

An-Najah National University

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

Computer Engineering Department

Software Graduation Project

CoreHR

Human Resource Management System

Student:

Abdrahman nedal yahya

Ahmed abo shames

Supervisor:

Dr. Muhannad Al-Jabi

2026

Acknowledgment

To the doctors of the Computer Engineering Department at An-Najah National University, I want to say a heartfelt thank you for everything you have given throughout these years. The knowledge you shared with us, the patience you showed when we struggled, and the care you put into every lecture have shaped the way I think about technology and engineering. None of this would have been possible without you.

A special thank you goes to my supervisor, whose guidance kept this project on track even when things felt overwhelming. Your feedback, your patience with my questions, and your willingness to push me to do better made a real difference. I learned far more than just how to build software under your supervision.

To the friends and colleagues who helped along the way, whether by listening to me complain about a stubborn bug or by sharing an idea that unlocked something I had been stuck on for days, thank you. You made the long nights easier.

And finally, to my family. Without your support, your prayers, and your patience, none of this would have ever happened. You believed in me even on the days I did not believe in myself, and reaching this point is as much your achievement as it is mine.

This project is the result of a long journey, and I am grateful to everyone who walked any part of it with me.

Abstract

CoreHR is a complete Human Resource Management System that brings together everything an organization needs to manage its people into one place. Instead of jumping between spreadsheets, paper forms, and disconnected tools, HR administrators, managers, and employees can all work from the same platform, reachable through a web dashboard or a mobile app.

The system covers the full employee journey, starting from recruitment and onboarding, through everyday operations like attendance tracking, leave requests, payroll, and performance reviews, all the way to offboarding when an employee leaves the company. It was built using React.js with Vite for the web side, Flutter for the mobile application, and Node.js with Express for the backend, with SQLite handling the data.

One of the things that sets CoreHR apart is its permission system. Instead of giving everyone the same access, administrators can pick from 32 individual permissions and grant only what each user actually needs. This keeps sensitive data like salaries and personal records protected without making the system harder to use.

CoreHR also includes a few features that go beyond the basics. There is a smart insights engine that looks at the data and tells you things you might have missed, like which employees have been absent for several days in a row or which contracts are about to expire. A built-in chatbot answers employee questions in both English and Arabic, and the system sends real-time notifications whenever something important happens. The mobile app shows employees and managers different views, so each person sees only what is relevant to their role.

By the time the project was finished, the system had grown to 161 API endpoints, 35 database tables, 18 web pages, and 18 mobile screens. The goal was always to build something that could actually be used by a real company, especially smaller businesses in regions where affordable HR software with proper Arabic support is hard to find.

Table of Content

Acknowledgment	2
Abstract	3
Table of Content	4
Introduction	5
Motivation.....	6
Aims and Objectives	7
Constraints	9
Literature Review	10
Methodology	12
Non-Functional Requirements	12
Design.....	13
Implementation	20
Tools	22
Results	23
Results - Web Dashboard	23
Results - Mobile App (Employee)	30
Results - Mobile App (Admin)	35
Results and Discussion	40
Conclusion	42

Introduction

Managing people inside an organization has always been one of the harder parts of running a business. Even small companies end up needing to track who showed up to work, who is on leave, how much each person is supposed to be paid, what equipment they have been given, how their performance is going, and a hundred other details that pile up over time. Many companies still try to handle all of this with spreadsheets, email chains, and paper forms, and the result is usually the same: things get lost, mistakes happen, and both HR staff and employees end up frustrated.

CoreHR started from a pretty simple idea. Instead of having a leave request travel through three different emails before it gets approved, what if an employee could just open an app, tap a button, and submit it directly? Instead of HR digging through Excel files to figure out who took how many days off this year, what if the system could just tell them with one click? And what if all of this worked the same way whether someone was at their desk or out somewhere with only their phone?

That is what CoreHR does. It is a full Human Resource Management System that handles the entire employee lifecycle, from the moment a candidate applies for a job to the day someone finishes their offboarding process. It is built as three pieces that talk to each other but each have their own job.

The first piece is the web application. It is built with React.js and Vite, and it is mainly for HR administrators and managers. This is where the heavy work happens: adding new employees, approving leave requests, generating payroll reports, configuring permissions, and looking at charts and reports. The interface is responsive, supports both English and Arabic with proper right-to-left layout, and includes things like real-time notifications and a chatbot.

The second piece is the mobile application, built with Flutter. What makes the mobile app interesting is that it does not show the same thing to everyone. A regular employee opens the app and sees their own dashboard, with their attendance, their leave balance, and their goals. A manager opens the same app and sees something completely different, with team statistics, pending approvals, and organizational metrics. Each user only sees what makes sense for them, which keeps the interface clean and avoids overwhelming people with information they do not need.

The third piece is the backend API, built with Node.js and Express. This is the part of the system that no user ever sees directly, but it is doing most of the work behind the scenes. It handles login, checks permissions on every request, runs the business logic, and stores everything in a SQLite database. The backend exposes 161 endpoints, and every single one of them is protected by a

permission check, so even if someone tries to bypass the user interface and call the API directly, they still cannot access data they are not supposed to see.

Throughout this report, I will go through how CoreHR was built, the choices that were made along the way, the problems that came up, and what the final result looks like. The aim was never just to finish a graduation project. The aim was to build something that could actually be used.

Motivation

The reason for building CoreHR comes from looking at how most small and medium companies in our region actually manage their HR work. The truth is that in many places, including a lot of businesses in Palestine, HR is still being run on a mix of spreadsheets, WhatsApp messages, and paper forms that were never meant to work together.

Take a simple example. An employee wants to take a day off. They send an email to their manager. The manager forwards it to HR. HR opens a spreadsheet to check how many leave days the employee has left, then writes back to say it is approved, and updates the spreadsheet manually. By the time everyone has been informed, two or three days have already passed. The employee had no idea what was happening with their request until someone replied. The manager has no easy way to see who else from the team is also off that week. And HR has no clean record of who takes leave when, so they cannot really spot patterns or plan ahead.

This is not just one bad case. Attendance is often tracked with a paper sign-in sheet that anyone can fake. Payroll gets calculated manually in Excel, which is fine until you have to add allowances, deductions, and overtime, at which point mistakes become almost guaranteed. Performance reviews happen once a year if at all, because there is no good way to set goals and follow up on them through the year. Even something as basic as onboarding a new employee usually depends on whichever HR person is handling it that day, so different new hires end up with completely different experiences.

There are professional HR platforms that solve all of this. BambooHR, Workday, SAP SuccessFactors, and others. The problem is that they cost a lot. Monthly subscriptions for these tools are out of reach for most small companies, and they are mostly built for Western markets, so the interfaces are in English only and the workflows do not always match how things are done locally. The few Arabic options that exist tend to look outdated and miss important features.

CoreHR was built to fill that gap. It runs on lightweight infrastructure with SQLite, which means a company can deploy it on a single server without paying for database licenses or cloud subscriptions. It supports both English and Arabic from day one. And even though it is simple to deploy, it does not skip out on features. Twelve modules cover the main areas of HR work, 32 permissions give administrators fine control over access, and AI-powered features add a layer of intelligence that you usually only see in expensive platforms.

The other thing that drove this project is the realization that different people inside the same company need very different things from an HR system. An HR administrator wants full control and visibility over everything. A regular employee just wants to clock in, request leave, and see their salary. Building a system that gives both of them what they need without getting in their way is harder than it sounds, but it is also where most existing tools fall short.

Aims and Objectives

Aims

The main aim of CoreHR is to build a Human Resource Management System that is actually ready to be used in real organizations, not just demonstrated for a graduation defense. The system should digitize and automate the full employee lifecycle, give HR teams a single place to do their work, and stay affordable and easy to deploy. The target is small and medium organizations, somewhere between 10 and 500 employees, who do not have the budget for enterprise software but still need proper HR tools.

Objectives

Make Employee Management Simple:

Build a full employee information system where you can add, view, edit, and delete employee records, including personal details like name and date of birth, work details like department and position, and salary information.

Add search and filtering so managers can find an employee by name, department, or status without scrolling through a long list.

Give every employee a personal profile page where they can see their own information, attendance history, and leave balance.

Streamline Attendance and Leave:

Build a clock-in and clock-out system that works in real time, both from the web and from the mobile app, replacing paper sign-in sheets.

Create a full leave request workflow where employees submit requests with dates and reasons, managers approve or reject with one click, and the system notifies everyone involved automatically.

Add smart suggestions that look at the chosen leave dates and give employees useful tips, like recommending they start their leave on a Monday so the weekend before counts as part of the break.

Automate Payroll:

Build a payroll module that handles employee contracts, processes payslips through a draft to confirmed to paid lifecycle, and supports configurable allowances and deductions.

Generate payroll reports that show costs broken down by department, with summary statistics that help with financial planning.

Build a Strong Permission System:

Design and implement 32 individual permissions, organized into 7 categories, that cover every operation in the system.

Enforce these permissions in three places: in the backend API through middleware, in the web frontend through route guards and conditional rendering, and in the mobile app through dynamic UI that hides anything the user is not allowed to see.

Provide an interface where administrators can flip individual permissions for each user, or apply role-based defaults with one click.

Deliver a Cross-Platform Mobile Experience:

Build a Flutter mobile app that gives employees and managers different experiences, with each user seeing a dashboard and navigation tailored to what they actually do.

Make sure all the core HR operations, including clock in and out, leave requests, ticket creation, and goal tracking, can be done entirely from the phone.

Add Intelligent Features:

Build an AI Smart Insights engine that looks at attendance, leave, payroll, and ticket data and surfaces things worth paying attention to.

Build a chatbot that can answer employee questions about their leave balance, salary, attendance, company policies, and holidays, in both English and Arabic.

Support Both English and Arabic:

Make sure the entire web and mobile interface can be switched between English and Arabic, including proper right-to-left layout when Arabic is selected.

Constraints

The work on CoreHR started by picking the technology stack: React.js for the web, Flutter for mobile, Node.js with Express for the backend, and SQLite for the database. This combination was chosen because it gives a good mix of performance and flexibility while staying easy to deploy.

That said, working with this stack came with its own set of problems. The first big one was keeping things consistent between two completely different frontend frameworks. React uses JavaScript with JSX and a component-based approach with hooks. Flutter uses Dart with a widget-based system and Provider for state management. Every feature had to be thought through once but then implemented twice, in two different languages with two different ways of doing things, while making sure the user experience still felt the same on both sides.

One issue that kept coming up was making sure field names matched between the Flutter app and the backend. The backend would return data with names like `employee_name`, `is_active`, and `check_in`, but the mobile screens were sometimes written assuming `first_name` and `last_name` as separate fields, or `status` instead of `is_active`, or `clock_in` instead of `check_in`. These small mismatches caused data not to display, forms not to save, and features to look broken even when the underlying logic was fine. Fixing every one of these required going through every endpoint and every screen carefully.

Flutter on web also caused some visual headaches that took a while to figure out. Modal dialogs that were supposed to have transparent backgrounds left gray overlays on the screen. `TabBarView` widgets sometimes rendered content from other tabs behind the active tab, leaving gray patches. Even the default scaffold background color had a slight gray tint instead of being pure white, which created small inconsistencies between screens. Each of these had to be tracked down separately and worked around.

SQLite was great for keeping the deployment simple, but it has limits when it comes to handling many people writing to the database at the same time. For a small or medium company this is fine, but if you wanted to deploy CoreHR for a much larger organization with hundreds of users hitting the system at once, you would need to switch to PostgreSQL or MySQL. SQLite also lacks some of the more advanced SQL features that would have made certain queries shorter, so some of that complexity ended up in the application code instead.

Security was something I had to keep thinking about throughout development. Every API route needed to check permissions. Every SQL query had to use parameterized statements so nobody

could inject malicious input. Every piece of data going to the user had to be filtered based on who was asking for it. One missed permission check could leak salary data. One unparameterized query could open the door to attacks. Auditing all 161 endpoints to make sure they were all protected was tedious but necessary.

Despite all of this, every problem ended up teaching me something. The field name mismatches taught me to be more systematic about checking my work. The Flutter rendering issues taught me how web rendering pipelines actually work. The permission system pushed me to design something flexible enough to handle real authorization scenarios. The constraints made the project harder, but they also made it better.

Literature Review

Human Resource Management Systems have come a long way over the past few decades. What started as basic computerized record-keeping has turned into platforms that touch almost every part of the employee experience. To understand where CoreHR fits, it helps to look at how this evolution happened and where things stand today.

The earliest HR software, going back to the 1990s, was mostly about automating payroll and keeping basic employee databases. These systems replaced paper ledgers with digital ones, but they did not really change how HR worked. They were installed on company servers, needed dedicated IT staff to keep running, and could only be accessed from specific computers in the office. They cut down on calculation mistakes, but they did not transform anything.

In the 2000s, a second generation of HRMS platforms appeared. Companies like SAP with its SuccessFactors module and Oracle with PeopleSoft started offering integrated suites that tried to cover everything from hiring to retirement. These were more powerful, but they were expensive, often costing hundreds of thousands of dollars to implement. As a result, only large corporations with serious HR technology budgets could actually use them.

The cloud computing wave of the 2010s changed everything. Software-as-a-Service HR platforms, like BambooHR, Gusto, Namely, and Zenefits, made HR technology affordable for smaller companies through monthly subscriptions. They had modern web interfaces, no on-premises infrastructure to maintain, and they updated themselves automatically. Reports from analyst firms like Sierra-Cedar showed that cloud HR adoption kept growing, with small and medium businesses leading the shift.

Research published by Stone, Deadrick, Lukaszewski, and Johnson in the Human Resource Management Review in 2015 pointed out something important: employees actually prefer being able to handle their own HR tasks themselves. When companies gave employees self-service portals, satisfaction went up, data accuracy improved, and HR teams had more time to focus on strategic work. CoreHR is built around this idea. Employees get a personal dashboard on their phone where they can manage their leave, see their attendance, check their salary, and track their goals without having to go through anyone.

The role-based access control model, first defined formally by Sandhu, Coyne, Feinstein, and Youman in their well-known 1996 paper in IEEE Computer, laid the groundwork for managing user permissions in enterprise systems. The traditional approach gives users a role like

administrator or manager or employee, with each role having a fixed set of permissions. This works for simple cases, but it can be too rigid when individuals need custom mixes of permissions. CoreHR takes the standard role-based approach and extends it: there are 32 individual permissions that an administrator can flip on or off for any user, independently of their role. This way you get the simplicity of role defaults with the flexibility to customize when needed.

Mobile is also a big factor in modern HR. A Deloitte report on Global Human Capital Trends noted that employees, especially younger ones, expect to use their phones to interact with workplace systems. A study in HR Magazine specifically looked at mobile HR applications and found that companies offering mobile access saw better employee engagement and faster response times. CoreHR was designed with this in mind. The Flutter mobile app is not just a shrunken version of the web interface, it is built specifically for mobile use, with role-specific dashboards that focus each user on what they actually need.

AI in HR is still a newer area, but the major platforms are starting to explore it. Common uses include screening resumes, analyzing employee feedback, and predicting which employees might leave. Full AI integration usually needs serious computing power, but CoreHR shows that you can add useful AI features without going that far. The smart insights look at patterns across all the modules. The chatbot answers common HR questions instantly. The leave suggestions help employees pick better dates. None of this requires expensive cloud AI services, but all of it adds real value.

In the end, CoreHR sits at the intersection of several trends in HR technology: cloud-style accessibility, employee self-service, fine-grained permissions, mobile-first design, and intelligent features. It builds on what decades of research and industry experience have shown works, while addressing the specific needs of organizations in regions where the big commercial platforms are not really an option.

Methodology

Non-Functional Requirements

Screen Adaptation:

The web application is built with responsive CSS that adjusts the layout based on the screen size, all the way from large desktop monitors down to tablet displays. The sidebar collapses on smaller screens, tables scroll horizontally when needed, and modal dialogs resize to fit the viewport. The mobile app, since it is built with Flutter, handles screen adaptation natively through its widget system, so it works well on small phones and on larger tablets without any special configuration. Both platforms share the same color schemes and design patterns, so the system feels consistent no matter where you are using it from.

Security:

Security is something that had to be addressed at multiple levels. User passwords are hashed using the bcryptjs library with automatic salting, so even if someone got their hands on the database, they could not easily reverse the hashes. Login tokens are JSON Web Tokens signed with HMAC-SHA256, set to expire after seven days. Every API route that touches sensitive data is protected by a middleware called checkPerm, which verifies that the user is logged in and that they have the specific permission required for that operation. All database queries use parameterized statements through better-sqlite3, which means SQL injection attacks are not possible. HTTP security headers are added through Helmet, and CORS is configured so that only the official frontend can call the API.

Ease of Use:

The interface is designed to be approachable for people who are not technical. The sidebar on the web and the bottom navigation on mobile only show the features the user has access to, so nobody is presented with options they cannot use. Forms have clear labels, sensible input types, and immediate validation feedback when something is wrong. The mobile app uses bottom sheet edit dialogs with fields already filled in, one-tap quick actions, and swipe gestures for common operations. When errors happen, the messages explain what went wrong and what to do about it.

Usability:

Each page in the system clearly says what it is for through descriptive titles, helpful subtitles, and empty-state messages that guide the user when there is no data yet. Charts and visualizations use consistent colors that match what they mean across the whole system: green for positive, orange for warning, red for serious problems. The permission-based interface means each user sees a system that feels tailored to their role, instead of one cluttered with options that do not apply to them.

Design

Good design planning matters a lot in any software project. The interface is what users actually see and interact with, so it has a big effect on whether they find the system easy or frustrating. CoreHR uses a clean, simple visual language across both the web and mobile platforms, built around a green color palette that feels calm and professional, with accent colors used carefully to indicate status and actions.

The system architecture keeps the different concerns clearly separated. The backend handles all the business logic and data validation. The web frontend provides the administrator and manager experience. The mobile app delivers the on-the-go experience for employees and managers in the field. All three communicate through JSON-based REST API calls, so any one of them could in theory be replaced or upgraded without affecting the others.

Authentication and Authorization

CoreHR has an authentication and authorization system that goes beyond just letting people log in. When a user enters their credentials, the system checks them against bcrypt-hashed passwords stored in the database. If they match, the server creates a JWT token containing the user's ID and sends it back along with the user's full profile, including their role and all 32 permission flags.

The frontend stores this token and attaches it to every API request that follows. The backend checks the token on every request and re-loads the user's permissions from the database, which means that if an administrator changes someone's permissions, the change takes effect on their very next API call. The mobile app also re-fetches permissions every time it starts, so a stale cache cannot accidentally let someone access something they should not.

Dashboard

The dashboard is the first thing users see after logging in, so it is designed to give them an immediate picture of what matters most based on who they are. CoreHR actually has two completely different dashboards.

For regular employees, the dashboard shows a personal clock-in card that tells them whether they have clocked in today and what their times are, quick action buttons for things like requesting leave or opening a support ticket, a summary of their recent leave requests with color-coded statuses, their active performance goals with progress bars, and a small motivational card showing how many goals they have completed.

For managers and administrators, the dashboard shows organization-wide statistics: total employee count, today's attendance numbers, leave numbers, and open job positions. Below that are quick action buttons that go straight to attendance, leave management, helpdesk, and reports. There are stat cards for pending leave requests, active candidates, open tickets, and active projects, each linked to its own page. An attendance trend line chart shows the past seven days, a team distribution section shows department sizes with progress bars, upcoming birthdays for the current month are listed, and a recent activity feed shows what is happening in the system.

Every element on both dashboards is tappable and goes to the relevant detail page. Everything is also conditionally rendered based on permissions, so a manager who does not have recruitment access will not even see the candidates card.

Employee Management

The employee management module is the heart of the system. It provides full create, read, update, and delete operations for handling the workforce. The employee list page shows everyone in a searchable, filterable table on the web, and a scrollable list on mobile, displaying each person's name, badge ID, department, position, work type, joining date, salary, and active status.

Adding a new employee means filling out a form that collects personal information like first name, last name, email, phone, date of birth, and gender, along with work information like department, position, work type, shift, joining date, salary, and status. The system creates both the user account for login and the employee record linked to it, all in one go.

On the web, Add, Edit, and Delete buttons only show up for users who have the matching permissions. The mobile app does the same thing through its edit dialogs and delete confirmations.

Attendance Management

The attendance system gives employees a real-time clock-in and clock-out function from both the web and the mobile app. When someone clocks in, the system records the current time. When they clock out, it calculates the total worked hours automatically. The system also prevents duplicate clock-ins on the same day and requires a clock-in before clock-out can happen.

Managers can see a summary of the day's attendance, including total employees, how many are present, how many are absent, and how many are on leave. They can also browse historical attendance with filters for employee, date range, and status. The mobile app shows attendance history as cards displaying check-in time, check-out time, and worked hours for each day.

One critical security detail here is data isolation. Regular employees can only see their own attendance through the API, while managers and administrators can see everyone. This is enforced at the backend level, which means an employee cannot get to another employee's attendance even if they try to call the API directly.

Leave Management

CoreHR supports five configurable leave types: Annual Leave with 20 days, Sick Leave with 10 days, Casual Leave with 7 days, Maternity Leave with 90 days, and Unpaid Leave with 30 days. Each type can be marked as paid or unpaid, and you can decide whether unused days carry over to the next year.

When an employee submits a leave request, they pick the type, the start and end dates, and optionally a reason. The system calculates the number of days automatically and validates the request. The smart suggestion feature looks at the dates the employee picked and gives helpful tips. For example, if the leave starts on a Monday, it suggests combining it with the weekend before for a longer break.

As soon as the request is submitted, the system sends a notification to all managers and administrators. They can approve or reject the request with one click, and either decision triggers a notification back to the employee. The status of each request goes through a lifecycle: pending, approved, or rejected, each shown in a different color.

Like attendance, leave data is isolated by permission. Employees see only their own requests, while managers see all of them.

Payroll Management

The payroll module handles the full compensation cycle. It manages employee contracts, which specify the contract type (permanent, contract, part-time, or freelance), start and end dates, base wage, and pay frequency. Contracts can be edited through a bottom sheet interface on mobile, so managers can update terms or status without leaving the screen.

Payslips are generated with basic pay, allowances, deductions, and a calculated net pay. Each one moves through a lifecycle from draft, to confirmed, to paid, with managers advancing them through each stage. The system maintains configurable lists of allowances like housing, transport, and meal, and deductions like tax, insurance, and social security, each set up as either a fixed amount or a percentage.

Reports give summary statistics, including total payroll, average salary, total payslips, and breakdowns by department, displayed with interactive charts.

Recruitment Pipeline

The recruitment module covers the full hiring process from posting a job to hiring a candidate. Job postings can be created with titles, descriptions, vacancy counts, and an open or closed status. The pipeline itself has customizable stages: Applied, Screening, Interview, Technical Test, Offer, and Hired.

Candidates are tracked through these stages with their name, email, phone, and current stage. Managers can move candidates between stages and eventually mark them as hired, which sends a notification to all administrators. The mobile app allows editing candidate details by tapping list items and deleting them through long-press gestures.

Onboarding and Offboarding

CoreHR provides structured task management for both onboarding new employees and offboarding people who are leaving. Administrators can create template tasks like "Complete personal information form," "Set up company email," "Review company policies," and "Meet with IT for equipment setup." Each task can be marked as required or optional.

When a new employee joins, all the template tasks get assigned to them automatically, giving them a personal checklist. The onboarding coordinator can then track which tasks have been done and which are still pending. The same flow works for offboarding, with tasks like "Return company assets," "Complete knowledge transfer," "Conduct exit interview," and "Revoke system access."

Performance Management

The performance module is built around continuous goal tracking and feedback rather than the once-a-year review approach. Goals are created with a title, description, target date, and progress percentage. Managers can update progress with quick buttons that set it to 25, 50, or 100 percent, or through a detailed edit dialog. The progress bars use color coding, with red for goals below 30 percent, orange between 30 and 80, and green above 80.

There is also a 360-degree feedback system that lets managers give structured feedback to employees, with star ratings, comments, and a period classification. Both goals and feedback show up on the employee dashboard and feed into the organizational reports.

Helpdesk and Ticketing

The helpdesk module is a structured support ticketing system. Tickets are created with a title, description, category (IT Support, HR Queries, or Facilities), and priority level (low, medium, high, or urgent). Each ticket follows a lifecycle: open, in progress, resolved, or closed.

When an employee creates a ticket, the system assigns it to them and sends a notification to all managers. Managers can update the status and priority from the web or the mobile app. Tickets can be filtered by status and priority, and the dashboard shows a count of open tickets so managers always know what is waiting for them.

Data isolation applies here too. Regular employees only see their own tickets, while managers see everything across the organization.

Asset Management

The asset management module tracks company-owned equipment and who has it. Assets are organized into categories like laptops, phones, and monitors, each with a name, tag or serial number, purchase date, cost, and status. The status can be available, allocated, in maintenance, or retired.

Assets can be allocated to employees and later returned, with the system keeping track of the full allocation history including dates and notes. The mobile app shows this through a clean tabbed interface with separate views for assets, allocations, and categories, each with edit and delete capabilities for users who have permission.

Project Management

The project management module is intentionally simple but functional. It handles projects with name, description, start and end dates, and status (active, on hold, completed, or cancelled). Each project contains tasks that can be assigned to employees with priority levels and due dates.

Tasks move through a workflow from todo to in progress to done, with managers updating the status as they go. There is also a timesheet feature that lets employees log hours worked on specific tasks, which feeds into project cost analysis and reporting. The mobile app supports creating projects, adding tasks to existing projects, and logging time, all through bottom sheet dialogs.

Reports and Analytics

CoreHR provides five report types, each combining interactive charts with detailed data tables:

Headcount Report: Shows the employee distribution across departments, with a stacked bar chart comparing male and female counts, a doughnut chart for gender distribution, and a detailed table with percentages.

Attendance Report: Shows attendance statistics per employee for a chosen month and year, with a horizontal bar chart for days present and a line chart for average hours per employee.

Leave Report: Shows leave usage with a doughnut chart breaking down approved, rejected, and pending requests, and a bar chart showing approved days by leave type.

Payroll Report: Provides a financial overview with summary stat cards for total basic pay, allowances, deductions, and net pay, plus a bar chart by department, a doughnut chart for pay components, and a detailed employee-level table.

Recruitment Report: Shows the recruitment funnel with a doughnut chart of hired, in-pipeline, and rejected candidates, summary stat cards, and a detailed table with hire rates per position.

Settings and Configuration

The settings module gives administrators the tools to set up the organizational structure. This includes managing companies with their contact info and addresses, departments that group employees by function, job positions that define roles within departments, shifts that specify working hours, and holidays for company-wide days off.

Each settings entity supports full CRUD with the appropriate permission checks. The web interface organizes them into clear sections with search, while the mobile app provides list views with edit and delete capabilities.

AI Features

CoreHR includes three different AI-powered features that add a layer of intelligence on top of regular HR work:

Smart Insights look at data across all the HR modules and generate context-aware recommendations that show up on the web dashboard. The engine looks at attendance rates and trends, picks out employees absent for three or more days in a row, counts pending leave requests, checks for contracts that are about to expire, watches for draft payslips that need processing, tracks urgent support tickets, finds underperforming goals, and summarizes recruitment activity. Each insight is classified by severity, success, info, warning, or danger, and is only shown to users who have permission to view that module.

The HR Chatbot is a conversational interface accessed through a floating button on the web app. Employees can ask questions in plain English or Arabic, and the chatbot responds with data pulled directly from the database. It can answer questions about leave balance, salary details, attendance records, performance goals, support tickets, company policies, shift schedules, holidays, department information, and personal profile details.

Smart Leave Suggestions kick in when an employee fills out a leave request form. The system looks at the chosen start date and gives contextual tips. If the leave starts on Monday, it suggests using the weekend before for a longer break. If it starts on Thursday, it recommends Friday instead for a three-day weekend. If the leave is longer than five days, it suggests splitting it for better team coverage. If the start date falls on a weekend, it reminds the employee to pick a workday.

Notification System

CoreHR has a notification system that automatically generates alerts when important things happen. When an employee submits a leave request, all managers get notified. When a manager approves or rejects a leave request, the employee who requested it gets notified. When a new support ticket is created, managers are notified. When a new employee is added, administrators are notified. When a candidate is hired through the recruitment module, administrators are notified.

Notifications show up as a bell icon in the web navbar with an unread count badge, and on the mobile app through a dedicated notifications screen with an unread badge on the More tab in the bottom navigation. Each notification shows its title, message, and creation time, with unread ones highlighted. Users can mark individual notifications as read, mark all as read, or delete the ones they no longer want.

Implementation

Key Libraries - Server Side

Node.js and Express:

For the backend, we picked Node.js as the runtime and Express as the framework. Node.js performs well for I/O-heavy work like API servers because of its non-blocking event loop, which lets it handle a lot of concurrent requests without slowing down. Express is minimal but flexible, and its middleware approach makes it easy to add things like authentication, permission checking, CORS, and error handling as separate composable layers. The backend is organized into 19 route files with 161 endpoints, all following REST conventions.

SQLite with better-sqlite3:

For data storage, we went with SQLite through the better-sqlite3 driver, which gives synchronous access with very good performance. Unlike client-server databases like PostgreSQL or MySQL, SQLite stores everything in a single file, with no configuration and no separate database server. This makes deploying CoreHR almost trivial. The database is in Write-Ahead Logging mode for better concurrent reads, and foreign key constraints are enforced for data integrity. The schema has 35 tables with carefully designed relationships and CHECK constraints for validation.

Security Libraries:

Authentication uses jsonwebtoken for generating and verifying JWT tokens, and bcryptjs for hashing passwords with automatic salting. Helmet adds security-related HTTP headers, including Content-Security-Policy, X-Frame-Options, and X-Content-Type-Options. The cors middleware restricts API access to authorized frontend origins so cross-site request attacks are blocked.

Key Libraries - Client Side

React.js with Vite (Web):

The web frontend is built with React 18 using the Vite build tool, which gives near-instant hot module replacement during development and produces optimized production builds. The application uses React Router v6 for client-side navigation with protected routes that enforce permission checks. Axios handles HTTP communication, with request interceptors that automatically add the JWT token. Chart.js paired with react-chartjs-2 powers all the interactive charts on the dashboard and reports pages, supporting bar, line, doughnut, and polar area chart types.

Flutter with Dart (Mobile):

The mobile application is built with Flutter 3.x using Dart. State management uses the Provider pattern for things like authentication state and notification counts, with local state through StatefulWidgets for screen-specific data. Navigation uses GoRouter with StatefulShellRoute for the bottom navigation structure, supporting deep linking and route guards. The Dio HTTP client handles API communication with interceptors for automatic JWT token injection. fl_chart provides native Flutter charts for the dashboard attendance trend and department distribution. SharedPreferences stores authentication tokens and cached user data so the app starts faster and works offline to some degree.

Tools

Visual Studio Code:

VS Code was the main development environment for both the web frontend and the backend. Its extension ecosystem has great support for JavaScript, React, Node.js, and SQLite. The integrated terminal made it possible to run dev servers right inside the editor, and the built-in Git integration

handled version control. Extensions like ESLint, Prettier, and Thunder Client kept code quality consistent and made API testing simple throughout the project.

Android Studio:

Android Studio was used for the Flutter mobile development side. It provides the Android SDK, emulator management, and Flutter-specific tooling. The Flutter inspector and widget inspector were really helpful for debugging layout issues, especially the persistent gray background problem that took a while to track down because it required understanding how Flutter's web rendering pipeline works. The emulator made it easy to test on different screen sizes and Android versions.

Chrome DevTools:

The browser developer tools were essential for debugging the web app. Inspecting network requests, watching API responses, and testing responsive layouts all happened here. The Application tab was particularly useful for checking localStorage to make sure user sessions and permissions were being stored correctly.

Postman and Thunder Client:

API testing was done with both Postman and the Thunder Client extension inside VS Code. Every endpoint was tested individually to make sure it behaved correctly for both authorized and unauthorized requests, returned proper error responses for invalid input, and isolated data correctly between different user roles. A test script ended up automating 66 critical API operations.

Results

CoreHR was successfully built, tested, and deployed as a complete Human Resource Management platform. The sections below go through the results across the web dashboard, the mobile employee experience, and the mobile administrator experience, along with system-wide statistics that show the overall scope of the project.

System-Wide Statistics:

- Total API Endpoints: 161 across 19 route files
- Automated Test Results: 66 out of 66 tests passing (100% success rate)
- Database Tables: 35 tables with enforced foreign key constraints
- Permission System: 32 granular permissions across 7 categories, enforced at 3 levels
- Web Pages: 18 page components with full CRUD operations
- Mobile Screens: 18 screens with role-specific UI
- Charts and Visualizations: 14+ interactive charts across dashboard and reports
- Language Support: Full English and Arabic with RTL layout
- Notification Events: 5 automatic notification triggers
- AI Features: Smart Insights, HR Chatbot, Smart Leave Suggestions

Test Account Credentials:

Role	Username	Password	Permissions
Super Admin	admin	admin123	Full access to all 32 permissions
Manager	emma.williams	password123	22 permissions (no payroll manage, settings manage)
Employee	michael.chen		9 basic permissions (attendance, leave, helpdesk)

Results - Web Dashboard

Login Page

Figure 1: Login Page

C **CoreHR**
HR Management System

Welcome back

Sign in to your CoreHR account to continue.

Username or Email

Password

Sign In

[Forgot password?](#)

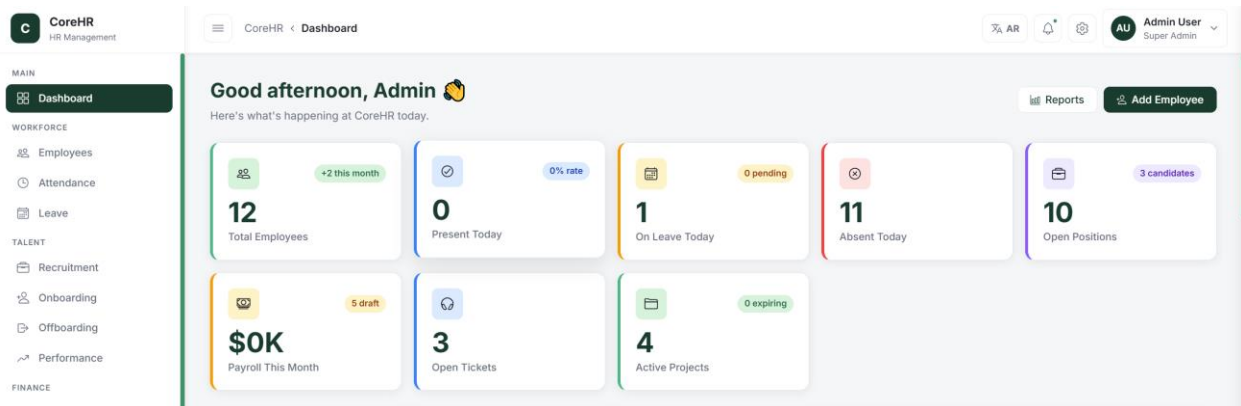
Demo: [admin / admin123](#)

Employee: [sarah.johnson / password123](#)

The login page is the entry point for the web application. Users enter their email or username and password, and on successful authentication they are redirected to the dashboard. If the credentials are wrong, a clear error message appears. There is also a language toggle that lets users switch between English and Arabic before logging in, with the layout flipping to right-to-left for Arabic users.

Admin Dashboard

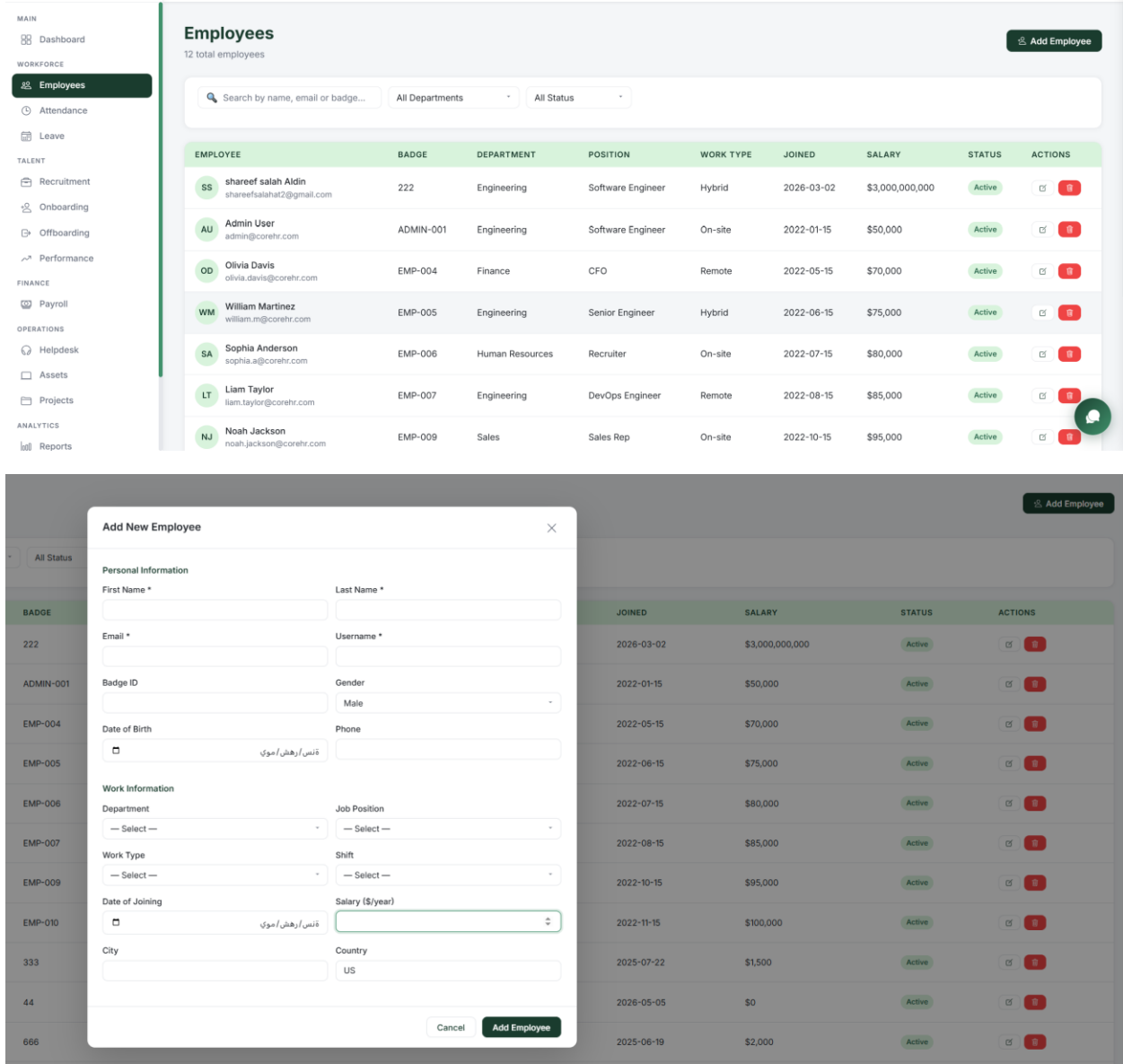
Figure 2: Admin Dashboard with Statistics and Charts



The admin dashboard is the first thing managers and administrators see after logging in. It shows organization-wide statistics at the top, including total employees, today's attendance, employees on leave, and open positions. Below that are quick action buttons that link to the most common tasks. Stat cards display pending leave requests, active candidates, open tickets, and active projects, each tappable to go straight to its detail page. An attendance trend chart covers the last seven days, and a team distribution section shows how employees are spread across departments.

Employee List Page

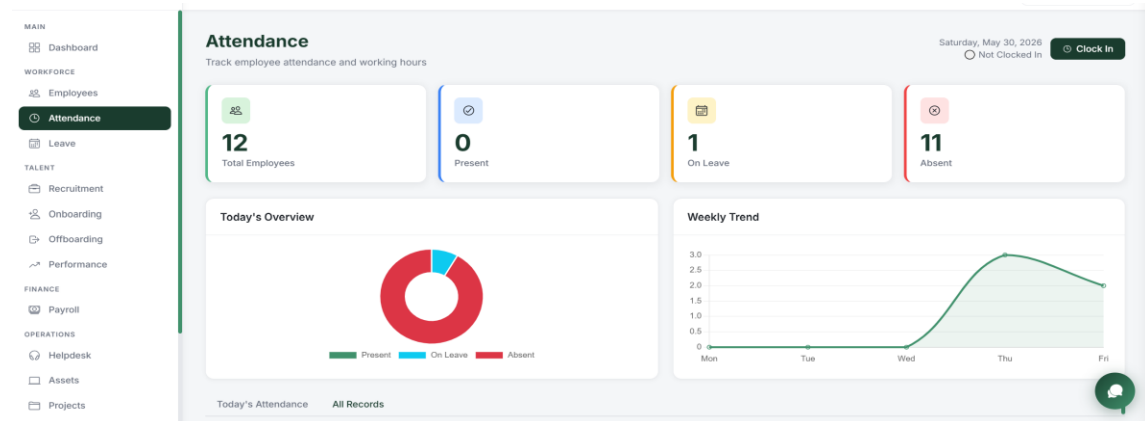
Figure 3: Employee List with CRUD Operations



The employee list page displays everyone in the organization in a searchable, filterable table. Each row shows the employee's name, badge ID, department, position, work type, joining date, salary, and active status. Add, Edit, and Delete buttons appear only for users who have the matching permissions. The search bar at the top filters in real time as the user types, and the page supports pagination for organizations with many employees.

Attendance Page

Figure 4: Attendance Page with Clock In/Out



Today's Attendance All Records

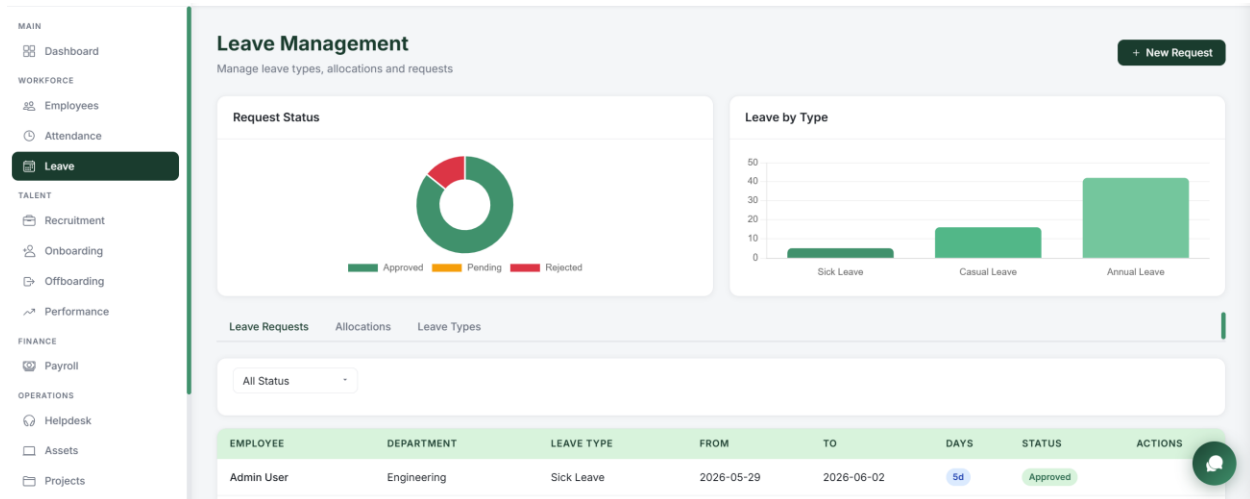
All Employees 0 نس / رهش / موي to 0 نس / رهش / موي

EMPLOYEE	DATE	CHECK IN	CHECK OUT	HOURS	STATUS
Admin User	2026-05-29	10:59	11:00	0.0h	Present
Admin User	2026-05-19	13:21	—	—	Present
Sophia Anderson	2026-05-19	13:20	—	—	Present
Admin User	2026-05-17	12:56	12:56	—	Present
Admin User	2026-05-16	18:38	18:38	—	Present
shareef salah Aldin	2026-05-16	18:12	—	—	Present
Sophia Anderson	2026-05-16	18:07	18:07	—	Present
Admin User	2026-05-02	12:02	—	—	Present
Admin User	2026-05-01	14:26	—	—	Present

The attendance page provides a clock-in and clock-out button at the top for the current user, and a table of historical records below. The summary cards show present, absent, and on-leave counts for the day. Managers can filter the records by employee, date range, and status.

Leave Management

Figure 5: Leave Management with Approval Workflow



New Leave Request

Employee: Admin User

Leave Type *: — Select —

Start Date *:

End Date *:

Reason:

Leave Requests Allocations Leave Types

All Status

EMPLOYEE	DEPARTMENT	LEAVE TYPE	FROM	TO	DAYS	STATUS	ACTIONS
Admin User	Engineering	Sick Leave	2026-05-29	2026-06-02	5d	Approved	
Sophia Anderson	Human Resources	Casual Leave	2026-05-20	2026-05-23	4d	Approved	
Admin User	Engineering	Casual Leave	2026-05-20	2026-05-22	3d	Approved	
Admin User	Engineering	Maternity Leave	2026-05-17	2026-05-18	1d	Rejected	
Sophia Anderson	Human Resources	Casual Leave	2026-05-16	2026-05-24	9d	Approved	
Admin User	Engineering	Annual Leave	2026-04-06	2026-05-16	41d	Approved	
Admin User	Engineering	Annual Leave	2026-05-02	2026-05-03	1d	Approved	

Leave Requests Allocations Leave Types

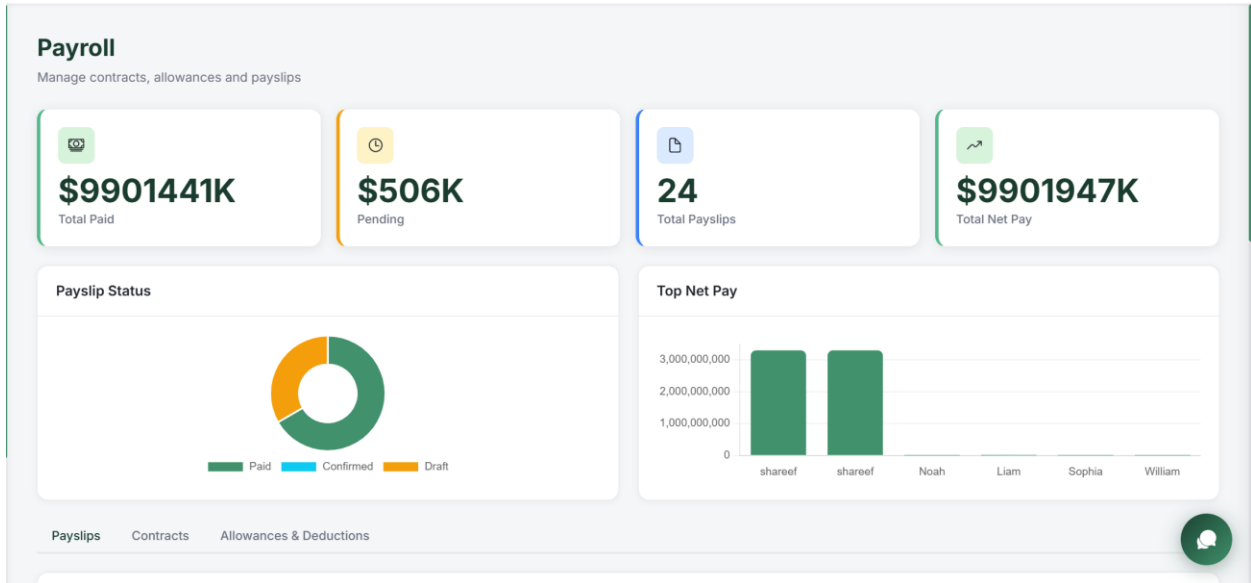
Leave Types 5 + Add Leave Type

LEAVE TYPE	TOTAL DAYS	CARRY FORWARD	PAID	ACTIONS
Annual Leave	20	Yes	Paid	
Casual Leave	7	No	Paid	
Maternity Leave	90	No	Paid	
Sick Leave	10	No	Paid	
Unpaid Leave	30	No	Unpaid	

The leave management page lists all leave requests with their status, requester, dates, type, and reason. Managers can approve or reject pending requests with a single click. The smart leave suggestion feature appears when creating a new request, giving the user contextual tips about their chosen dates.

Payroll Management

Figure 6: Payroll with Payslips and Contracts



Navigation: Payslips | Contracts | Allowances & Deductions

All Status

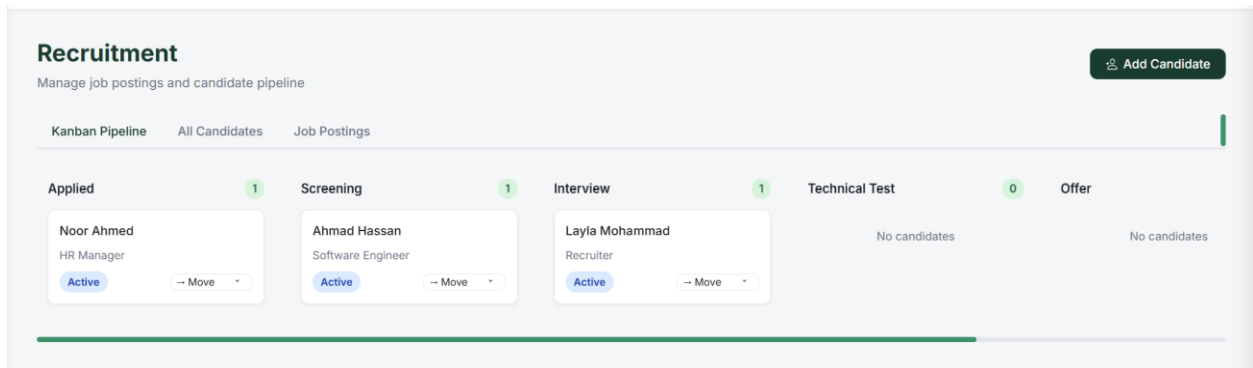
EMPLOYEE	DEPARTMENT	PERIOD	BASIC PAY	ALLOWANCES	DEDUCTIONS	NET PAY	STATUS	ACTIONS
shareef salah Aldin	Engineering	2026-04	\$3,000,000,000	+\$450,000,000	-\$150,000,000	\$3,300,000,000	Paid	
Admin User	Engineering	2026-04	\$55,000	+\$8,250	-\$2,750	\$60,500	Paid	
Olivia Davis	Finance	2026-04	\$75,000	+\$11,250	-\$3,750	\$82,500	Paid	
William Martinez	Engineering	2026-04	\$80,000	+\$12,000	-\$4,000	\$88,000	Draft	Confirm
Sophia Anderson	Human Resources	2026-04	\$85,000	+\$12,750	-\$4,250	\$93,500	Draft	Confirm
Liam Taylor	Engineering	2026-04	\$90,000	+\$13,500	-\$4,500	\$99,000	Draft	Confirm
Noah Jackson	Sales	2026-04	\$100,000	+\$15,000	-\$5,000	\$110,000	Draft	Confirm
Isabella White	Finance	2026-04	\$105,000	+\$15,750	-\$5,250	\$115,500	Draft	Confirm
shareef salah Aldin	Engineering	2026-03	\$3,000,000,000	+\$450,000,000	-\$150,000,000	\$3,300,000,000	Paid	

Allowances			Deductions		
NAME	AMOUNT	TYPE	NAME	AMOUNT	TYPE
Housing	\$500	Fixed	Insurance	\$200	Fixed
Meal	\$150	Fixed	Social Security	5%	Percentage
Performance Bonus	10%	Percentage	Tax	10%	Percentage
Transport	\$200	Fixed			

The payroll page is split into sections for contracts, payslips, allowances, and deductions. Payslips can be advanced through their lifecycle from draft to confirmed to paid. Contracts can be created and edited, with their type, dates, and base wage clearly displayed.

Recruitment Pipeline

Figure 7: Recruitment Pipeline View



Add Candidate ✕

Full Name *

Email Phone

Job Posting

— Select —

Stage

— Select —

Notes

Ahmad Abu Shams	aabushams80@gmail.com	Leadership TeamWork Project Management Communication Skills Node JS JavaScript Data-Structure Node Js Visual Studio code Ts.Ed TypeScript Mngodb Postman Azure Devops React Node Java Flutter MySQL Firebase SQL	Recruiter	Offer	Hired	2026-05-15	<input type="button" value="🗑️"/> <input type="button" value="🔍"/> <input type="button" value="🔄"/>
Ahmad Hassan	ahmad.hassan@email.com	React JavaScript Java HTML CSS Bootstrap Git REST API	Software Engineer	Screening	Active	2026-04-11	<input type="button" value="🗑️"/> <input type="button" value="🔍"/> <input type="button" value="🔄"/>
Sara Ali	sara.ali@email.com	React JavaScript Java HTML CSS Bootstrap Git REST API	Senior Engineer	Offer	Hired	2026-04-11	<input type="button" value="🗑️"/> <input type="button" value="🔍"/> <input type="button" value="🔄"/>

The recruitment page shows all candidates organized by their pipeline stage: Applied, Screening, Interview, Technical Test, Offer, and Hired. Candidates can be moved between stages with drag-and-drop or through the edit dialog. Job postings are managed in a separate section.

Onboarding

Figure 8: Onboarding

The screenshot displays the onboarding interface for two candidates. The top section shows the progress for 'Abed yahya' (1/1 tasks completed, 100%) and 'Ahmad Abu Shams' (0/5 tasks completed, 0%). Below these are detailed task lists for each candidate. The bottom section shows a 'Task Templates' table for new hire onboarding tasks.

Abed yahya (1/1 tasks completed, 100%)

TASK	REQUIRED	STATUS	COMPLETED	ACTION
task2	Optional	Completed	2026-05-29	

Ahmad Abu Shams (0/5 tasks completed, 0%)

TASK	REQUIRED	STATUS	COMPLETED	ACTION
task2	Optional	pending	—	✓ Complete
task1	Optional	pending	—	✓ Complete
Create Employee Account	Optional	pending	—	✓ Complete
Assign Company Email	Optional	pending	—	✓ Complete
Allocate Laptop	Optional	pending	—	✓ Complete

Onboarding
New hire onboarding tasks

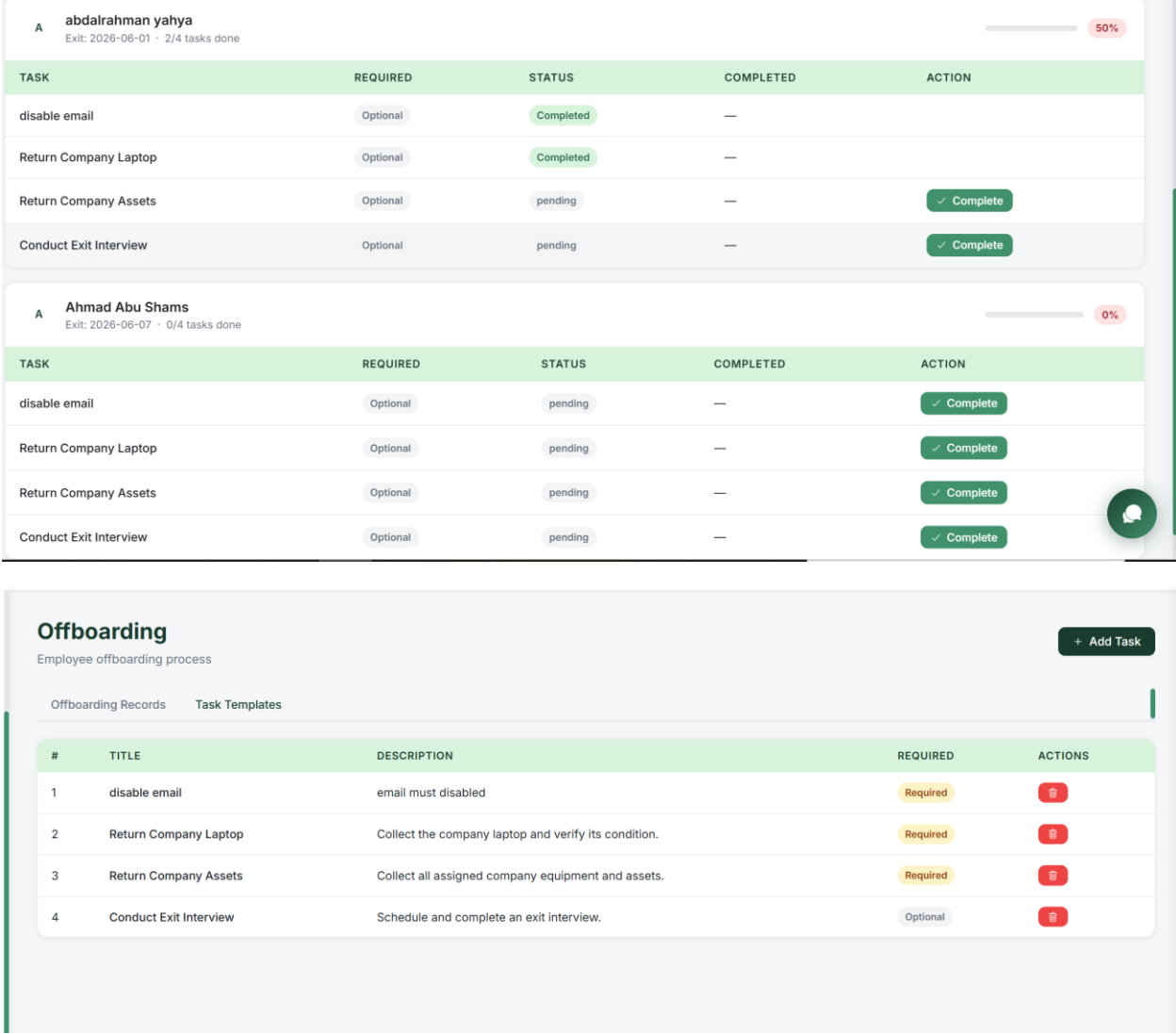
Onboarding Records | Task Templates

#	TITLE	DESCRIPTION	REQUIRED	ACTIONS
1	task2	bbb	Required	🗑️
2	task1	task1 Emp	Required	🗑️
3	Create Employee Account	Create the employee's system account and login credentials.	Required	🗑️
4	Assign Company Email	Set up and activate the employee's company email address.	Required	🗑️
5	Allocate Laptop	Assign a company laptop and record asset details.	Required	🗑️

The Onboarding module manages all activities related to welcoming and preparing new employees. It allows HR teams to assign onboarding tasks such as account setup, equipment allocation, document completion, and orientation activities. This ensures a smooth and organized transition for new hires into the company.

Offboarding

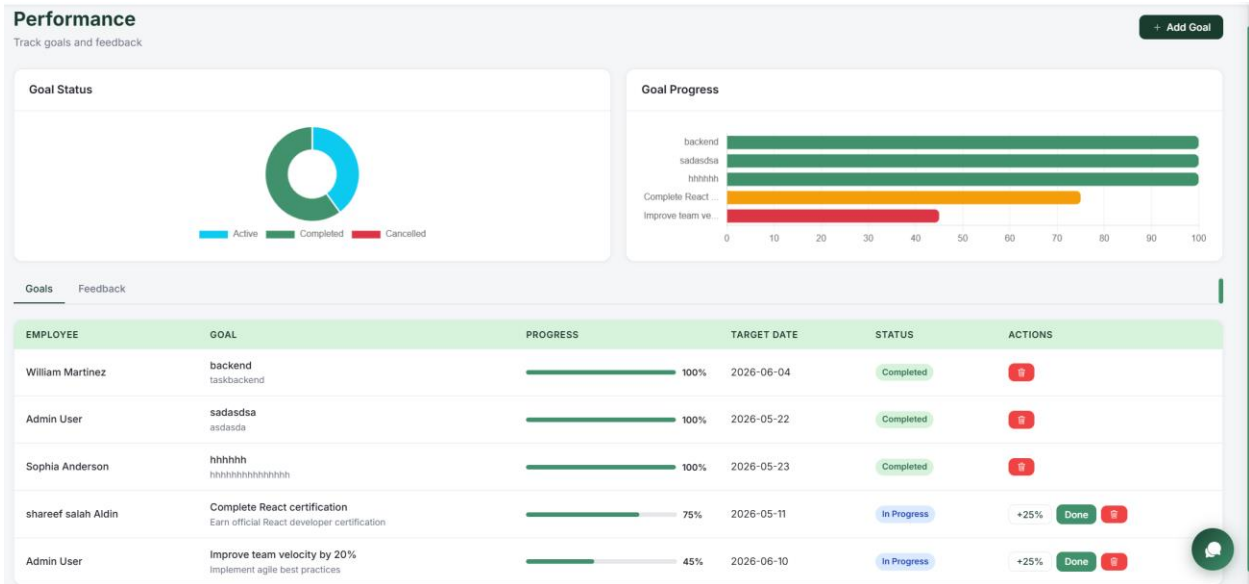
Figure 9: Offboarding



The Offboarding module manages the process of employees leaving the organization. It helps track required tasks such as returning company assets, revoking system access, and completing final documentation. This ensures that employee departures are handled efficiently while maintaining organizational security and compliance.

Performance

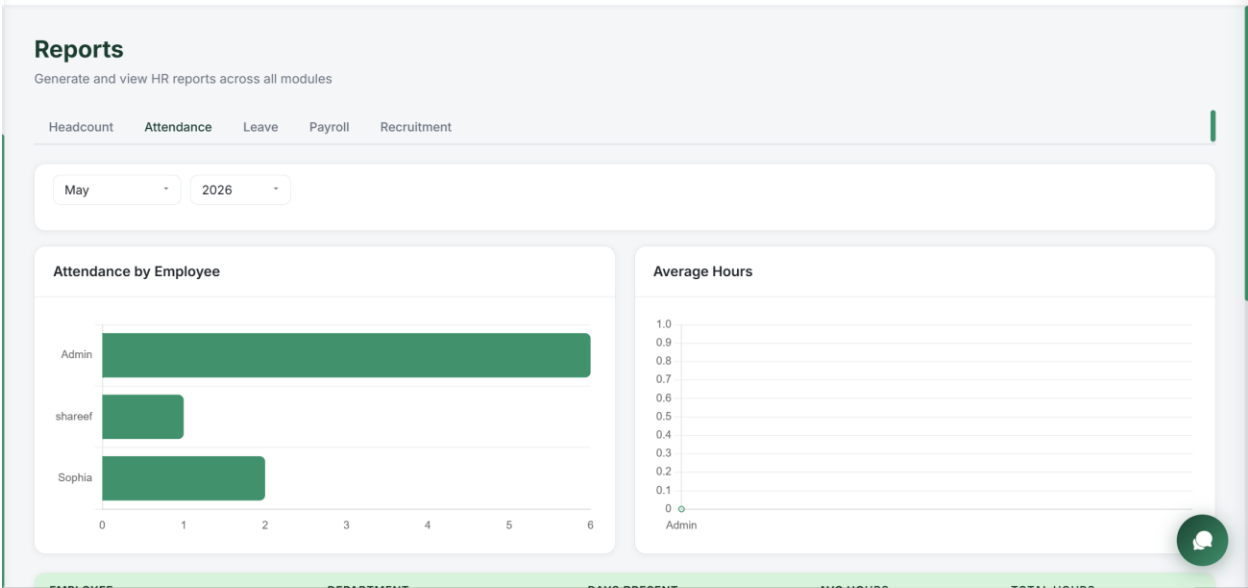
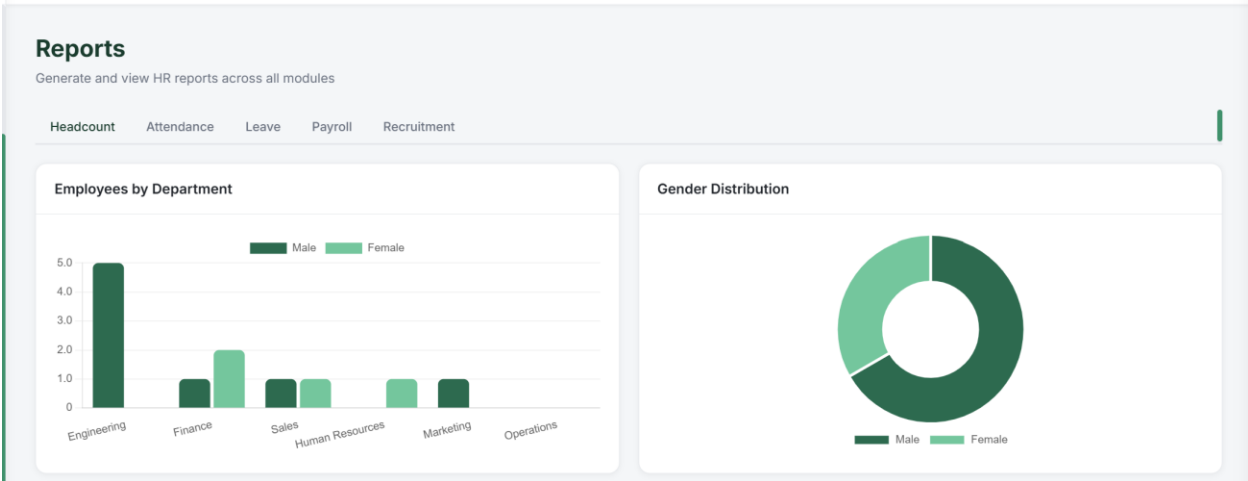
Figure10 : Performance



The **Performance** module is designed to track employee goals, monitor progress, and support continuous performance improvement. It allows managers and employees to create goals, set target dates, measure completion percentages, and provide feedback throughout the evaluation period. The module also includes visual dashboards and charts that display goal status, progress levels, and overall performance trends. By providing clear visibility into achievements and areas for improvement, the system helps increase productivity, align individual objectives with organizational goals, and support informed performance evaluations.

Reports and Charts

Figure 11: Reports Page with Interactive Charts



Reports

Generate and view HR reports across all modules

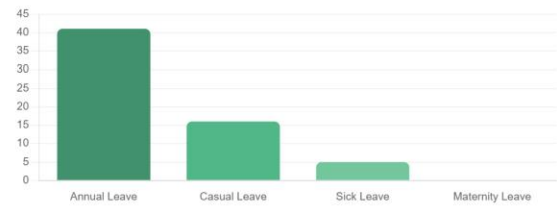
Headcount Attendance Leave Payroll Recruitment

Leave Overview



Approved Days Rejected Pending

Leave by Type



EMPLOYEE	DEPARTMENT	LEAVE TYPE	REQUESTS	APPROVED DAYS	REJECTED
Admin User	Engineering	Annual Leave	2	41d	0
Sophia Anderson	Human Resources	Casual Leave	2	13d	0

Reports

Generate and view HR reports across all modules

Headcount Attendance Leave Payroll Recruitment

8
Employees Paid

\$9001310K
Total Basic

\$1350197K
Allowances

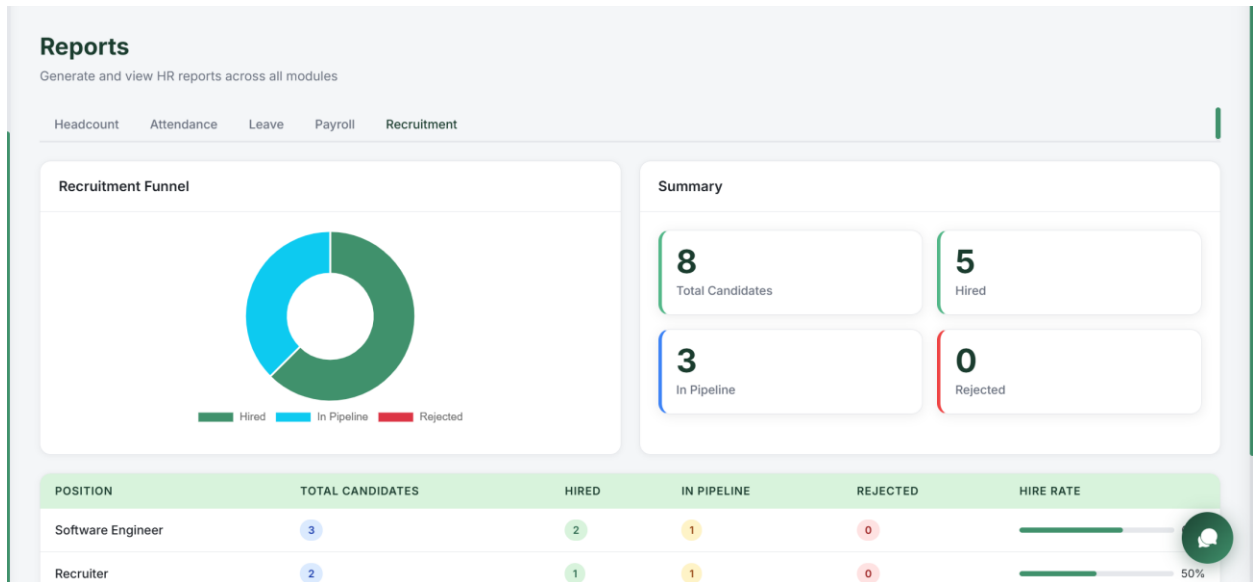
\$9901441K
Net Pay

Payroll by Department



Pay Breakdown

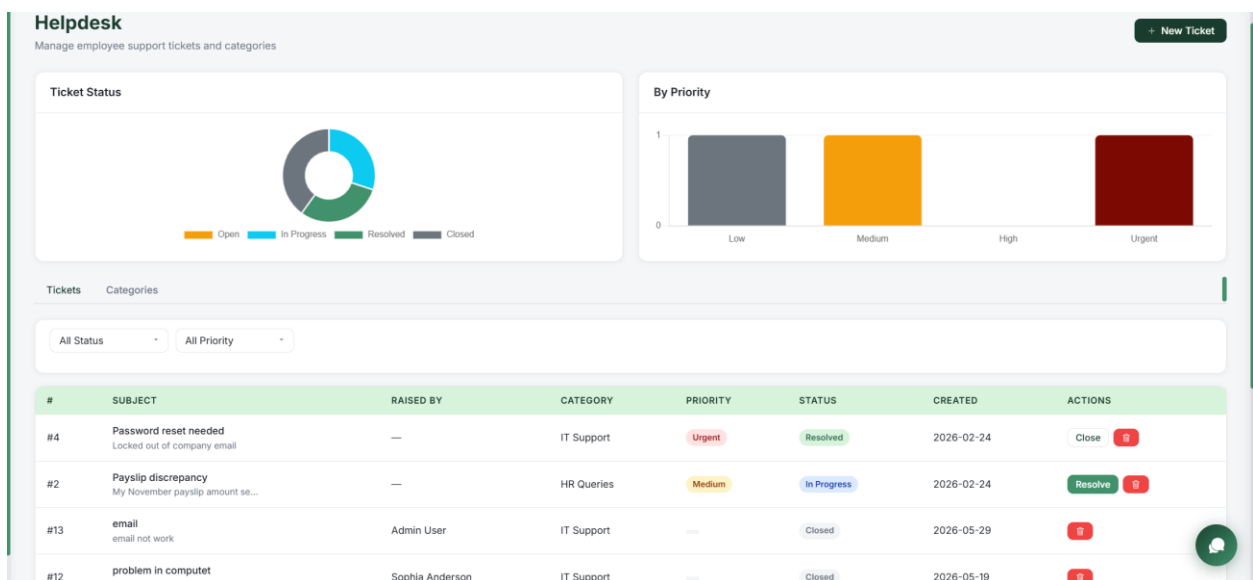




The reports page provides five comprehensive report types: headcount, attendance, leave, payroll, and recruitment. Each one combines interactive charts with detailed data tables. The charts are interactive, with tooltips on hover and click-to-drill-down where appropriate.

Helpdesk

Figure 12: Helpdesk



The **Helpdesk** module provides a centralized system for managing employee support requests and internal service tickets. Employees can submit tickets for issues such as IT problems, HR inquiries, payroll questions, or workplace requests, while administrators can categorize, assign, and track each ticket until it is resolved. The module helps improve communication between employees and support teams by ensuring that requests are handled efficiently, monitored through different status stages, and resolved within a reasonable timeframe. It also provides a clear history of all interactions, making support processes more organized, transparent, and accountable.

Assets

Figure 13: Assets

The figure displays two screenshots of the 'Assets' management interface. The top screenshot shows a table with columns for Name, Tag, Category, Purchase Date, Cost, Status, and Actions. The bottom screenshot shows a table with columns for #, Name, Description, and Actions.

Assets
Track company assets, allocations and categories

Assets Allocations Categories

All Status

NAME	TAG	CATEGORY	PURCHASE DATE	COST	STATUS	ACTIONS
car	123	Monitor	2026-05-30	\$500	Available	<input type="checkbox"/> <input type="checkbox"/>
laptop	aaa	Phone	2026-05-17	—	Available	<input type="checkbox"/> <input type="checkbox"/>
asdasd	dsfsd	Laptop	2026-05-26	\$2,323	Available	<input type="checkbox"/> <input type="checkbox"/>
ssssss	232131	Phone	2026-04-18	—	Available	<input type="checkbox"/> <input type="checkbox"/>
wrwr1	wqeqqw1	Laptop	2026-04-29	\$9,089,890	Available	<input type="checkbox"/> <input type="checkbox"/>
sharee	—	Phone	2026-04-09	\$22,222	Retired	<input type="checkbox"/> <input type="checkbox"/>
salsdfdk	—	Laptop	—	—	Allocated	<input type="checkbox"/> <input type="checkbox"/>

Assets
Track company assets, allocations and categories

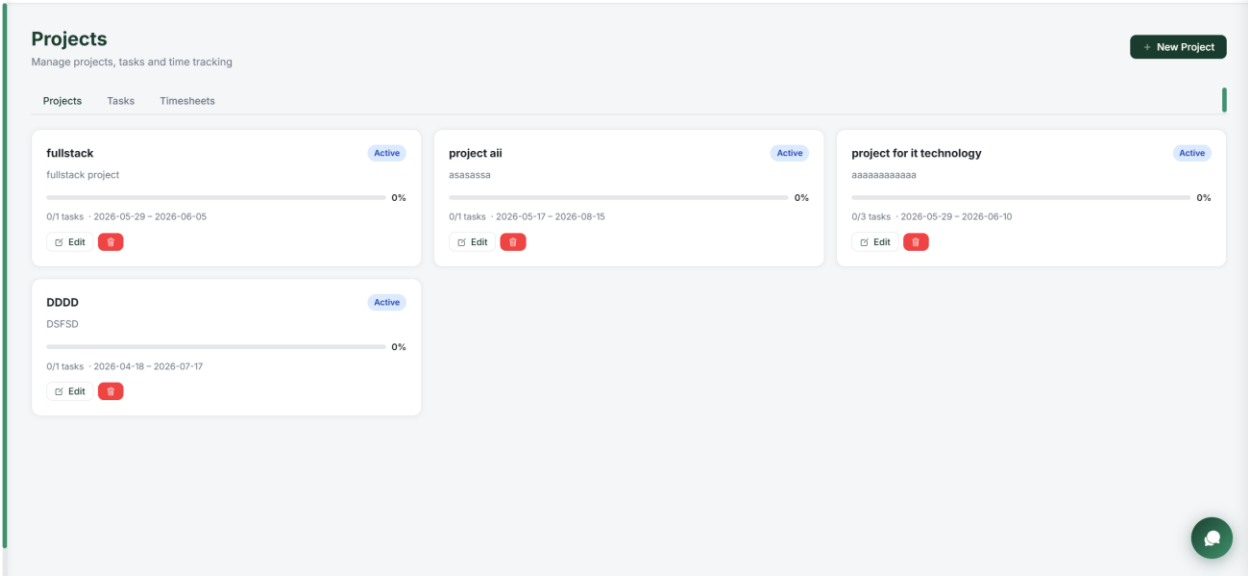
Assets Allocations Categories

#	NAME	DESCRIPTION	ACTIONS
1	Laptop	تلفون	<input type="checkbox"/> <input type="checkbox"/>
2	Monitor	—	<input type="checkbox"/> <input type="checkbox"/>
3	Phone	—	<input type="checkbox"/> <input type="checkbox"/>
4	cars	—	<input type="checkbox"/> <input type="checkbox"/>

The **Assets** module is used to track and manage all company-owned resources assigned to employees, such as laptops, mobile phones, access cards, and office equipment. It allows administrators to record assets, organize them into categories, and monitor their current status, whether they are in use, under maintenance, or returned. The module also keeps track of which employee is responsible for each asset and when it was assigned. This helps organizations maintain better control over company resources, reduce asset loss

Project

Figure 14: project



The **Projects** module helps organizations plan, organize, and track work across different teams and departments. It allows managers to create projects, assign team members, define start and end dates, and monitor project progress through different stages. Employees can view their assigned projects and responsibilities, while managers can track completion rates and overall project status. This module improves collaboration, enhances productivity, and ensures that projects are completed on time by providing clear visibility into tasks, deadlines, and team performance.

Settings Page

Figure 15: Settings and Configuration

Settings

Configure companies, departments, positions, shifts and holidays

Companies 1

Q Search... + Add

NAME ▲	PHONE ▲	EMAIL ▲	COUNTRY ▲	ACTIONS
hashtagll	0598573001	abdcom@gmail.com	Saudi Arabiaewrwr	

Departments 6

Q Search... + Add

NAME ▲	EMPLOYEES ▲	ACTIONS
Engineering	5	
Finance	3	
Human Resources	1	
Marketing	1	
Operations	0	

Job Positions 12

Q Search... + Add

TITLE ▲	DEPARTMENT ▲	ACTIONS
Accountant	Finance	
CFO	Finance	
Content Writer	Marketing	
DevOps Engineer	Engineering	
HR Manager	Human Resources	
Marketing Manager	Marketing	
Operations Manager	Operations	
Recruiter	Human Resources	
Sales Manager	Sales	
Sales Rep	Sales	
Senior Engineer	Engineering	

Shifts 2			
NAME	START	END	ACTIONS
Evening Shift	14:00	23:00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Morning Shift	08:00	17:00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Holidays 5			
HOLIDAY	DATE	RECURRING	ACTIONS
New Year	2026-01-01	Once	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Eid Al-Fitr	2026-03-20	Once	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Eid Al-Adha	2026-05-27	Once	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Independence Day	2026-07-04	Once	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Christmas	2026-12-25	Once	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

The settings page lets administrators configure the organizational structure, including companies, departments, job positions, shifts, and holidays. Each section has its own list with add, edit, and delete operations, all behind permission checks.

Permissions Management

Figure16 : Granular Permissions Management

User Permissions

Manage individual user permissions

Users 13

Search users...

- AU** Admin User (ADMIN) Engineering
- OD** Olivia Davis (EMPLOYEE) Finance
- WM** William Martinez (EMPLOYEE) Engineering
- SA** Sophia Anderson (EMPLOYEE) Human Resources
- LT** Liam Taylor (EMPLOYEE) Engineering
- NJ** Noah Jackson (EMPLOYEE) Sales
- IW** Isabella White

Admin User (ADMIN) Engineering Apply Defaults

Main 1/1

Dashboard

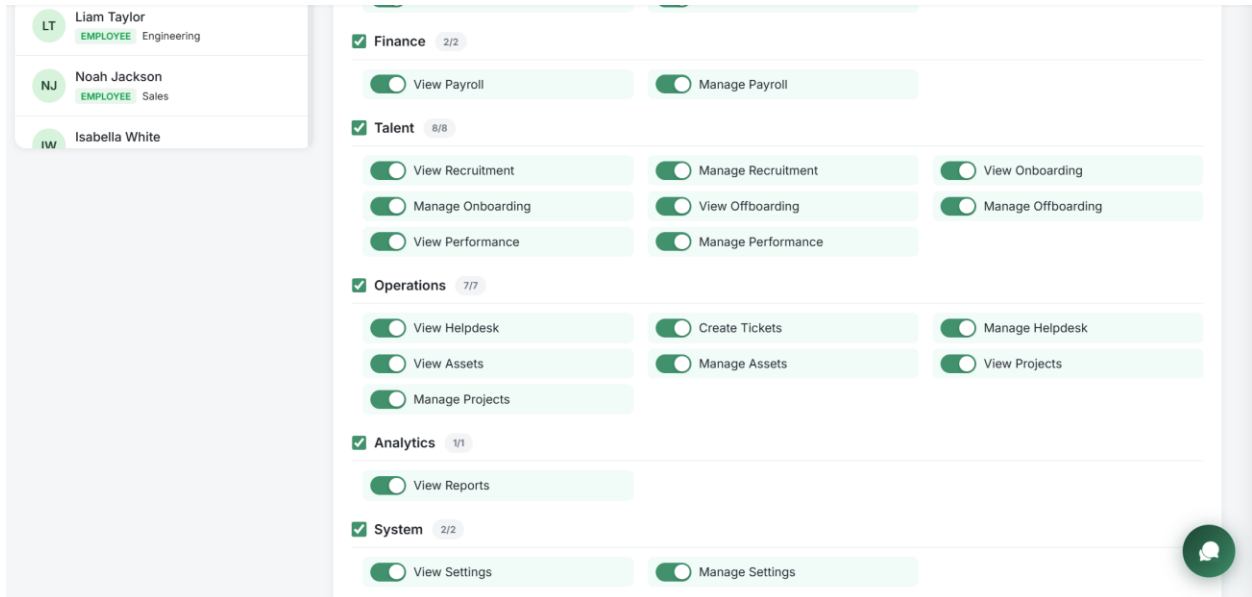
Workforce 11/11

View Employees Create Employees Edit Employees
 Delete Employees View Attendance Clock In/Out
 Manage Attendance View Leave Apply Leave
 Approve/Reject Leave Manage Leave Types

Finance 2/2

View Payroll Manage Payroll

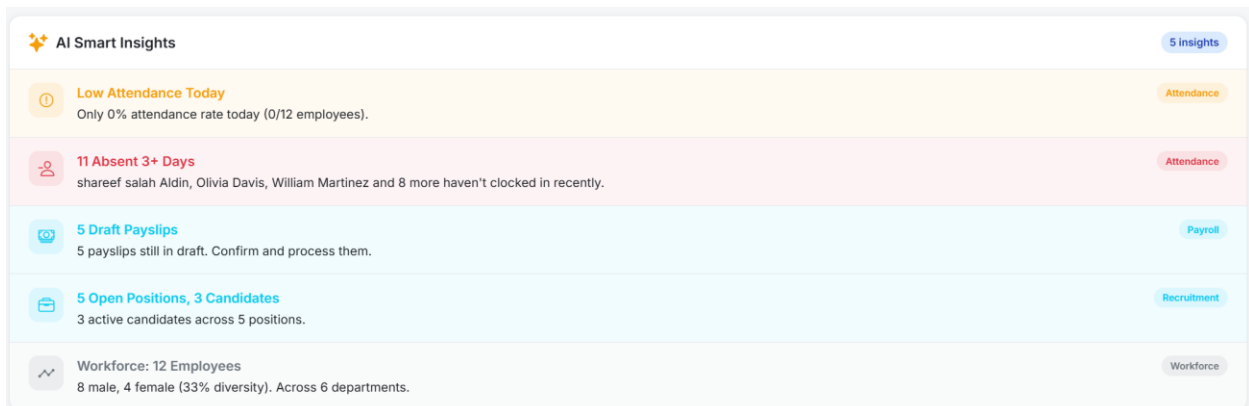
Talent 8/8



The permissions management page is one of the most powerful parts of the system. Administrators can see every user, expand their permission list, and toggle each of the 32 permissions individually. There are also role-based defaults that can be applied with one click, after which the administrator can still customize specific permissions if needed.

AI Smart Insights

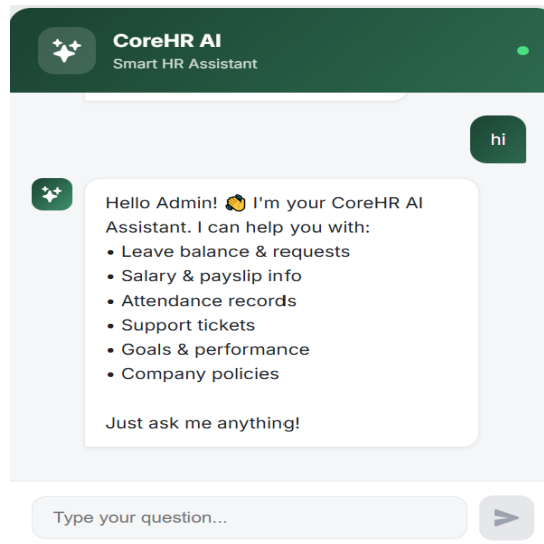
Figure 17: AI-Powered Smart Insights Panel



The AI insights panel appears on the dashboard and shows context-aware recommendations based on the data. Each insight is color-coded by severity, and clicking on one takes the user to the relevant page where they can take action.

HR Chatbot

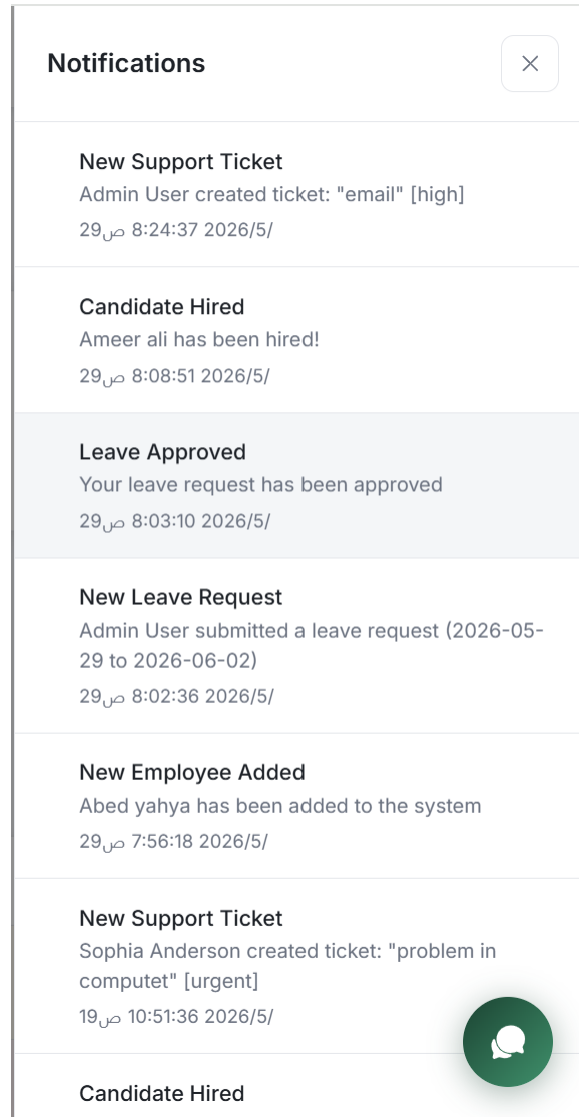
Figure 18: HR Chatbot Interface



The chatbot is accessed through a floating button at the bottom-right corner of every web page. Users can type questions in English or Arabic, and the chatbot responds with relevant information pulled from the database. Common queries include checking leave balance, salary details, attendance, and company policies.

Notifications Panel

Figure 19: Notifications Panel

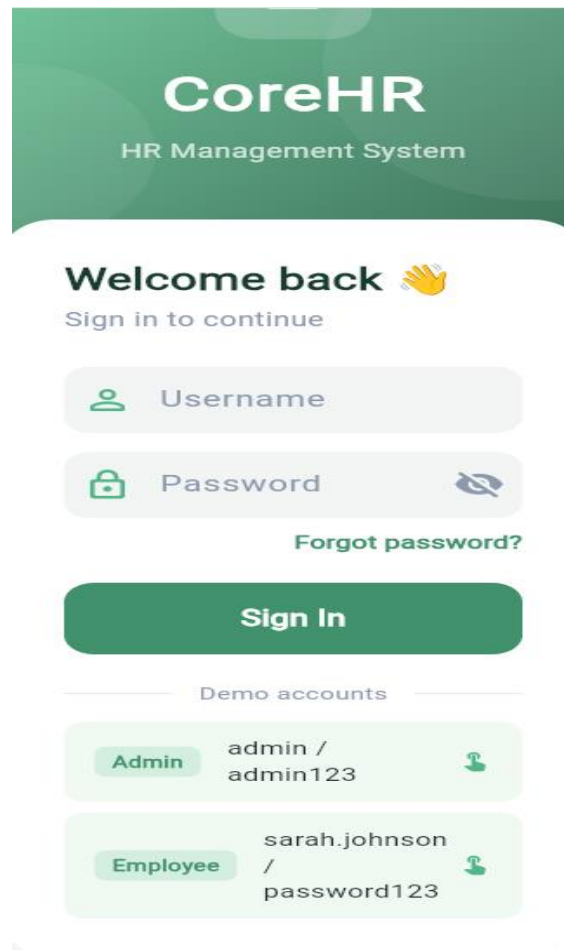


Notifications appear in a dropdown panel accessed through the bell icon in the navbar. The unread count is shown as a badge on the bell. Each notification displays its title, message, and time, and unread ones are visually highlighted. Users can mark notifications as read individually, mark all as read, or delete them.

Results - Mobile App (Employee View)

Login Screen

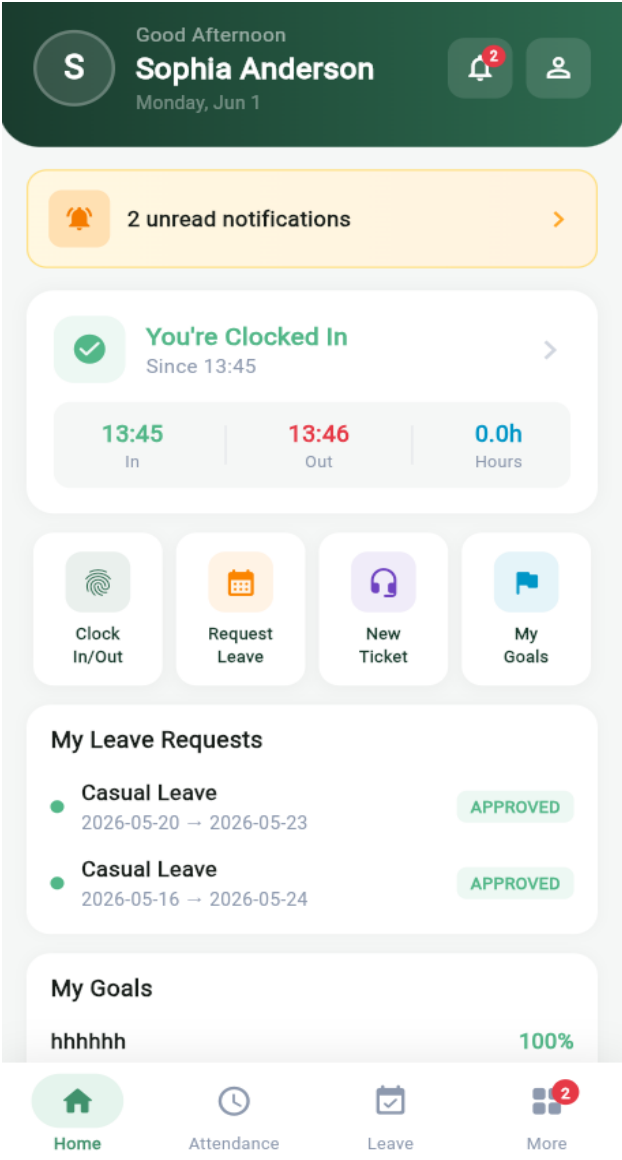
Figure 14: Mobile Login Screen



The mobile login screen mirrors the web login functionality but is designed specifically for phone-sized screens. After successful login, the user's permissions are fetched from the server, and the navigation structure adapts to show only the screens they can access.

Employee Dashboard

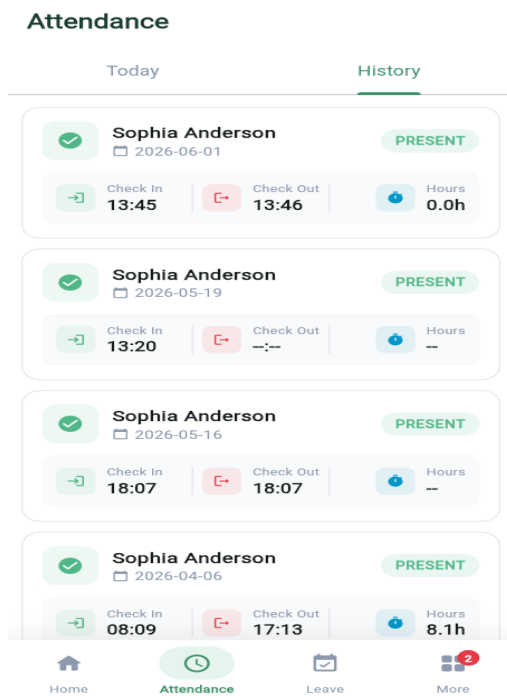
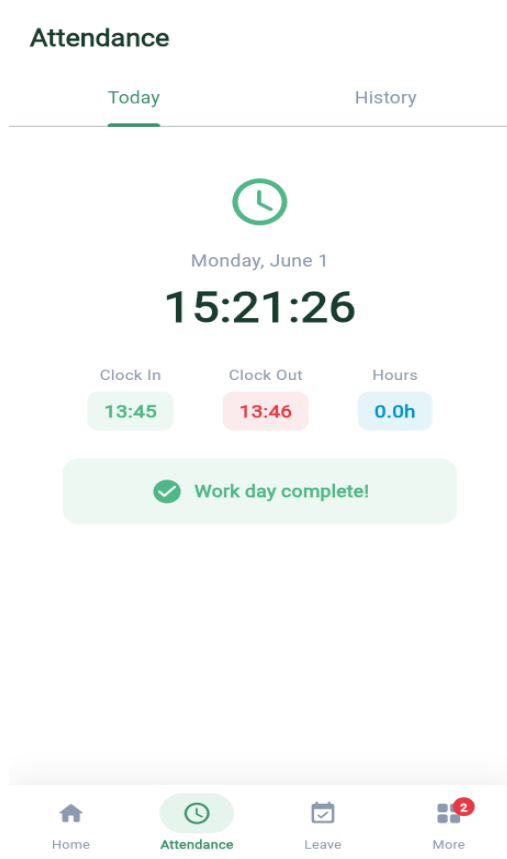
Figure 15: Employee Dashboard



The employee dashboard is the personal HR hub. At the top is a clock-in card showing the current status for today, with check-in and check-out times if available. Below are quick action buttons for the most common tasks. A leave summary section shows the most recent requests with their status, and a goals section displays active performance goals with progress bars.

Attendance History

Figure 16: Attendance History



The attendance history screen shows the user's past attendance records as cards. Each card displays the date, check-in time, check-out time, and total worked hours. The data is automatically filtered to show only the current user's records.

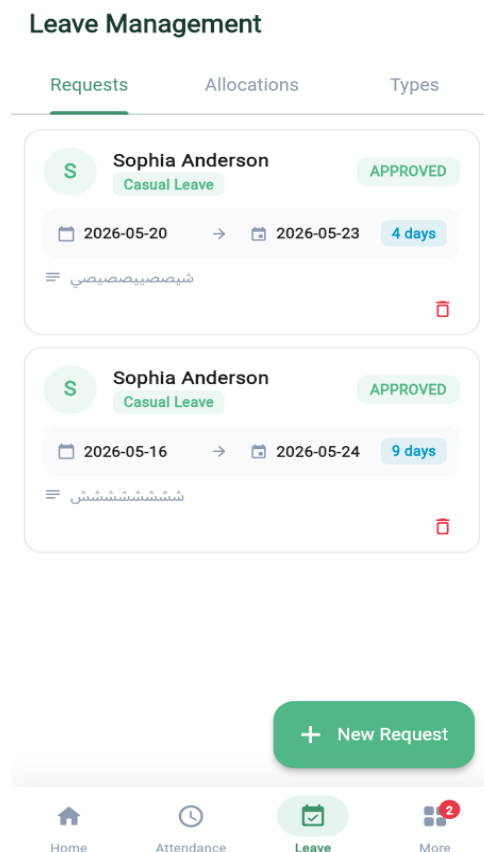
Leave Request

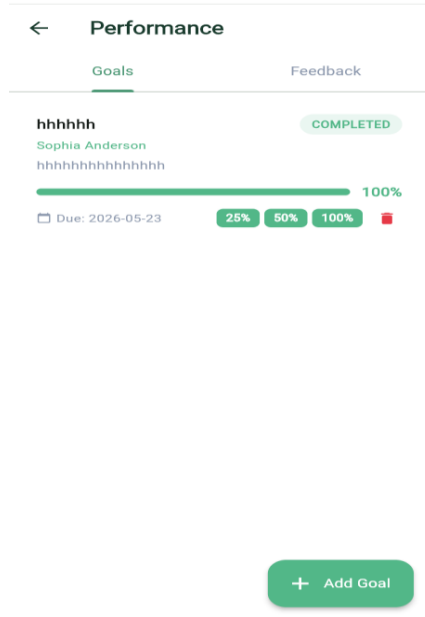
Figure 17: Leave Request with AI Suggestions

The leave request screen lets employees submit a new request by selecting the type, dates, and entering a reason. As the user picks dates, the AI smart suggestion engine analyzes the selection and shows contextual tips. After submission, the request appears in the user's leave history with a pending status.

Performance Goals

Figure 18: Performance Goals with Progress

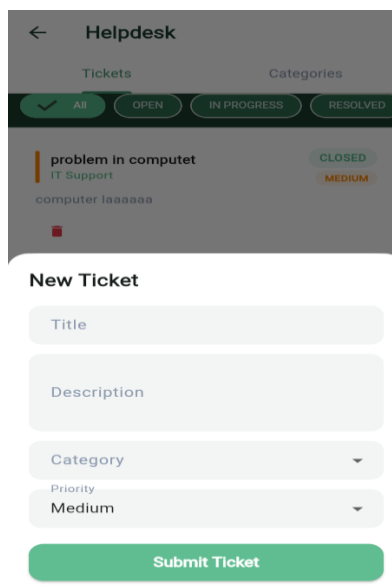


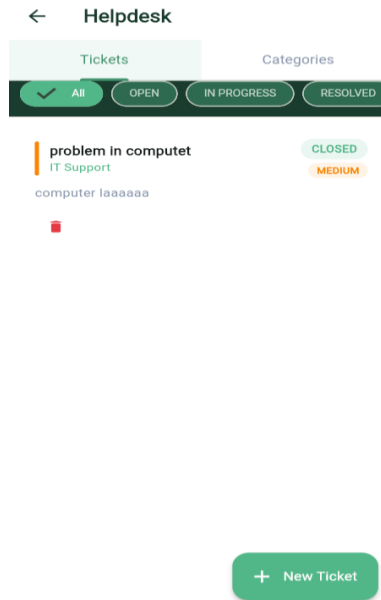


The goals screen displays all the user's active performance goals with visual progress bars. Each goal shows its title, description, target date, and current progress percentage. The colors of the progress bars indicate how close to the target the user is.

Helpdesk - Create Ticket

Figure 19: Helpdesk Ticket Creation

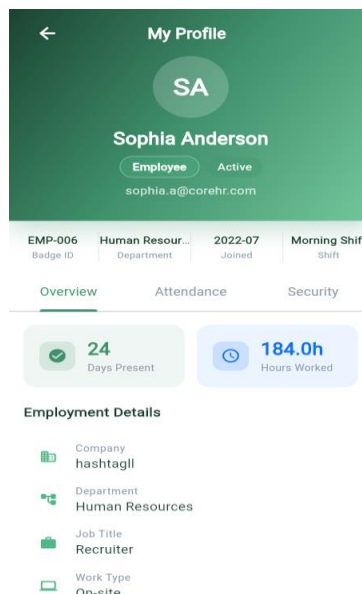




The helpdesk screen allows employees to create new support tickets and view their existing ones. Creating a ticket involves filling in a title, description, category, and priority. After creation, the ticket appears in the list with its current status.

Profile Page

Figure 20: Employee Profile



Resume / CV



No CV uploaded yet

Upload CV

PDF, DOC, DOCX – max 5 MB

Account Information



User ID

#8



Username

sophia.anderson



Email

sophia.a@corehr.com



Role

Employee

Overview

Attendance

Security

Change Password



Current Password



New Password

Minimum 6 characters



Confirm Password

Update Password

Account Security



Password

Enabled



Two-Factor Auth

Not set up



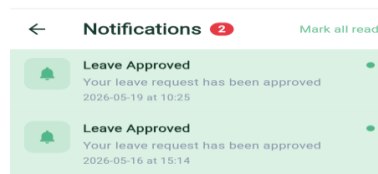
Login History

Available

The profile page shows the user's personal and work information, including their photo, name, email, phone, department, position, and joining date. Some fields can be edited directly from the page, while others are read-only and require an HR administrator to change.

Notifications Screen

Figure 21: Mobile Notifications Screen



The notifications screen on mobile is accessed through the More tab in the bottom navigation. An unread badge appears on the tab when there are new notifications. Each notification shows its title, message, and creation time, with unread ones highlighted.

Results - Mobile App (Admin View)

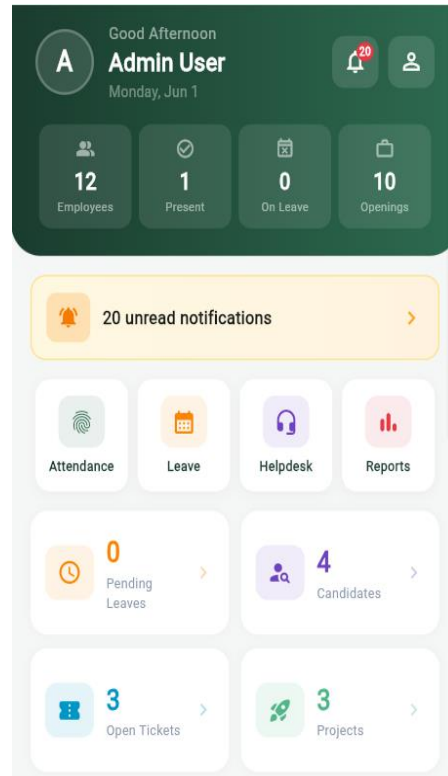
Admin Dashboard

Figure 22: Admin Mobile Dashboard

The admin dashboard on mobile shows organization-wide statistics in a compact form: total employees, today's attendance, employees on leave, and open positions. Below the stats are charts showing attendance trends and team distribution, all sized appropriately for mobile screens.

Employee Management

Figure 23: Mobile Employee List with Edit/Delete



Managers can view, edit, and delete employees from the mobile app. Tapping an employee opens an edit dialog with their information pre-filled. Long-pressing an employee gives the option to delete, with a confirmation prompt.

Edit Employee

Personal Information

First Name *	Last Name *
shareef	salah Aldin
Email *	Username
shareefsalahat2	shareef
Badge ID	Gender
222	Male ▾
Date of Birth	Phone
07/31/2001 📅	0583800428

Work Information

Department	Job Position
Engineering ▾	Software E... ▾
Work Type	Shift
Hybrid ▾	— Select — ▾
Date of Joining	Salary (\$/year)
03/02/2026 📅	3000000000

Add New Employee

Personal Information

First Name *	Last Name *
Email *	Username *
Badge ID	Gender
	Male ▾
Date of Birth	Phone
mm/dd/yyyy 📅	

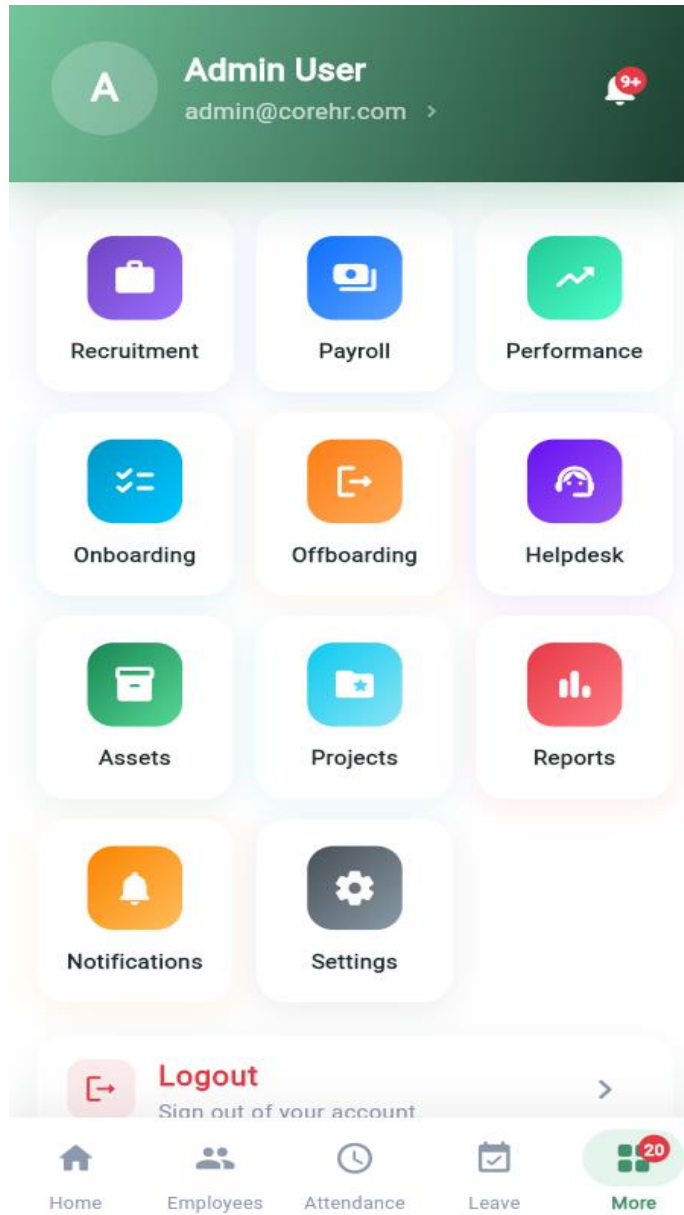
Work Information

Department	Job Position
— Select — ▾	— Select — ▾
Work Type	Shift
— Select — ▾	— Select — ▾
Date of Joining	Salary (\$/year)
mm/dd/yyyy 📅	

More Screen

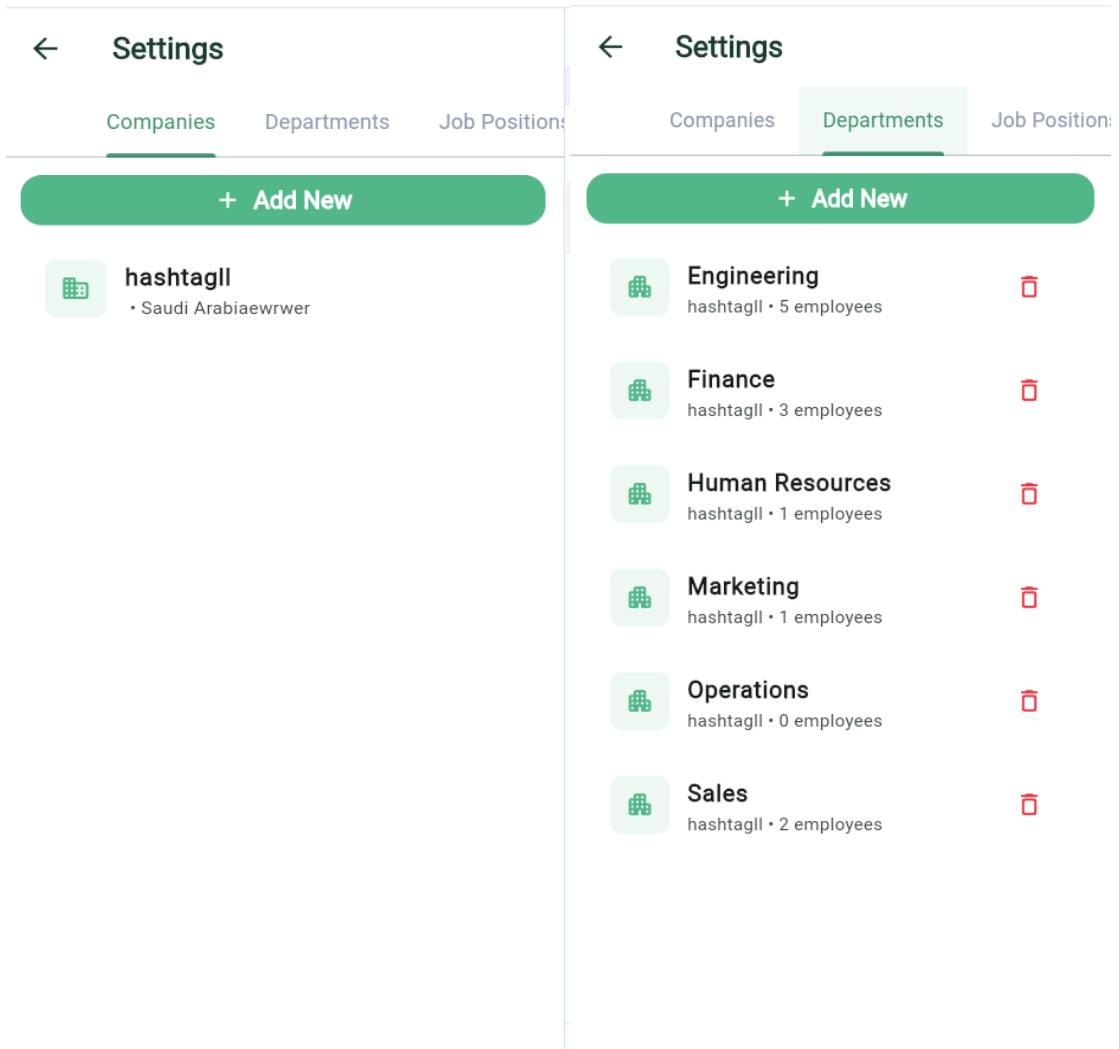
Figure 24: More Screen with All Modules

The More screen provides access to all the modules that do not fit in the main bottom navigation: settings, payroll, projects, assets, reports, and others. Each module is shown only if the user has the relevant permission.



Settings Management
















Figure 25: Mobile Settings Page



← Settings

Companies Departments **Job Positions** Shifts



+ Add New

-  **CFO**
Finance 
-  **Content Writer**
Marketing 
-  **DevOps Engineer**
Engineering 
-  **HR Manager**
Human Resources 
-  **Marketing Manager**
Marketing 
-  **Operations Manager**
Operations 
-  **Recruiter**
Human Resources 
-  **Sales Manager**
Sales 

← Settings

Departments Job Positions Shifts **Holidays**











+ Add New

-  **Evening Shift**
14:00 → 23:00
-  **Morning Shift**
08:00 → 17:00

← Settings

Departments Job Positions Shifts **Holidays**

+ Add New

	New Year 2026-01-01	
	Eid Al-Fitr 2026-03-20	
	Eid Al-Adha 2026-05-27	
	Independence Day 2026-07-04	
	Christmas 2026-12-25	

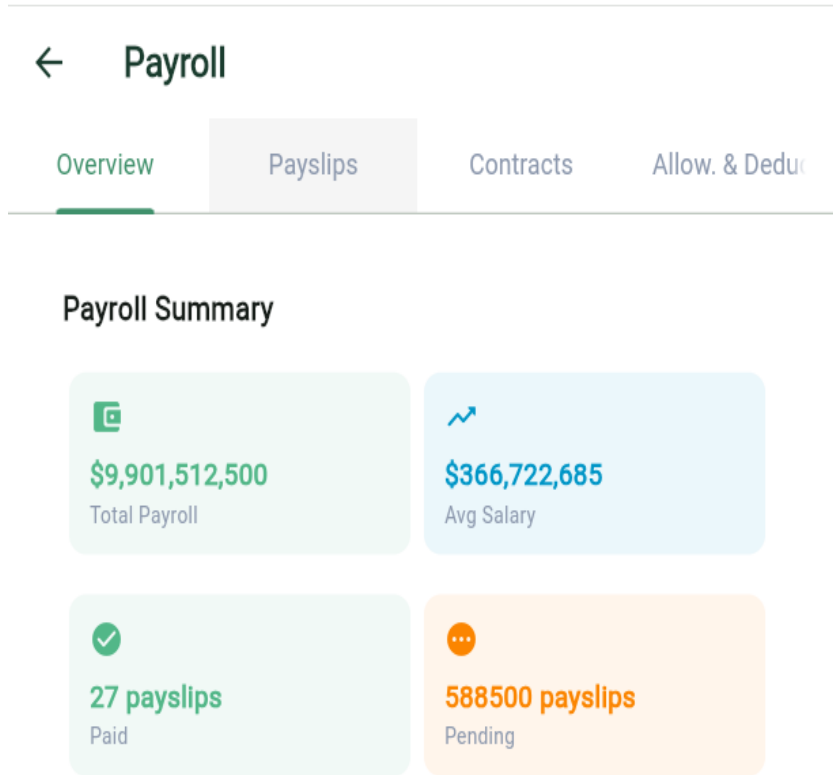
The settings screen on mobile lets administrators manage companies, departments, positions, shifts, and holidays. Each section has its own tab with full CRUD capabilities.

Payroll on Mobile

Figure 26: Mobile Payroll with Contract Editing

The screenshot displays a mobile payroll application interface. At the top, there is a navigation bar with a back arrow and the title 'Payroll'. Below this is a secondary navigation bar with tabs: 'Overview', 'Payslips', 'Contracts', 'Allow. & Deduct.', and 'Allowances'. The 'Contracts' tab is currently selected. The main content area is divided into two columns. The left column lists employees: 'shareef salah Aldin', 'Admin User', 'James Brown', and 'Olivia Davis'. For each employee, their contract details are shown: 'Basic Salary', 'Start Date', and 'Type'. The right column shows a list of 'Allowances' and 'Deductions' for each employee, with their respective percentages. The 'Allowances' section is highlighted with a green background, and the 'Deductions' section is highlighted with a red background.

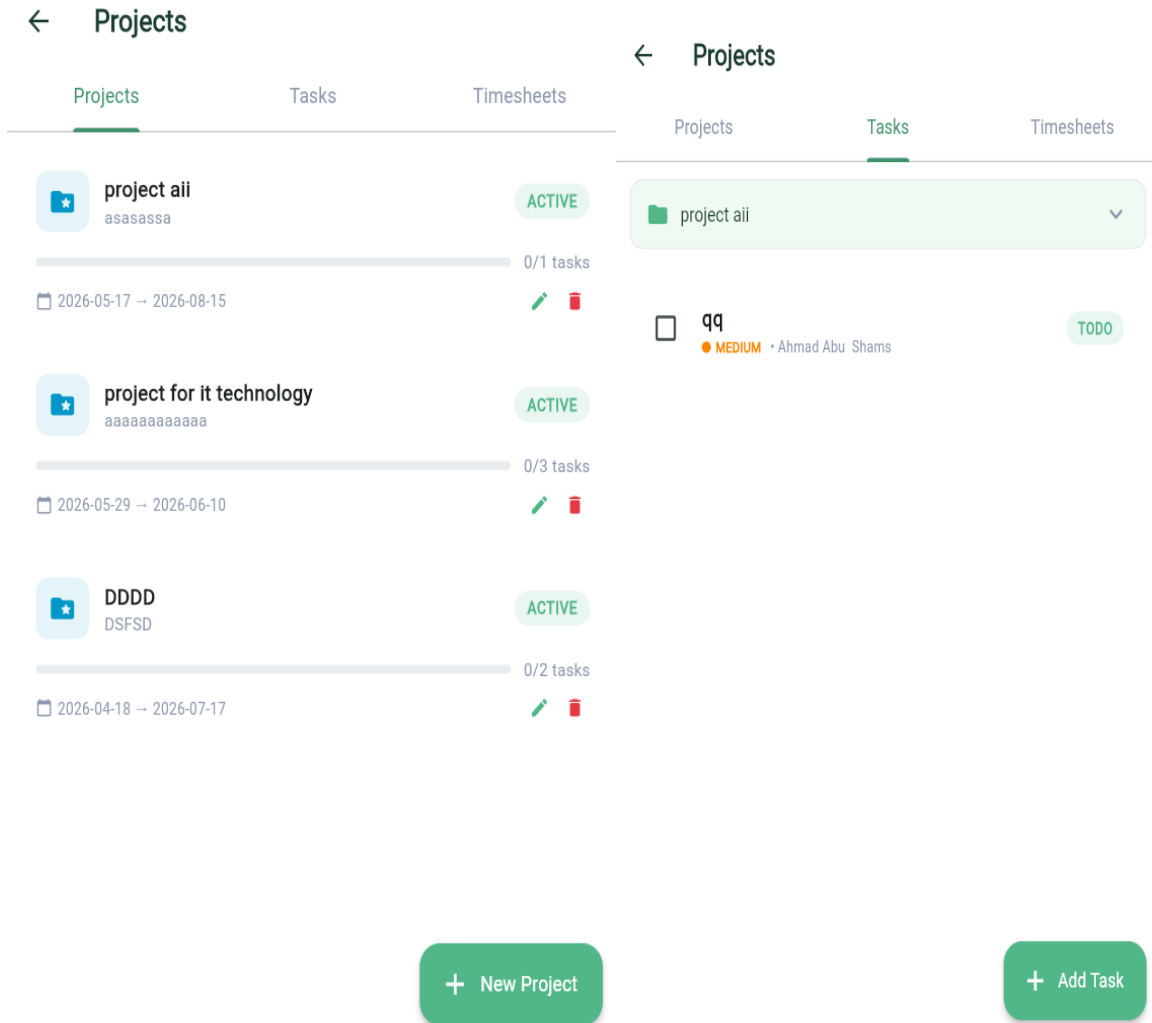
Employee	Basic Salary	Start Date	Type	Contract Status	Allowances	Deductions
shareef salah Aldin	\$3,000,000,000	2026-03-02	permanent	ACTIVE	Housing: 500.0% Meal: 150.0%	
Admin User	\$55,000	2022-01-15	permanent	ACTIVE	Performance Bonus: 10.0% Transport: 200.0%	
James Brown	\$70,000	2022-01-15	permanent	ACTIVE		Insurance: 200.0% Social Security: 5.0%
Olivia Davis				ACTIVE		

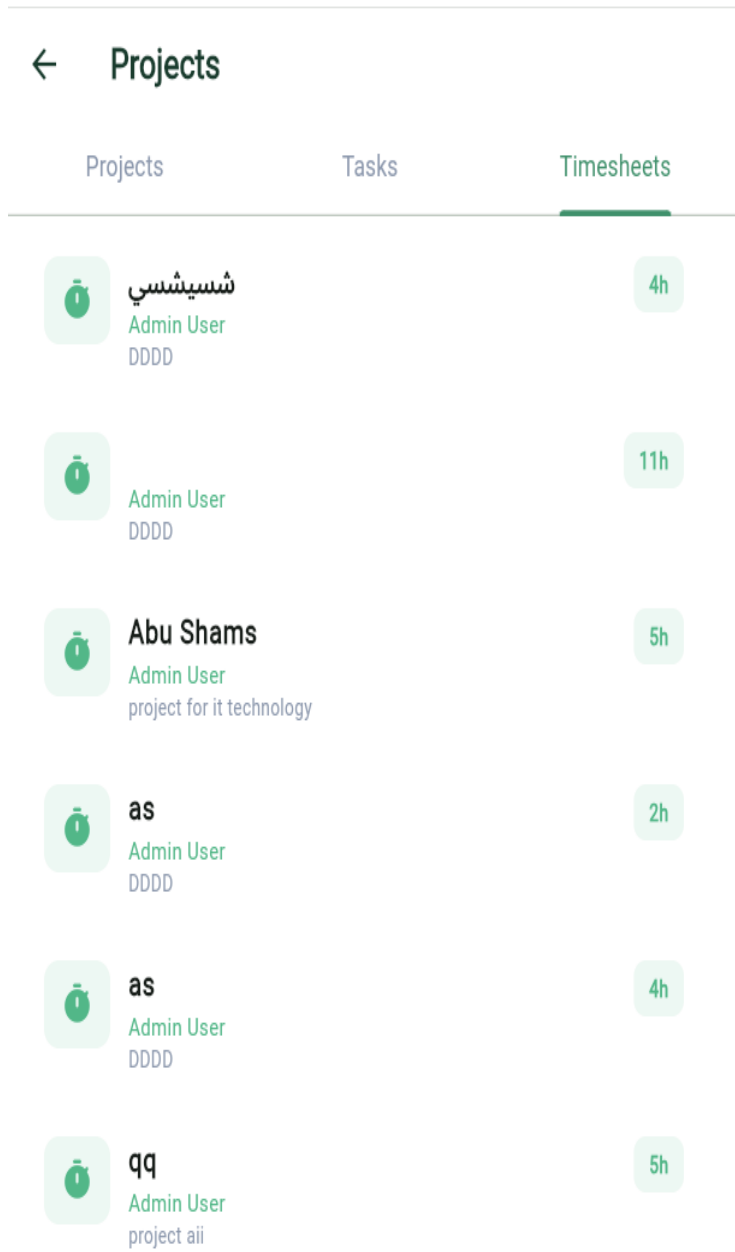


The payroll screen on mobile uses bottom sheet dialogs to edit contracts and payslips. Managers can advance payslips through their lifecycle and update contract terms with a few taps.

Project Management

Figure 27: Mobile Project Management

















The project management screen lets managers create projects, add tasks, and log time, all through bottom sheet dialogs. Each project shows its tasks with their status and assigned employees.

