smart-hrms

A role-based Human Resource Management System (HRMS) built using the MERN stack as part of the CDAC final project. Includes employee management, leave tracking, and admin panel.

Attendance

While showing the employee dashboard:

"Here we are logged in as an employee.
The system automatically loads attendance records in descending order, meaning the latest dates appear first.

Right now, you can see there's no record for today, so the Check In button is visible.
When I click Check In, the system immediately creates today's record, marks the status as Pending, and hides the check-in button."

Show after check-in:

"After check-in, the Check Out button becomes available.
Once the employee clicks Check Out, the system automatically calculates the total duration in minutes between check-in and check-out."

Show table data change:

"The status initially starts as Pending and can later be updated by the logic for Full Day, Half Day, or Rejected depending on company rules on durations."
For now we have 480minutes for Full Day. and 240 minutes for Half Day.

Switch to admin login:

"Now let's log in as an admin.
The admin panel provides two viewing modes: View by Date and View by Employee.

In **View by Date mode**, I can select a specific date and see all employees' attendance for that day — including their check-in, check-out, total duration, and status.

In **View by Employee mode**, I can choose an employee's name from a dropdown and see all their attendance records across different dates."

While switching between modes:

"The filtering logic for both modes is handled entirely on the client side for faster interaction."

Leave Management

View all Leave

First, let's look at the employee's view. Here, an employee can see all their leave requests listed in a table. They can filter the leaves based on status — Pending, Approved, or Rejected. There's also a search box to filter leaves by the reason."

Add a new Leave Request

To apply for a new leave, the employee fills out this form. They select the leave type, such as Casual or Sick, choose the 'From' and 'To' dates, and provide a reason. Note that the system validates the dates to ensure the 'From' date is not after the 'To' date. Once submitted, the leave request appears in their list with a status of 'Pending'."

Editing or Deleting Leave Requests

"Employees can edit or delete their leave requests only when the status is 'Pending'. Once a leave is approved or rejected, editing or deleting is disabled to maintain records' integrity. Here, I'll demonstrate editing a pending leave — the form populates with existing data for easy updates."

Admin - Viewing all leave requests

Now switching to the admin view, admins can see all leave requests from every employee. They can search by employee name and filter leaves by status to quickly find specific requests."

Approving or Rejecting Leave Requests

"For pending leave requests, admins have the option to Approve or Reject. Upon clicking either, a comment box appears, prompting them to provide a reason for their decision. This comment is saved alongside the leave request. After confirming, the leave's status updates accordingly in the list."

Viewing Leave Details

"Admins can also view detailed information of any leave request — including employee name, leave type, dates, status, and comments — in a popup modal for easy reference."

Admin - Salary Management

When the component loads, it fetches the list of employees and existing salary records from the server using API calls. This data is stored in state variables to be displayed and manipulated."

Salary Form Overview

At the top, there is a form for adding or updating salary records. The form fields include:

Selecting the employee from a dropdown list populated with employee names.

Entering the salary amount.

Choosing the date from which the salary is applicable.

Optionally entering PF deduction amount.

The form validates that employee, amount, and applicable date are mandatory fields.”

Adding a New Salary Record

“To add a new salary, fill in the form fields and click ‘Add Salary’. The form data is sent to the server through an API call. Upon success, the form resets and the salary list refreshes to show the new entry.”

Editing an Existing Salary Record

“In the salary table below, each record has an ‘Edit’ button. Clicking it populates the form with the existing data, switching the form into edit mode. After making changes, clicking ‘Update Salary’ saves the changes via an API call and updates the list accordingly. There’s also a ‘Cancel’ button to exit edit mode without saving.”

Employee - View Salary Detils

Employees can view their own salary records showing their current and past salary amounts, applicable dates, and deductions like PF.

Download Salary Slips:

Employees can download or print their monthly salary slips for record-keeping and official purposes.

No Edit Access:

Employees cannot edit or modify salary information — this is managed only by the admin.

MAnage EMployess

Our component uses React hooks like useState and useEffect to manage state and side effects.
On component mount, we fetch two sets of data:

Employees

Designations (job titles or roles)

We store those in state variables and render a searchable table of employees with their details like name, email, phone, department, and designation.

The search box filters employees by name as you type.
It’s case-insensitive and updates in real-time using React’s controlled input pattern.
Filtered employees are then displayed in the table below.

Click the Add New Employee button to open a modal with a form.
This form has fields like name, email, DOB, gender, phone, address, and designation.
On submission, it calls the API to add the employee and updates the list on success.
If adding fails, an alert notifies the user.

Each employee row has an Edit button.
Clicking it opens the same modal but pre-filled with that employee's details.

You can update any field, and submitting calls the update API.
Once updated, the employees list refreshes with the new data.

The Delete button prompts for confirmation to avoid accidental deletions.
On confirmation, the employee is removed via an API call and the list updates instantly.

Department

There’s a search box at the top where you can filter departments by name in real time.
If no departments match, it shows a friendly message: "No departments found."

Click Add New Department to open the modal with a form to enter the name and description.
The form requires both fields. On submit, it calls the API to create the department.
If successful, the modal closes and the table refreshes with the new data.
If fields are empty or API call fails, you get an alert or error message.

Click Edit next to any department to open a pre-filled modal form.
You can update the name and description, then save.
After updating via the API, the modal closes and the table refreshes.
Validation and error handling are similar to the add form.

The Delete button prompts for confirmation to avoid accidental deletions.
If confirmed, it calls the API to delete the department.
If deletion fails due to constraints (like linked designations), an alert explains why.

Designations

There’s a search bar at the top that filters designations by their name as you type, showing only matching results.

Click the Add New Designation button to open a modal form.
Here, you enter the designation’s name, description, and select the associated department from a dropdown.
On submitting the form, the new designation is sent to the API, and the list refreshes to show the update.

Each designation row has an Edit button. Clicking it opens a modal with the designation’s current data pre-filled.
You can update the name, description, or department, then save changes.
After submission, the list refreshes with the updated information.

Clicking the Delete button removes the designation after confirmation.
If deletion fails due to references elsewhere (like assigned users), an alert informs you about the issue.

The designations are displayed in a table with columns for ID, name, description, department name, and actions.
Actions include Edit and Delete.

Profile

Once loaded, the profile displays key details such as:

Full Name

Email

Phone

Department

Designation

the same format is for employee also

Settings

This component features a simple form with three fields:

Current password

New password

Confirm password

On form submission, we first check if the new password and confirm password match.
If they don’t, we show an error right away without making the API call.

If validation passes, the component calls the updatePassword API with the form data.
While waiting for the response, a loading state disables the submit button and changes its label to “Updating...”.
On success, a success message is shown and the form resets.
If the API call fails, the error message from the server is displayed.

Future Scope

- we can integrate features like geolocation-based check-in, automatic late detection, and exporting reports as Excel files."