inconvenience the examiner. If you want, you can add email verification/forget the password after receiving the assignment result.

On the Registration page, display errors when:

The password

- is less than 6 characters
- don't have a capital letter
- don't have a special character

On the Login page, display errors when:

- The password doesn't match
- email doesn't match

You can take the error message from Firebase. You can show the error below the input fields or via alert/toast. If you use alert/toast, do not implement the browser alert.

As you've guessed, the possible roles of the users of this website will be…

- Employee
- HR
- Admin

In your registration page, you need to add a required dropdown menu which allows you to specify the role of the user while registering via email/password authentication. **Don't allow the 'Admin' menu in the dropdown. Since it should be created from the backend or database.**

This must be a required field. Other necessary fields must be present in your registration form. You need to do this in the Email/Password authentication.

Make sure you create a collection named user or people in your database to keep track of the role of the users. Also, you may need other fields such as

- bank_account_no
- salary
- Designation [For example: Sales Assistant, Social Media executive, Digital Marketer etc.]
- photo

Use image upload for User's photo (imgbb) (as shown in modules) [URL is not allowed]

Add at least one additional social login such as - Google, Github, etc. Creating an account via Social login will be considered as an **Employee** role user.

📉 Routes 📈

📱 Dashboard [Private]

For Employee [Private] [Employee Only]

/work-sheet

There will be a form and a table on this page. Fit this form in a single row horizontally. The form will have the following fields

- Tasks (DropDown menu). The dropdown will have these options
    - Sales
    - Support
    - Content
    - Paper-work
    - (You may add more if you like)
- Hours Worked (numeric)
- Date (You may use React DatePicker component) [Default value, current date]
- Add / Submit button

Clicking the Add / Submit button will add the newest work according to the selected date to the table at the start (without reloading the page) and store the information in the DB as well. Only logged-in Employee-specific data will be shown here.

/payment-history

The monthly history of salary paid by HR will be shown in the table. [HR can pay employee from his /employee-list route] This table will be sorted by default. The earliest month's payment will be in the first row. The rest of the rows will have payment info in the past months.

- Month
- Amount
- Transaction Id

Implement pagination or infinite scrolling if it has more than 5 rows. Only logged-in Employee payment data will be shown here.

For HR (Human Resource Executive) [Private] [HR only]

/employee-list

This page will show all the employee information in a table. Use an appropriate package for creating this table. In this table, each row will contain. You **may** use a suitable table component that you feel comfortable using for a nice view of this page, for example, TanStack Table.

- Name
- Email
- Verified
- Bank Account
- Salary
- Pay
- Details

By default, each employee's verified status will be false. In this verified column, it will show a ❌ (cross) button indicating that the employee has not verified yet. The HR can click this ❌button to toggle it to ✅ , indicate that the employee has been verified. Or vice-versa. Also, don't forget to toggle this status in DB too.

As you can see, in the user or people collection, you may need to add the isVerfied field. For simplicity, let's say if the field doesn't exist, the employee is also considered not verified.

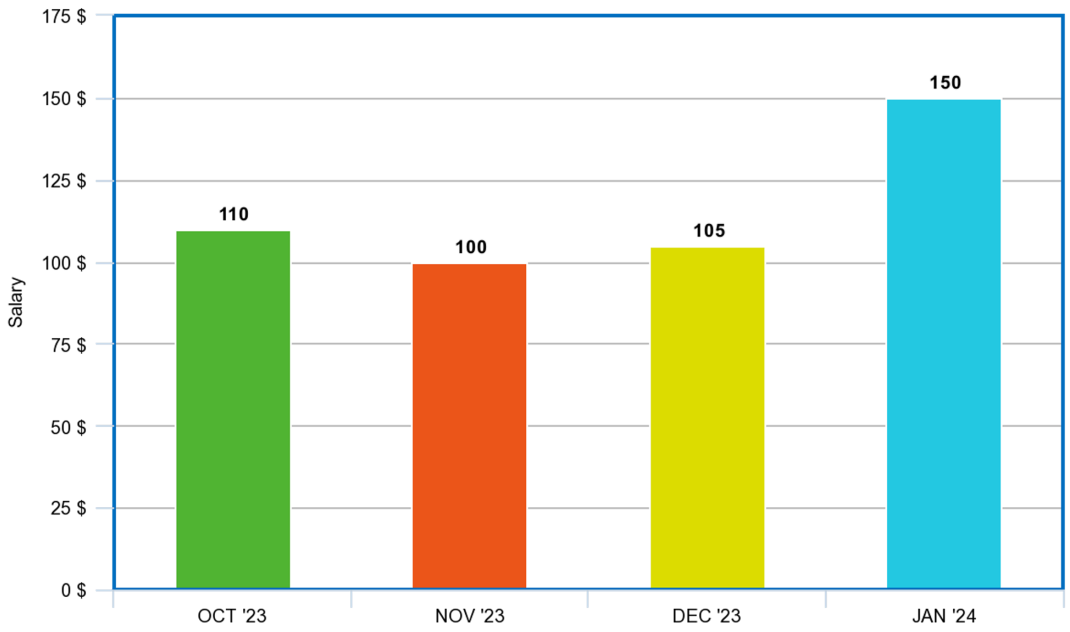
- Clicking the pay button will pop up a modal, In this modal, the salary of the employee in the row will be already written, and two input fields for Month and Year. Clicking the pay button will pay the employee that amount of money. HR can pay Verified employees only. The unverified employees' pay button will be disabled (won't work if

clicked).
[Checkout Challenge section, find 🟨[Yellow] icon].

/details/:slug

[slug could be employee email, uid (from firebase), or any unique identifier that indicates a single employee]

- Clicking the details button on the /employee-list page will take HR to a different page. This page will contain the user's name, photo (you may use photoURL), designation, and a bar chart indicating the **Salary vs. Month** and year plot. Something like below. Use data from your database to create this chart. Static chart is not allowed.



***Note**: You can ignore the labels inside the bar chart. Just make sure the horizontal / x-axis and vertical / y-axis have labels. Each row will have these fields.*

/progress

All the work records posted by employees will appear on this page. Now, HR can filter the table data by selecting the employee's name from a drop-down. Select Month from a drop-down to filter submitted work done in that month. If no employee name is selected, then simply show all the employee's submitted work for selected month.

[See Optional section, find 🟩[GREEN] icon]

For Admin [Private] [Admin only]

/all-employee-list

Admin can see all the verified employees in a table. (including HRs)

- Name
- Designation
- Make HR
- Fire

Yes, you guessed it, Admin can fire an employee or HR. After Firing someone, he won't be able to log in using his account anymore. The Fire column will have a button or icon. Clicking that button will pop up a modal for confirmation. After firing someone, the button or icon will be replaced by the text "**Fired".**

Admin can make an employee HR. After making an employee HR, he can visit the HR-specific routes. And all the stuff he can do. But he won't be able to access his routes as an employee anymore. In the Make HR column, conditionally render a button or icon if the row indicates an employee but not HR.

Admin can adjust the salary of Employees and HRs.

[Checkout Challenge part, find 🟦 [BLUE] icon]

✉ Contact Us

/contact-us

This page will have a dummy address of the company and a nice form, which takes the email, and a message field. So that any visitor can send their opinions. Admin can check these messages

Challenge

HR related

- 🟨[Yellow] Make the payment operation using a Payment Gateway
- 🟨[Yellow] Don't allow payment for the same month/year twice! You don't want to pay an employee twice a month, do you?

Admin related

- 🟦[BLUE] There will be a button which will toggle between Table View and Card Grid View on the same page.
- 🟦[BLUE] Only allow increasing of the salary. Don't allow decreasing.

- Use JWT token / middleware function in the backend for verifying the role for corresponding role-specific mutation operations [update, delete, post]. For example, when an Admin tries to update salary, in the backend, it must check if the user is admin or not via JWT token and a middleware function. Now do these for the rest of the roles. The token must contain the email and role of the users.

Optional Tasks

- Don't use Daisy UI. Instead, use other UI component libraries such as Material Tailwind, ShadCn, Flowbite or Ant Design. Any UI library of your choice, except Daisy UI.
- Use a reusable table component. You may use a table component from the UI library you use.
- Or, You may use TanStack table.
- You may use Headless UI for the drop-down menu
- 🟩[GREEN] Add a section above the table where the summation of work hours is shown. The summation must be changed based on the filtered data. That means, if an employee is selected, then it will sum up the work hours of that employee.

What to Submit

Assignment Category/variant:
Admin email:

Admin password:
Front-end Live Site Link:
Client Side Github Repository Link:
Server Side Github Repository Link: