**Introduction**

Human Resource Management (HRM) software development is the process of creating digital solutions to streamline, automate, and enhance HR processes within organizations. As businesses grow, managing tasks like employee recruitment, onboarding, payroll, performance evaluations, and attendance tracking becomes increasingly complex. HRM software addresses these challenges by providing an integrated platform where HR teams and employees can manage essential functions more efficiently.

Modern HRM systems incorporate various modules—such as employee management, performance tracking, task assignments, and reporting tools—that empower HR professionals to make data-driven decisions, improve employee engagement, and foster overall organizational growth.

**FrontEnd Requirements**

When designing a **Human Resource Management (HRM) system**, the front-end requirements should focus on usability, accessibility, and a clean, intuitive interface to handle a wide range of HR functions.

**User Interface Design(UI)**

**Clean and Intuitive Layout:** The UI should be easy to navigate, with a clear structure for dashboards, menus, and key actions.

* **Dashboard:** A main landing page with key stats (e.g., employee attendance, leave requests, recent activities).
* **Navigation Bar:** A responsive sidebar or top navigation that provides access to key modules (e.g., Employee Management, Payroll, Time & Attendance, Reports, etc.).
* **Responsiveness:** The design must be responsive to ensure usability across various devices like desktops, tablets, and mobile phones.
* **Consistency:** Uniform use of colors, fonts, and design elements across the platform to maintain a professional and cohesive look.

**Front End Technology.**

Candidates need to develop a User friendly Interface using any Following Frontend Framework

1. Bootstrap
2. Reactjs
3. Angular

**Note:** Developing Reactjs or Angular based Application will be a Added Advantage for the candidate.

**Backend Requirement**

Designing the backend for a **Human Resource Management (HRM) system** requires careful planning to ensure scalability, security, performance, and seamless interaction between the system's core functions. Below are the key **backend requirements** for an HRM system, covering database architecture, security, API management if required and business logic implementation.

**Database Design and Management**

The HRM system needs a well-structured and normalized relational database to handle the extensive data it collects and processes.

* **Relational Database (SQL):** Use a relational database like MySQL, PostgreSQL, or SQL Server to store and manage structured data such as employee details, payroll records, and performance metrics.
  + **Key Tables:**
    - Department: Stores Department information
    - Role: Stored the role assigned to a User(Admin, Manager,Team-Leader and Employee) .
    - User: Manages User.
    - Performance: Tracks employee reviews, feedback, goals, and accomplishments.
    - Task: Manages Task assigned for the User.
    - Leave: Manages leave requests, approvals, and balances.

Apart from above table there can be more number of tables as per the requirements, candidate need to design those tables and establish the relationship if required with other tables.

**Relationship Management:** Establish relationships between tables (e.g., one-to-many relationships between User and Task   table).

**Github**

1. Candidate must create a repository on the Github.
2. After completion of Each module candidate. must push the code to the repo, so that Internship evaluator at any can evaluate updated code.
3. Readme file must be designed Properly.

**Deployment**

1. candidate must deploy the project on any free server which support the backend Framework.
2. After completion of every module or topic candidate need to deploy it on server.
3. Application must be running 24 x 7 on live server after completion of 1st module.
4. Evaluator can check your domain link and completed functionality link at any time.

**Introduction**

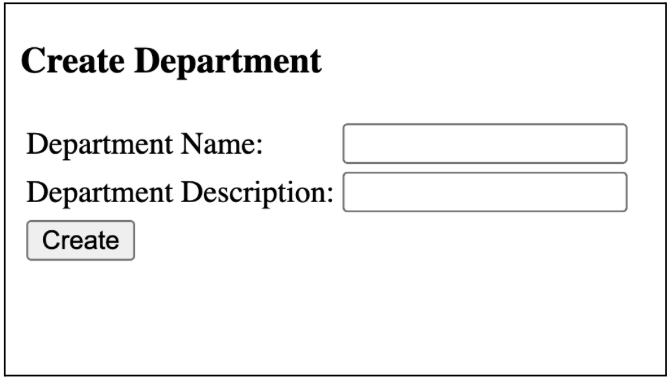
The Department Creation Module is a crucial part that allows administrators to organize and structure the company into various departments.This module enables the creation, modification, and management of departments within the organization, providing a clear hierarchy and improving employee management.

**Table Name**: department

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| dept\_id | int(pk,auto\_increment) | Unique identifier for each department |
| dept\_name | varchar(100) | Name of the department(IT,Sales etc) |
| description | varchar(300) | Small description about the department |
| created\_at | datetime | Timestamp when the department was created. |
| updated\_at | datetime | Timestamp when the last department was updated. |
| status | varchar(50) | This is a boolean field. Default value is True |

Department Management System **Add Department**Only Admin can create a new department.

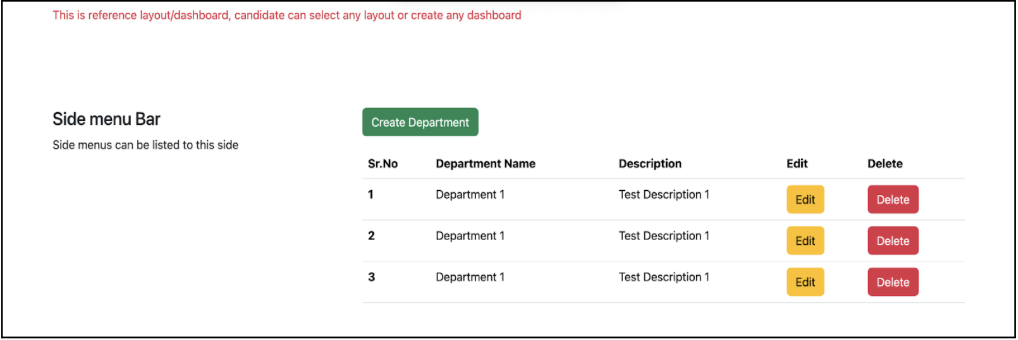
**Reference Form**



candidate need to create UI as shown above. Above UI is for reference purpose. Candidate  can create their own UI with given fields.

**Dashboard Creation**

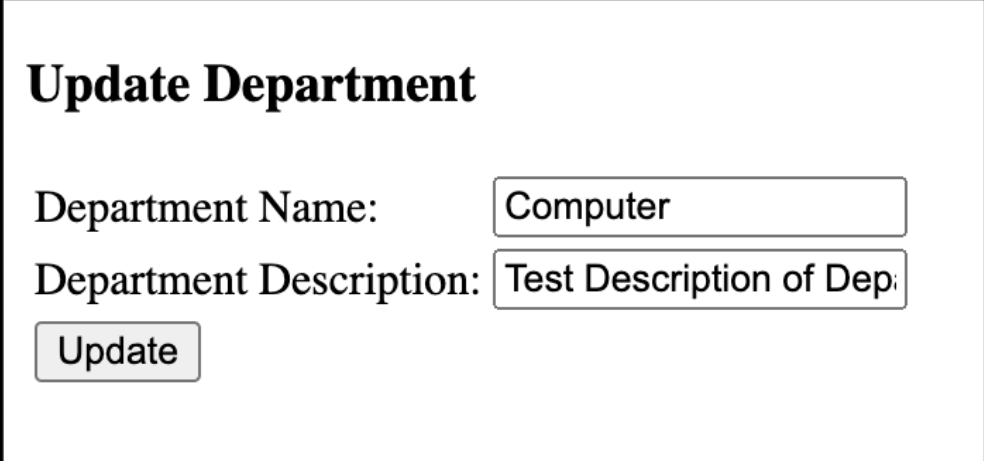
Candidate need to design the Dashboard where admin can manage department add , update and delete functionalities.



Additionally candidate can create search box for department search provision to User.

**Modify Department**

Only Admin can modify the department's details such as Department name and description of the department.



After updating department information user must be directed to Dashboard.

**Delete Department**

Admin can delete a department. Implement soft delete by providing a column status with boolean values for active or inactive in the database for department activation or inactivation. While delete or making department inactive, a  warning message  pops up must be given to the admin as making department inactive will cause department inactive for the employee linked with the department, so first assign different departments to those employees and then make department inactive.

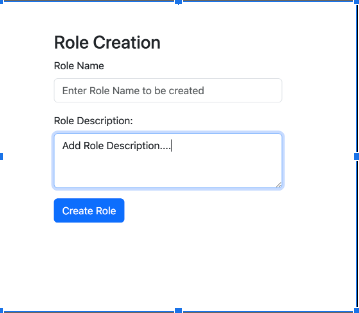
**Introduction**

The **Role Creation Module** in the HR management system is an essential feature that enables administrators to define and manage different roles within the organization. Roles define the level of access and permissions that employees have in the HR system, ensuring a secure and organized way of managing the company's workflow.

This module allows administrators to create new roles, such as Admin, Manager, Team Leader, and Employee. By assigning roles to employees, the system ensures that individuals have access only to the data and functionality necessary for their job responsibilities.

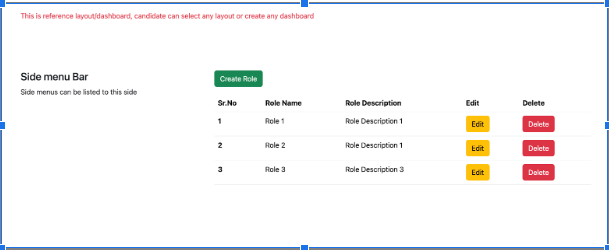
Role management system

1. **Adding a Role**: Only Admin can add a role.

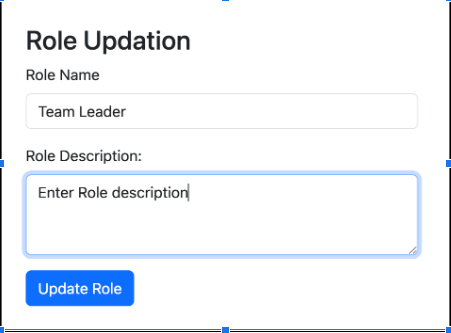


**Dashboard  Role Creation**

Dashboard contains a view of all the roles created with edit and delete button functionality for each role. With the Create Role button admin can access Create role form to create a new role as discussed above.



**Update  Role Details**



After Update user must be automatically directed to the Dashboard

**Delete Role**

Only Admin has permission to delete roles. Implement soft delete by providing a column status in Database table as shown below in the database section with Boolean values. While delete or making role inactive, a  warning message  pops up must be given to the admin as making role.

Table name: roles

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| role\_id | int(pk,auto\_increment) | Unique identifier for each role |
| role\_name | varchar(100) | Name of the Role(Admin,Manager,Team leader and Employee) |
| description | varchar(200) | Small description about responsibility of the role. |
| created\_at | datetime | Timestamp when role was created |
| updated\_at | datetime | Timestamp when role was last updated |

2. Assignment

Top of Form

After completion of Role Management System, candidate need to upload the code or push the code to Github.

Also candidate need to deploy the application on any free server  and prepare the End User Documentation for the module developed.

With respect to the Role Management System module completed, please provide following details as a part of Assignment.

1. Github Repository Link
2. URL of module Hosted on free Server.
3. Prepare the end user documentation.

Note: Upload Assignment in PDF format.

Bottom of Form

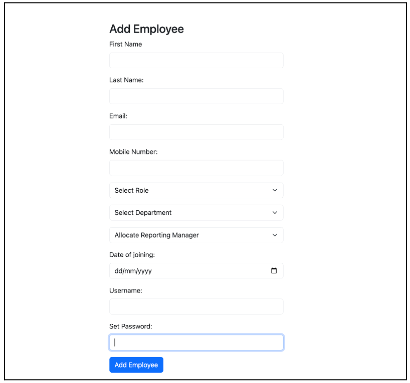
**Introduction**

The Employee Creation Module is a core feature of HRM software that enables administrators and HR personnel to add new employees to the system seamlessly. This module streamlines the process of onboarding employees by capturing essential details such as personal information, role assignment, and reporting structure.

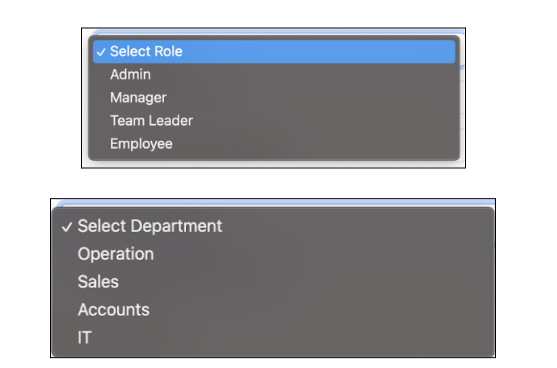
During the employee creation process, HR can assign a specific role (e.g., Admin, Manager, Team Leader, or Employee) to define the employee's access rights and responsibilities within the system.

Additionally, the module allows for the assignment of a reporting person (or manager) to establish the reporting hierarchy. This ensures that the employee is connected to the appropriate supervisor or team leader, facilitating clear communication and a structured chain of command within the organization.

**Adding a Employee**



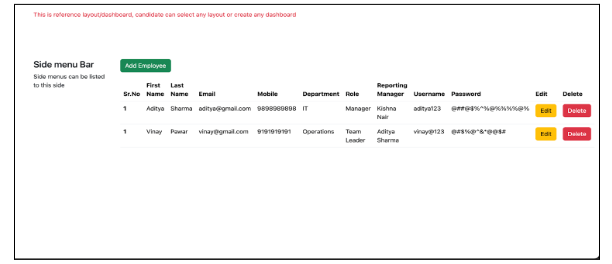
Select role , select Department are the drop down with their respective values in roles, department  tables.



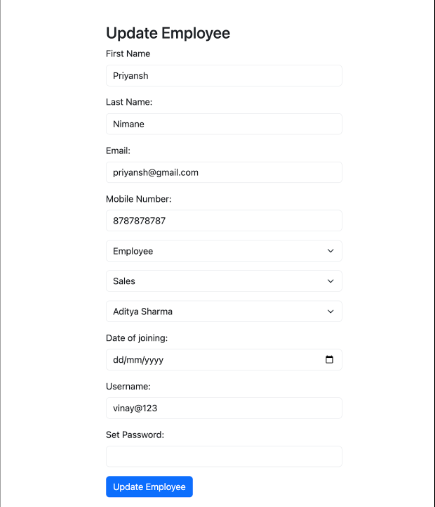
Allocating the reporting manager is also drop down with all the data fetching from the users table in the database. User table is used to store employee information.

**Dashboard for User Creation**

Admin and HR department manager can access this Dashboard. Dashboard contains provision to add employees, to update employee details and to delete employees on his resignation or termination.



**Update Employee Details**



**After update user must be directed to Dashboard.**

**Database Design**

Table name: User

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| employee\_id | int(pk,auto\_increment) | Unique identification of user |
| first\_name | varchar(100) | Employee first name |
| username | varchar(100) | Employee Username |
| Password | varchar(100) | Employee Password |
| last\_name | varchar(100) | Employee last name |
| email | varchar(100) | Employee Email |
| mobile | varchar(100) | Employee mobile |
| dept\_id | int(fk) | Foreign key taking reference from id column of department table. |
| role\_id | int(fk) | Foreign key taking reference from id column of roles table. |
| reporting\_manager\_id | int(fk) | Foreign key taking reference from id column of User table. |
| date\_of\_joining | date | Date at which employee joined the company. |
| created\_at | datetime | Timestamp at which User/ employee created for first time. |
| updated\_at | datetime | Timestamp at which User/employee was last updated. |

Assignment

Top of Form

After completion of  Employee Management System, candidate need to upload the code or push the code to Github.

Also candidate need to deploy the application on any free server  and prepare the End User Documentation for the module developed.

With respect to the Employee Management System module completed, please provide following details as a part of Assignment.

1. Github Repository Link
2. URL of module Hosted on free Server.
3. Prepare the end user documentation.

Note: Upload Assignment in PDF format.

Bottom of Form

**Introduction**

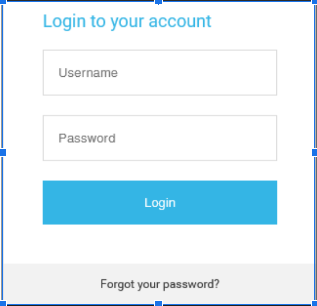
The Sign-In module is an essential component of the HR management system. It serves as the gateway for employees to interact with the system, ensuring that only authorized personnel can enter the System.

By enforcing robust security measures, such as two-factor authentication and password encryption, the module ensures that sensitive HR data remains protected from unauthorized access.

User Authentication System

**Username and Password Authentication**

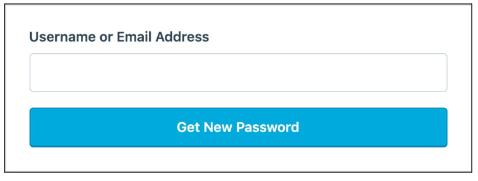
User Enter the Unique Username and password to access the System. This functionality validates the username and password stored in the database.

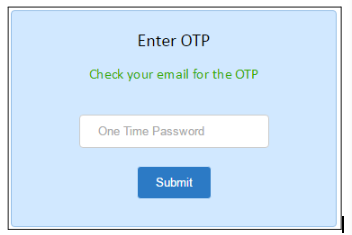


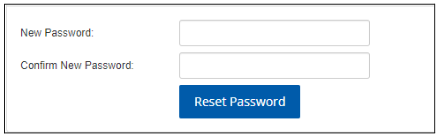
**Password reset**

Users enter a Registered email id with the system. A reset password otp is being sent to the Registered email address[Hint-Gmail integration].Entered OTP is validated and then is allowed to enter a new password.

After entering a new password, the user is again directed to the login page, where the user again logged in with the new password.







**Database Design**

Table name: User

|  |  |  |
| --- | --- | --- |
| **Column name** | **Datatype** | **Description** |
| employee\_id | int(pk,auto\_increment) | Unique identification of user |
| first\_name | varchar(100) | Employee first name |
| username | varchar(100) | Employee Username |
| Password | varchar(100) | Employee Password |
| last\_name | varchar(100) | Employee last name |
| email | varchar(100) | Employee Email |
| mobile | varchar(100) | Employee mobile |
| dept\_id | int(fk) | Foreign key taking reference from id column of department table. |
| role\_id | int(fk) | Foreign key taking reference from id column of roles table. |
| reporting\_manager\_id | int(fk) | Foreign key taking reference from id column of User table. |
| date\_of\_joining | date | Date at which employee joined the company. |
| created\_at | datetime | Timestamp at which User/ employee created for first time. |
| updated\_at | datetime | Timestamp at which User/employee was last updated. |

Assignment

Top of Form

After completion of User Authentication  System, candidate need to upload the code or push the code to Github.

Also candidate need to deploy the application on any free server  and prepare the End User Documentation for the module developed.

With respect to the User Authentication  module completed, please provide following details as a part of Assignment.

1. Github Repository Link
2. URL of module Hosted on free Server.
3. Prepare the end user documentation.

Note: Upload Assignment in PDF format.

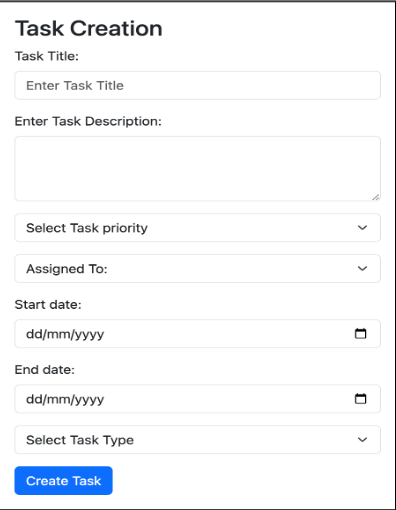
Bottom of Form

**Introduction**

A **Task Management System** in HRM software is a critical tool that enables organizations to assign, track, and manage employee tasks efficiently. This module deals with task creation, assigning tasks to employees and tracking the task of the employee.

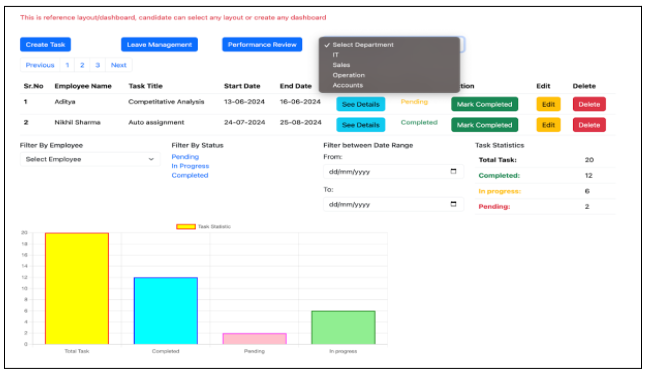
Task Management System

**Task creation and Task assignment**



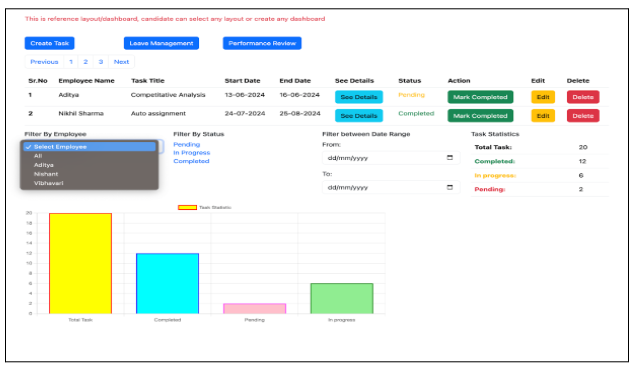
* Select Task Priority is a drop down menu with options: High,medium and low.
* Assigned to contains name of the employee to whom Team leader or Manager or admin wants to assign the task.
* Drop down of **Assigned To** must show only those employees who are reporting to the logged in user.
* Select Task Type drop down contains two options as Individual and Team.

**Dashboard Admin**

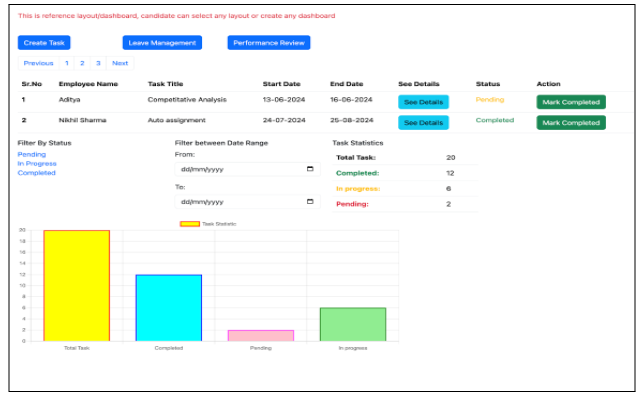


* + Create task button allows the user to create a task and assign it to team members. After creating the task, the user returns to the above dashboard.
  + Pagination is applied to see 10 records per screen.
  + Filter by Employee filters the task to be shown as per the selected employee.
  + Dropdown to select employees contains only names of those employees who are reporting to the logged in users.
  + Filter by status filters the task as per the status such as Pending, Completed and In progress task.
  + See details button , provide users to see the detailed view of the task along with its description.
  + Filter between date ranges works on the start date filter. That displays records based on the dates provided in the To and From section.
  + The Statistical section shows the number of completed tasks, pending and in progress tasks with respect to selection either for specific employees or for all the employees.
  + Multiple filters selection must be supported by the dashboard, meaning if a specific employee is selected by the user(Team leader/Manager/Admin)and the user also clicked on the completed option from the Filter by Status then the dashboard must show that selected employee completed the task.

**Dashboard Manager/Team Leader**



**Employee Dashboard**



**Update Task Details**

When the User(Admin/TL/Manager) clicks on the edit button, it opens the view to update Task details such as Task Title, Task Description, Assigned To, Select Task priority, Start date,end date and task Type.

After Updating task details, the user must be directed to the dashboard.

**Delete Task**

When the User(Admin/TL/Manager) clicks on the Delete button, there must be a pop for providing a warning for deletion with a confirm button to delete the task.

**Database Design**

**Table:** Task

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| task\_id | int(pk,auto\_increment) | Unique identification of task |
| task\_title | varchar(100) | Describe the title of the task |
| task\_description | varchar(300) | Provide details of the task |
| task\_priority | varchar(200) | Stored priority of the task as High,Medium and Low. |
| start\_date | date | Start date of the task |
| end\_date | date | Due date of the task |
| task\_type | varchar(50) | Store type as Individual or Team |
| created\_at | datetime | Timestamp at which the task is created |
| updated\_at | datetime | Timestamp at which task was updated last time. |

**Table:** task\_assignment

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| assignment\_id | int(pk,auto\_increment) | Unique identification of task |
| task\_id | int(fk) | Reference of id column in task table |
| employee\_id | int(fk) | Reference of id column in employee table to whom task is assigned |
| assigned\_by | int(fk) | Reference of the id column in employee table who assign the task |
| assigned\_date | datetime | Timestamp at which task is assigned |
| status | varchar(200) | Status of the task such as Pending, In progress and Completed. |
| completed\_at | datetime | Timestamp at which the task is marked completed |

Assignment

Top of Form

After completion of Task Management System, candidate need to upload the code or push the code to Github.

Also candidate need to deploy the application on any free server  and prepare the End User Documentation for the module developed.

With respect to the Task Management System module completed, please provide following details as a part of Assignment.

1. Github Repository Link
2. URL of module Hosted on free Server.
3. Prepare the end user documentation.

Note: Upload Assignment in PDF format

Bottom of Form

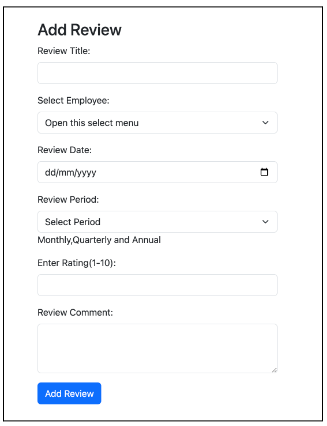
**Introduction**

Review management systems allow organizations to conduct periodic reviews over the task allocation to employees and provide them ratings which helps to keep progress of their work efficiency. Performance review report  helps to make a decision which department or employee is performing better and where training is required.

Performance Management System

**Add a Review**

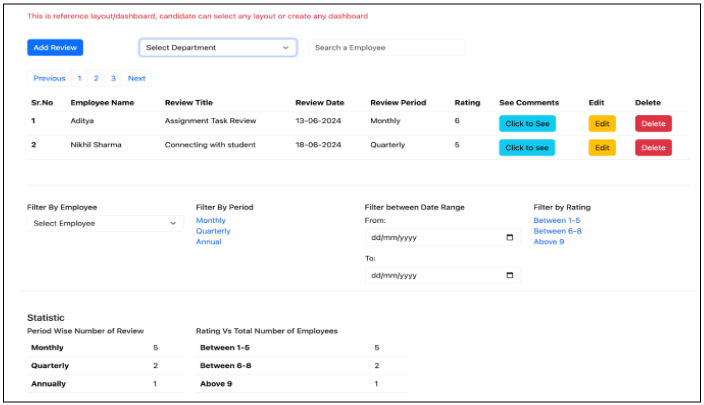
            This functionality is accessed by Admin/TL/manager to add review details when review is being conducted with the team members.



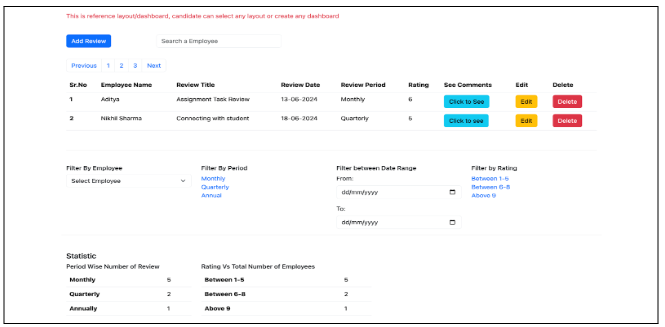
Select employee drop down must show a list of only those employees who are reporting to the logged in Admin/Manager/TL.

Review period contains three options viz. Monthly, Quarterly and Annual.

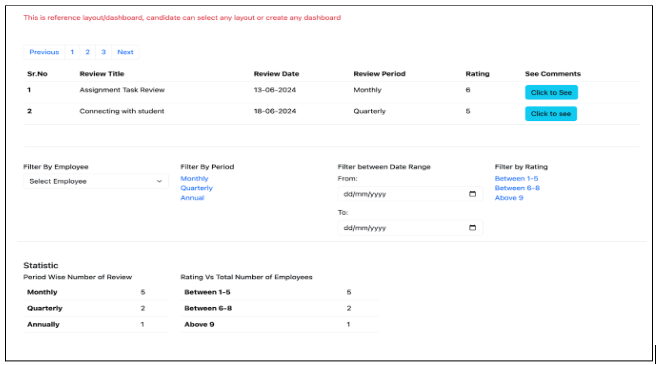
**Admin Dashboard**



**Manager/TL Dashboard**



**Employee Review Dashboard.**



Database Design

**Database Design**

Table Name: Performance\_review

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Datatype** | **Description** |
| review\_id | int(pk,auto\_increment) | Unique identification of the review |
| review\_title | varchar(100) | Title or small details of the review taken. |
| review\_date | date | Date on which the review is taken |
| employee\_id | int(fk) | Reference taken from the employee table for whom review is taken |
| reviewed\_by | int(fk) | Reference taken from the employee table who takes the review. |
| review\_period | varchar(100) | It contains whether review is monthly,quarterly and annually |
| rating | int | Rating given to employee between range 1-10 |
| comments | Varchar(300) | Any extra comment given by the reviewer. |
| created\_by | datetime | Timestamp at which review was taken for the first time |
| updated\_by | datetime | Timestamp at which the review was last updated. |

Assignment

Top of Form

After completion of Performance Management System, candidate need to upload the code or push the code to Github.

Also candidate need to deploy the application on any free server  and prepare the End User Documentation for the module developed.

With respect to the Performance  Management System module completed, please provide following details as a part of Assignment.

1. Github Repository Link
2. URL of module Hosted on free Server.
3. Prepare the end user documentation.

Note: Upload Assignment in PDF format.

Bottom of Form

**Introduction**

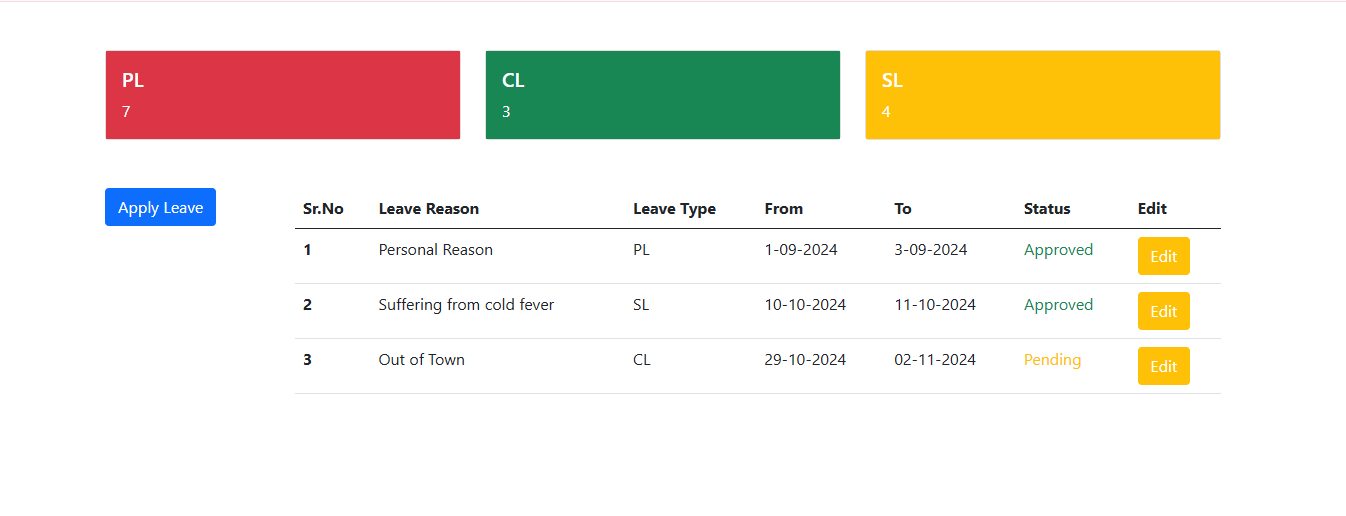
 Leave Management Systems are integral components of Human Resource Management in modern organizations.

These systems automate the tracking of employee   leave requests, and approvals, thereby improving organizational efficiency.

Funcationalities-Leave Management

**Functionalities**

* Leave Dashboard-Employee



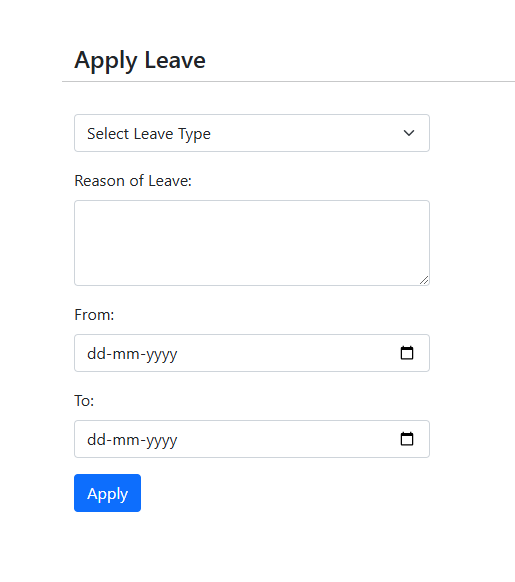
Dashboard must show all leave balance with all the leaves taken with the details shown in above images.

Employees can have edit access to the leave before it gets approved by the manager. Once the manager approves the leave, the employee edit button must be disabled.

Deleting leave is not available to an Employee. Employees  can request to delete leave to the reporting manager or admin or HR department.

* Apply Leave

By clicking on the apply leave button on the dashboard, the employee is directed to the following apply leave form as shown below.

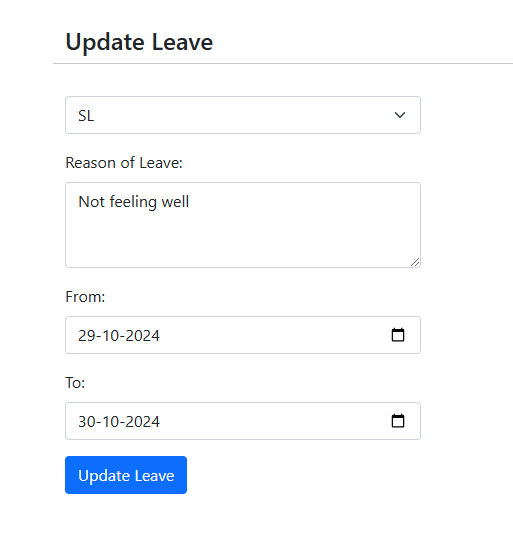


Select leave contains drop down with three options PL,CL and SL.

After applying this leave, use must be directed to his dashboard where the user can see the applied leave. By default leave status will be pending.

This applied leave is also visible to the reporting manager dashboard for approval and rejection. When the reporting manager approves it or rejects it.

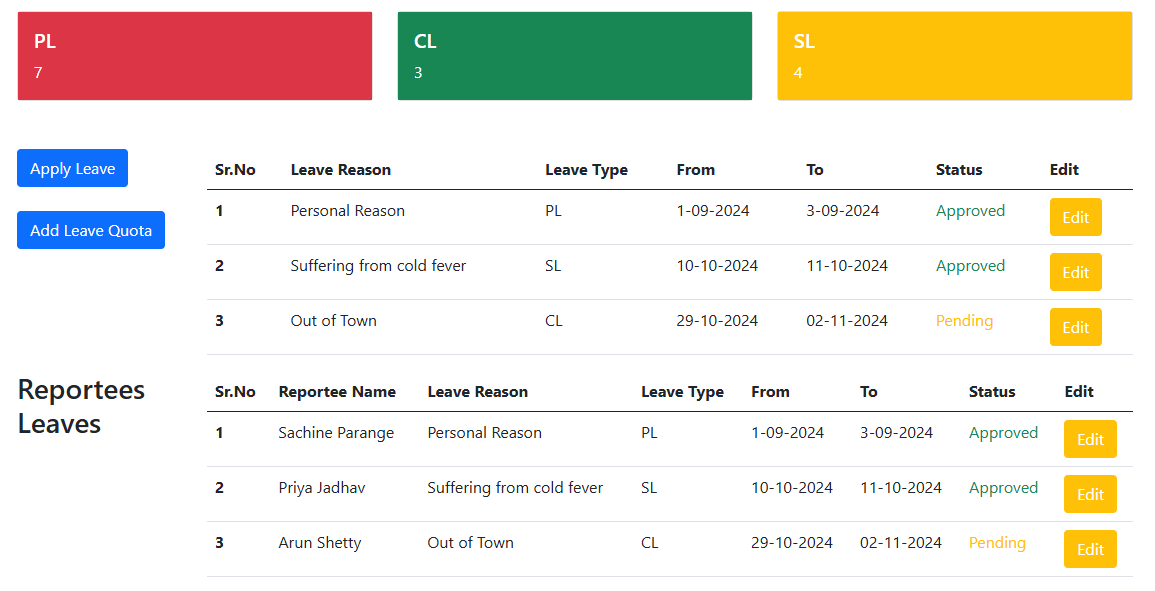
**Update Leave**



Employees can update the leave status before it is being approved or rejected by the reporting manager. Employees will not be able to edit the leave after it is being approved or rejected by the manager.

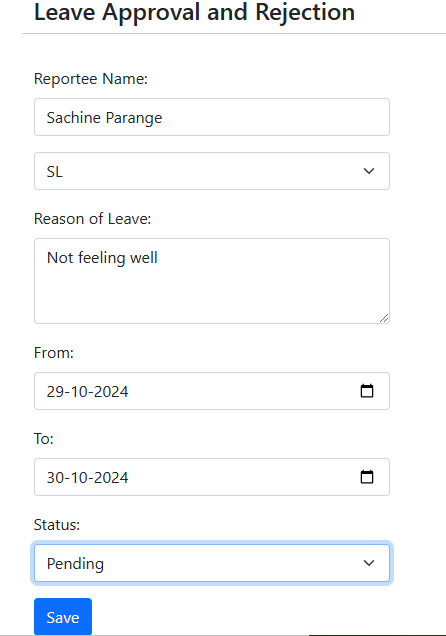
After updating the leave status, User is directed to the dashboard.

**Leave Dashboard-Admin**



Admin or manager dashboard contains two sections, Manager or admin Leave section and Reportees leaves where manager or admin can approve or reject the leave by clicking edit button.

After clicking the edit button in the reportees leaves section , the following form opens where Manager or admin can approve or reject the leave.



**Leave Balance**



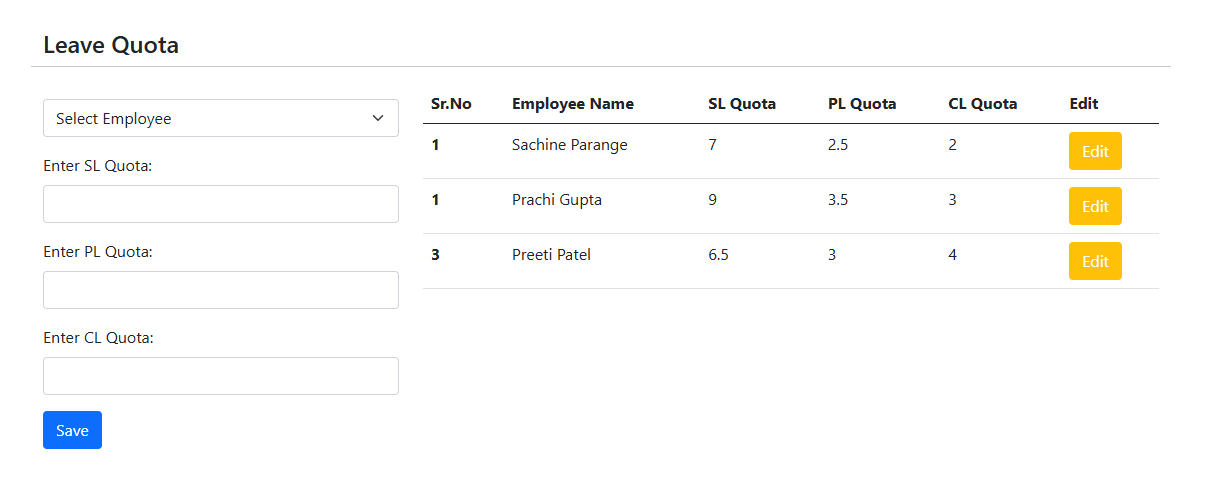
Leave balance must get updated as per the leaves consumed by the User.

**Leave Quota Functionality**

Leave quota for SL,PL and CL are updated by HR department managers and admin.

The Add Leave Quota button must be visible to HR managers and admin of the organization.

When a user clicks on add leave quota, user is directed to the add leave quota dashboard where user gets a form to add leave quota for the year for a particular employee selected from the dropdown as shown in the image.



Drop down contains a list of employees.

Edit functionality is given to edit or update leave quota for a particular employee. Pagination needs to be used to show a limited number of employees to be displayed per page.

**Database Design**

Table: Leave Table

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| leaveid | int(pk) | Unique identifier for leave request |
| employeeid | int(fk) | Link to employee table |
| leave\_type | enum | SL,CL,PL,LWP |
| reason | varchar(200) | Leave reason |
| start\_date | date | Start date of leave |
| end\_date | date | End date of leave |
| total\_days | int | Total number of leaves days |
| status | enum | approved,rejected,pending |
| approved\_by | int(fk) | Link to manager or HR in employee table |

**Table: Leave Quota Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Datatype** | **Description** |
| quotaid | int(pk) | Unique identifier for leave quota |
| employeeid | int(fk) | Link to Employee Table |
| leave\_type | ENUM | Type of Leave(e.g sick leave, casual leave,PL and LWP) |
| total\_quota | int | Total leaves allowed per year |
| used\_quota | int | Used Leave quota |
| remain\_quota | int | Remaining Leave quota |

Assignment

Top of Form

After completion of Leave Management System, candidate need to upload the code or push the code to Github.

Also candidate need to deploy the application on any free server  and prepare the End User Documentation for the module developed.

With respect to the Performance  Management System module completed, please provide following details as a part of Assignment.

1. Github Repository Link
2. URL of module Hosted on free Server.
3. Prepare the end user documentation.

Note: Upload Assignment in PDF format.

Bottom of Form