

Assignment 12 || Employee Management

assignment12_category_0004

This assignment is about managing employees of a company. The salary, work hours, contracts, etc. of an employee can be managed using this website.

● 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.

This assignment won't be evaluated if it shows an error page, and requires 4~5 times refreshing the page to fix it. Ensure your server-side code doesn't have this line `await client.connect()` or pinging the database.

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
- After successfully performing any mutation - Create, Update, Delete operations, please, show a relevant toast/sweet alert. (Don't use the browser alert() function. otherwise, marks will be reduced)
- Provide the Admin User's Email/Password in your **README.md** file. Admin accounts shouldn't be allowed to be created from your website directly.



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 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 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. 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
- photo

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



Dashboard [Private]

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

- 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 is 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.

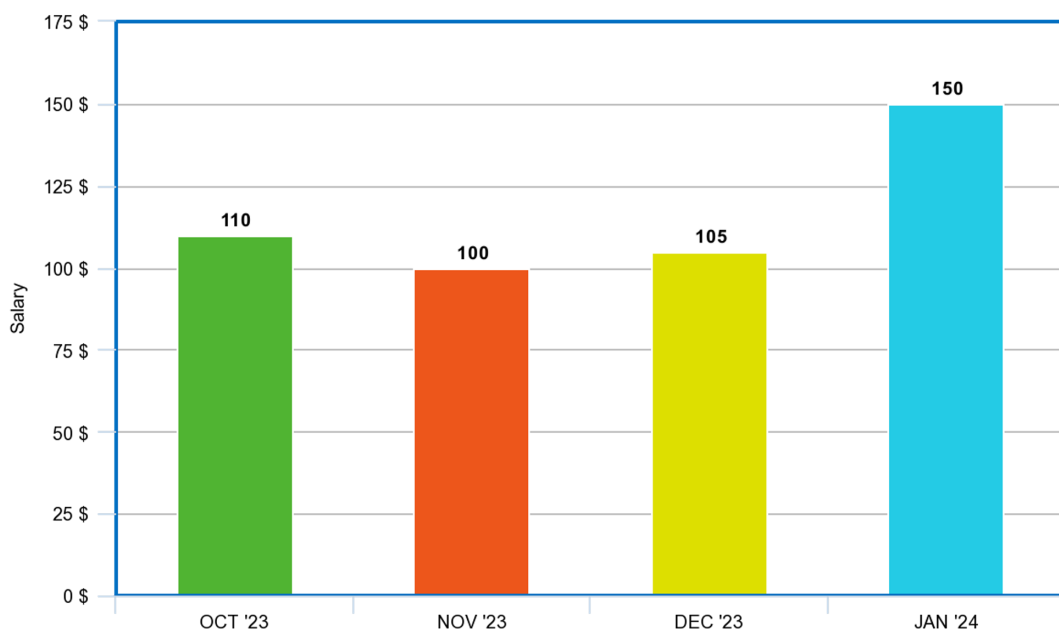
*As you can see, in the **user** or **people** collection, you may need to add **isVerified** field. For simplicity, let's say if the field doesn't exist, the employee is also considered not verified.*

- 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 bonus section **🟡**].

/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 the 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

On this page, HR can see all the rows from all the employees on the [/work-sheet](#) page. It's an amalgamation of all the work submitted by employees. Now, he 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 that month.

[See Optional section ]

For Employee [Private] [Employee Only]

/payment-history

The monthly history of salary paid by HR will be shown in the table. 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

You may use pagination or infinite scrolling if it has more than 5 rows.

/work-sheet

There will be a form and a table on this page. This form could fit 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 (Use DatePicker component)
- Add / Submit button

Clicking the Add / Submit button will add the newest done work to the table at the start (without reloading the page) and store the information in the DB as well.

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.

[Checkout Bonus part ]




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.


You Must do these

- **Commits & README**
 - Minimum 10 meaningful git commits on the client-side repository.
 - Minimum 5 meaningful commits on the server-side repository.
- Create a **README.md** for client-side and write about your project (at least 5 bullet points). ** Remember to add your client-side live link to your website in the **README.md** file.
- After reloading the page of a private route, the user should not be redirected to the login page.
- Make the website responsive. Make sure the site looks different on desktop and mobile responsive. Tablet responsive is optional.

Bonus

-  Make the payment operation using a Payment Gateway
-  Don't allow payment for the same month/year twice! You don't want to pay an employee twice a month, do you?
-  There will be a button which will toggle between Table View and Card Grid View on the same page.

Optional Tasks

- Don't use Daisy UI. Instead, use other UI component libraries such as Material Tailwind, 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
-  Add a section above the table where the summation of work hours is shown. The summation must be changed based on the filtered data. Pagination shouldn't affect the sum, even some rows go to the next page.

What to Submit

Assignment Category: assignment12_category_0004

Admin email:

Admin password:

Front-end Live Site Link:

Client Side Github Repository Link:

Server Side Github Repository Link: