

A CRUD Operation Web Application (School Employee Management System)

Frontend Development Advanced Project

Developer: *Salamatu Kabiru Adamu*

Tech Stack: ReactJS · Node.js · Express.js · MongoDB Atlas · JWT
Authentication

Frontend Hosting: [Vercel](#)

Backend Hosting: [Render](#)

Project Goal

To build a **React-based CRUD application** that allows schools to efficiently manage employee information and leave requests using a **secure and role-based interface**.

The system provides:

- Seamless **data management**
- **Role-based access control**
- **Secure login** for all users
- A clean, user-friendly and modern user interface

User Roles and Permissions

Role	Permission
Administrators	Can create, edit, and delete any employee record; manage leave requests; assign departments and roles.
Managers	Oversee department staff and approve/reject leave requests
Employees	Submit leave requests and view personal details

Authentication & Access Control

- Authentication is handled using **JSON Web Tokens (JWT)** for secure session management.
- Passwords are **hashed using bcrypt.js** before storage, ensuring user data privacy.
- **Role-based middleware** in Express.js ensures each user only accesses what their role permits.

Employee Creation Logic

When a new employee is added by an Admin:

- A record is created with full name, department, position, and role.
- The system **automatically generates a default password: defaultpassword123**
- The password is immediately **hashed using bcrypt** before being saved in MongoDB.
- Each employee has a **unique employee ID** for for login.

This feature ensures Admins don't need to manually create credentials for every staff member, it's automated.

Leave Request Workflow

- **Employee submits a leave request** with the reason, start date, and end date.
- The request automatically sets its **status to “Pending.”**
- **Managers/Admins** can view all pending requests and either **Approve** or **Reject**.
- Once a decision is made, the request is **updated** on the employee's dashboard.

Core Features 1

Employee Management (CRUD)

- Create, view, update, and delete employee records
- Auto-generate unique Employee IDs
- Default password assigned and encrypted automatically
- Department and role assignment
- Search and filter functions for large data sets

Core Features 2

Leave Request Management (CRUD)

- Employees submit leave requests with reason, start, and end dates
- Managers and Admins can approve or reject requests
- Automatic status updates and visibility control

Core Features 3

Authentication & Authorization

- Login secured with JSON Web Tokens (JWT)
- Passwords hashed with bcrypt.js
- Role-based permissions for Admin, Manager, and Employee

Core Features 4

UI/UX Design

- Built with Material UI for modern, accessible visuals
- Responsive across all devices
- Dashboard layouts differ based on role
- Color-coded indicators for quick status recognition

Tech Stack

Layer	Technologies
Frontend	ReactJS, React Router, Material UI
Backend	Node.js, Express.js
Database	MongoDB Atlas
Authentication	JWT, bcrypt.js
Hosting	Vercel (Frontend), Render (Backend)
State Management	React Hooks (useState, useEffect)

Test Login Details

Use the credentials below to explore the platform across different user roles. Each role has specific permissions to demonstrate role-based access control.

Role	Username	Password
Admin	admin1	Admin123!
Manager	manager1	Manager123!
Employee	employee1	Employee123!
Any employee added by Admin	their employee Id	defaultpassword123

Project Links Summary

Item	Link
Live Frontend	https://school-employee-management.vercel.app/
Live Backend	https://school-employee-management.onrender.com/
GitHub Repository	https://github.com/salmakabir000/school-employee-management
Colour Palette	https://colors.co/3b82f6-2563eb-f97316-f472b6-dc2626-f9fafb-111827

Thank you for reviewing my project.