← Assets

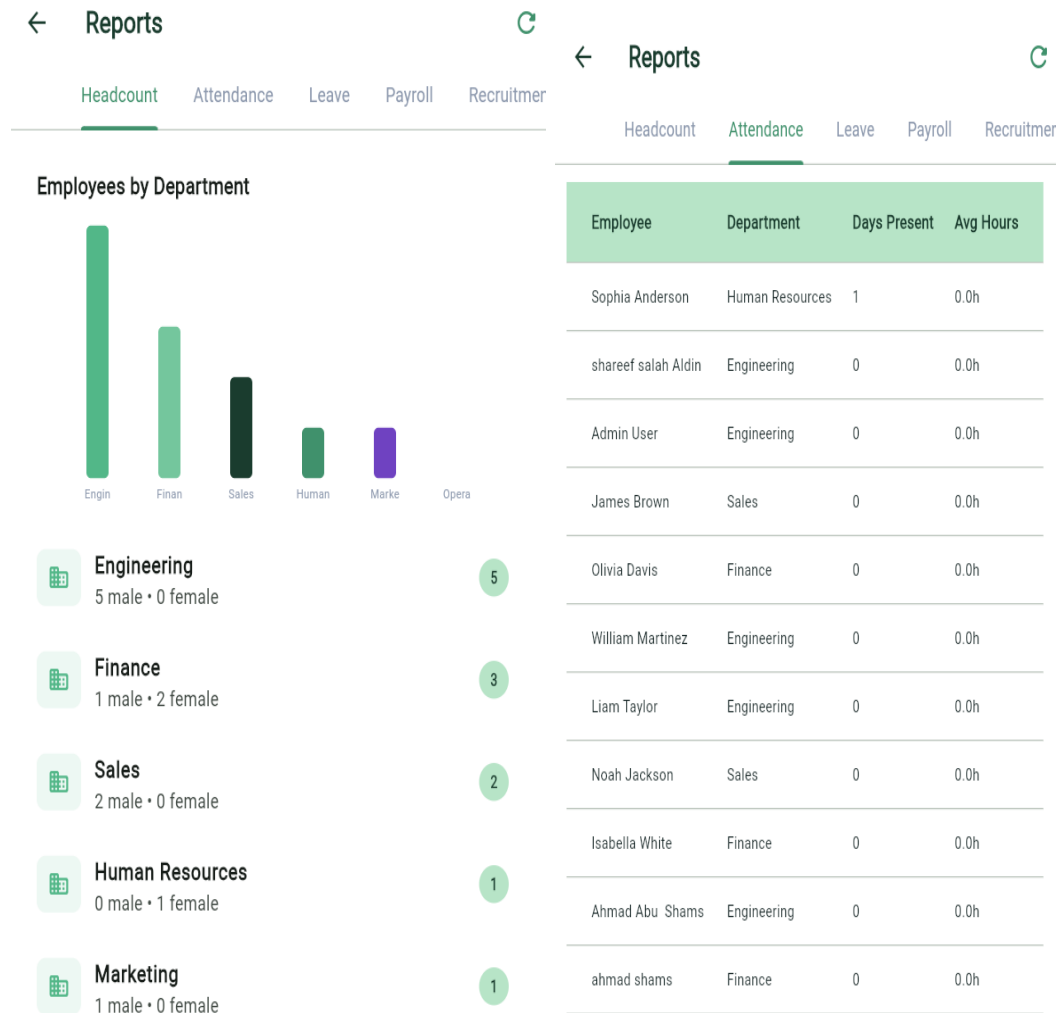
Assets 8 Allocated 1 Categories 4

 Laptop نصفصنقص	3		
 Monitor	0		
 Phone	5		
 fsdfs1q sdasd	0		

The asset management screen has a tabbed interface with separate views for assets, allocations, and categories. Managers can add new assets, allocate them to employees, and track their status.

Report Management

Figure 29: Mobile Report Management



← Reports



Headcount Attendance **Leave** Payroll Recruitment

Employee	Leave Type	Requests	Approved Days
Admin User	Annual Leave	2	41
Sophia Anderson	Casual Leave	2	13
Admin User	Casual Leave	1	3
shareef salah Aldin		0	0
Admin User	Maternity Leave	1	0
James Brown		0	0
Olivia Davis		0	0
William Martinez		0	0
Liam Taylor		0	0
Noah Jackson		0	0
Isabella White		0	0

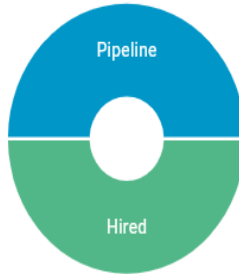
← Reports



Headcount Attendance Leave **Payroll** Recruitment

Employee	Department	Basic	Net Pay
shareef salah Aldin	Engineering	\$9,000,000,000	\$9,900,000,000
Isabella White	Finance	\$210,000	\$231,000
Noah Jackson	Sales	\$200,000	\$220,000
Liam Taylor	Engineering	\$180,000	\$198,000
Sophia Anderson	Human Resources	\$170,000	\$187,000
Admin User	Engineering	\$165,000	\$181,500
William Martinez	Engineering	\$160,000	\$176,000
Olivia Davis	Finance	\$150,000	\$165,000
James Brown	Sales	\$140,000	\$154,000

Recruitment Funnel



Software Engineer

Total: 3



Recruiter

Total: 2



Senior Engineer

Total: 1



DevOps Engineer

Total: 1



HR Manager

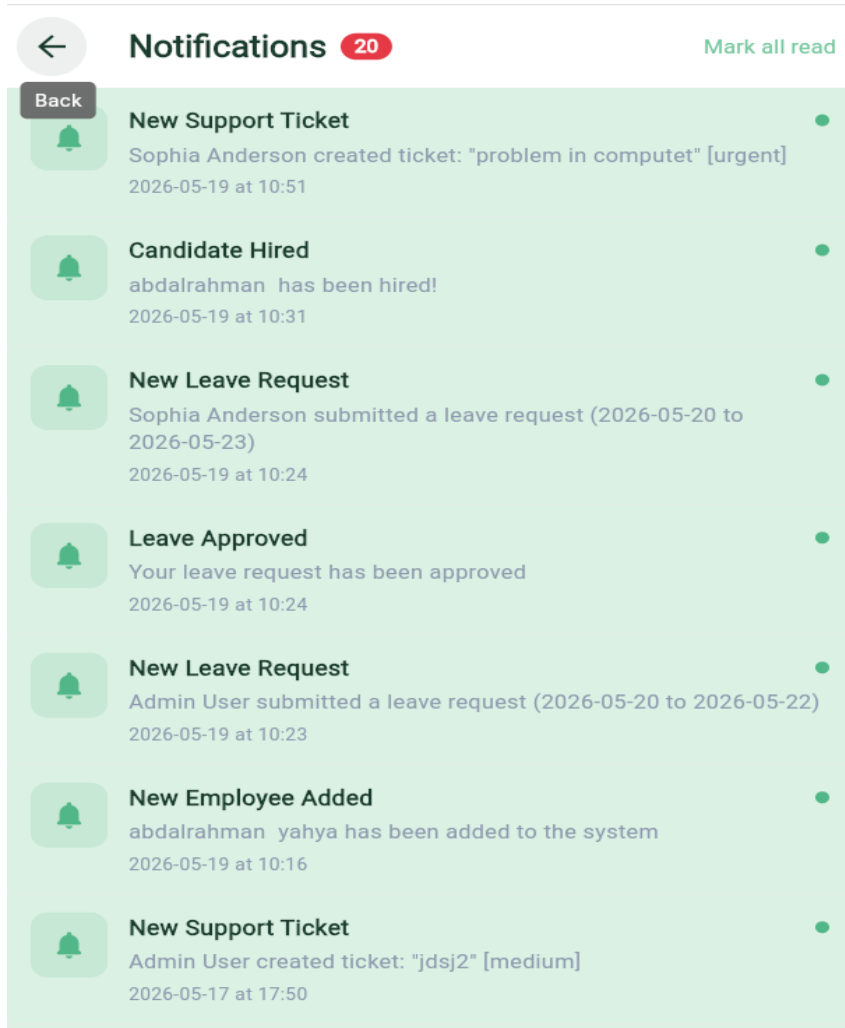
Total: 1



The mobile application offers a streamlined HR experience through an intuitive interface that enables users to manage attendance, leave requests, payroll details, and employee information on the go. It includes interactive dashboards, real-time updates, and easy navigation, allowing users to access key information and perform essential tasks efficiently from their mobile devices.

Notification Management

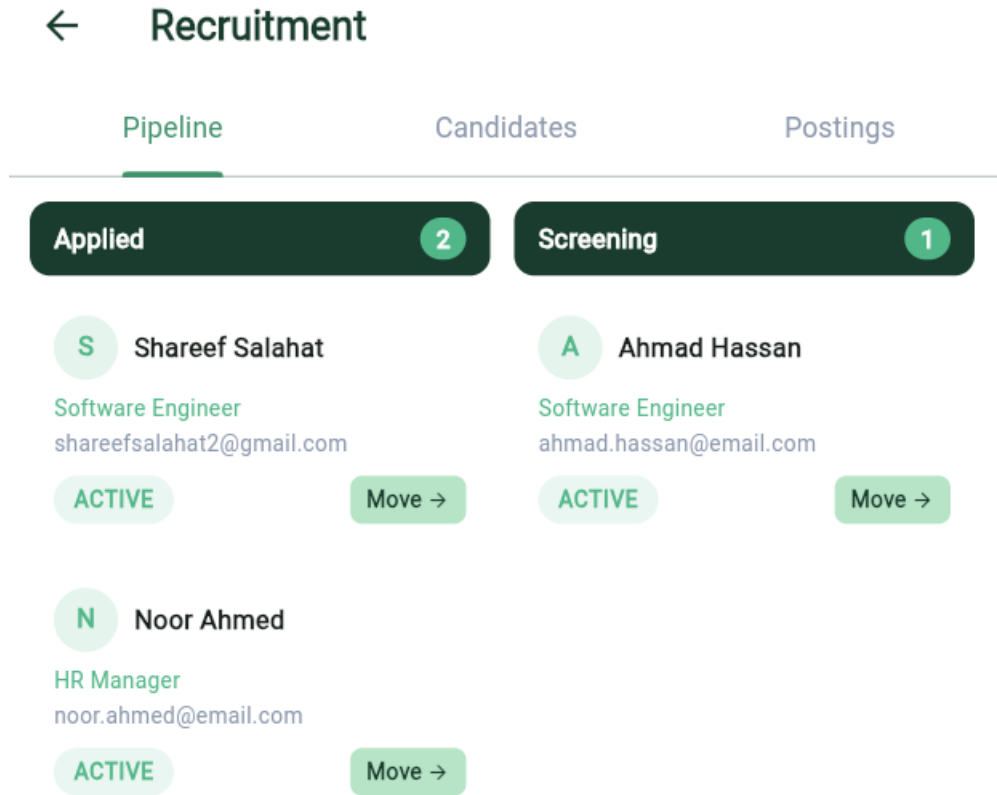
Figure 30: Mobile Notification



The notifications feature keeps users informed about important HR activities and updates in real time. Employees receive alerts for leave request approvals or rejections, attendance reminders, payroll updates, and other relevant announcements. The notification system enhances communication, ensures timely awareness of important events, and helps users stay updated with organizational activities directly through the application.

Recruitment Management

Figure 31: Mobile Recruitment Management



←

Recruitment

Back

Pipeline
Candidates
Postings

🔍

Stage

All stages
▼

Status

All
▼

8 candidates

S

Shareef Salahat
Software Engineer

shareefsalahat2@gmail.com

ACTIVE

Leadership

TeamWork

Project Management

Communication Skills

React

Node

↓ Download CV

↻ Replace

S

Shareef Salahat
Software Engineer

shareefsalahat2@gmail.com

HIRED

Leadership

TeamWork

Project Management

Communication Skills

React

Node

↓ Download CV

↻ Replace

A

Ahmad Abu Shams
Recruiter

aabushams80@gmail.com

HIRED

Leadership

TeamWork

Project Management

Communication Skills

Node JS

The Recruitment Management feature enables users to track job openings, review candidate applications, and monitor the hiring process directly from the mobile application. It provides quick access to recruitment information, helping managers stay updated on hiring activities and make timely decisions.

← Onboarding

Tasks

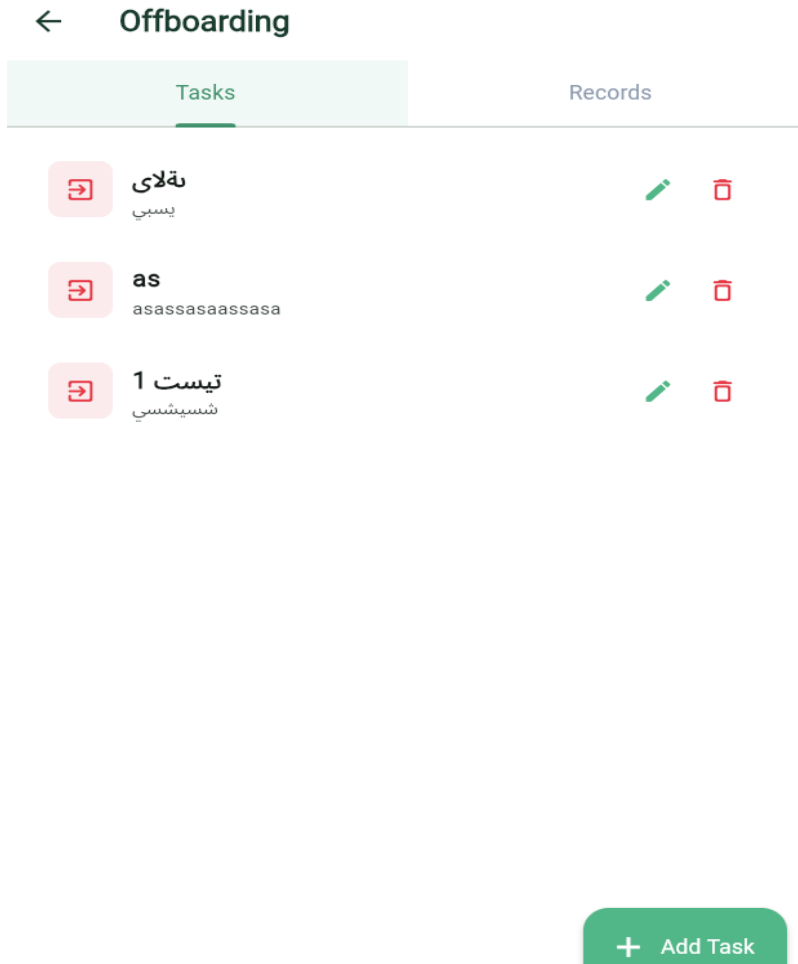
Records

<input checked="" type="checkbox"/>	رءبسىبس Isabella White	COMPLETED
<input checked="" type="checkbox"/>	بلا تلى Isabella White	COMPLETED
<input checked="" type="checkbox"/>	تنتنت 1 Isabella White	COMPLETED
<input type="checkbox"/>	رءبسىبس Ahmad Abu Shams	PENDING
<input type="checkbox"/>	بلا تلى Ahmad Abu Shams	PENDING
<input type="checkbox"/>	تنتنت 1 Ahmad Abu Shams	PENDING

The Onboarding feature helps streamline the integration of new employees into the organization. Through the mobile application, users can view onboarding tasks, track their progress, access required documents, and stay informed about the steps needed to complete the onboarding process efficiently.

Offboarding

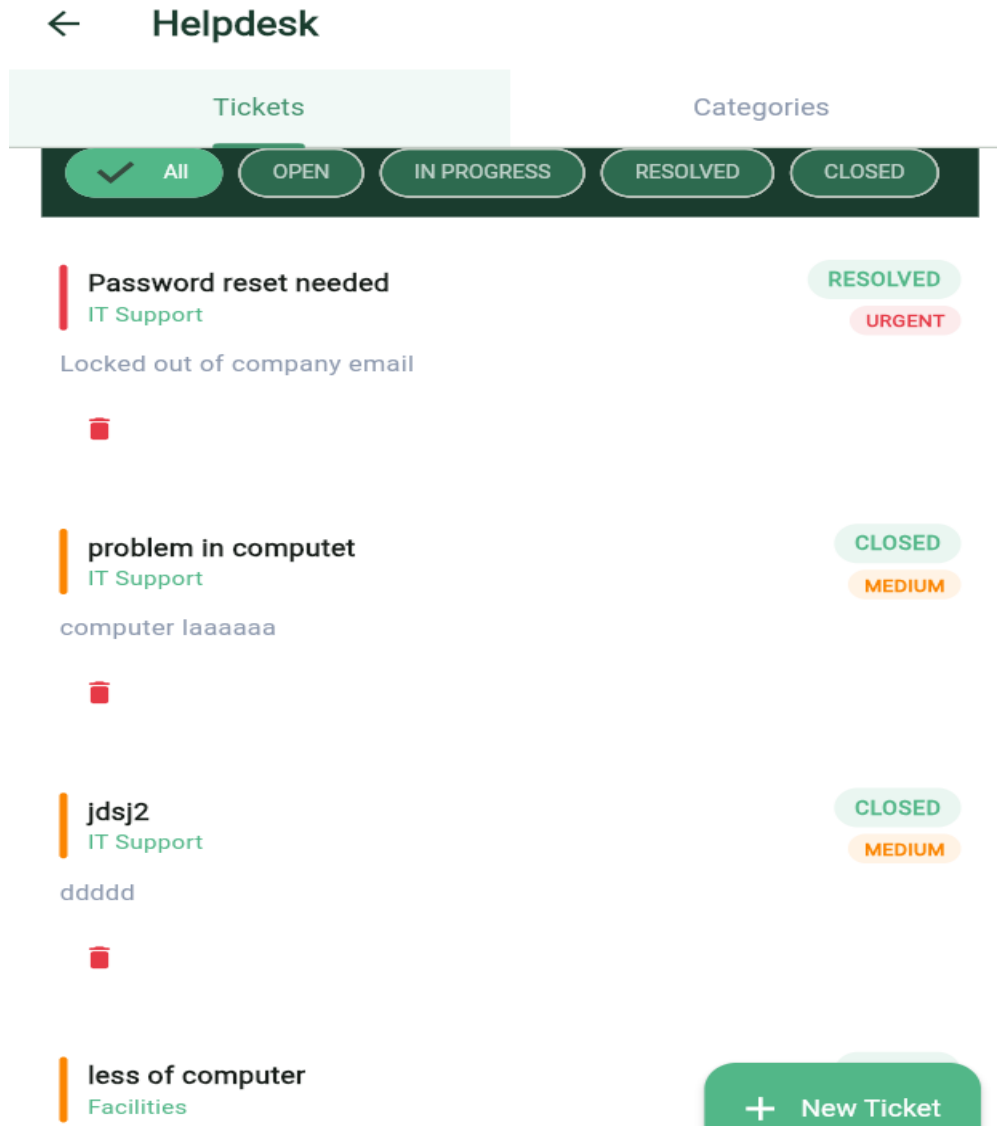
Figure 32: Mobile Offboarding

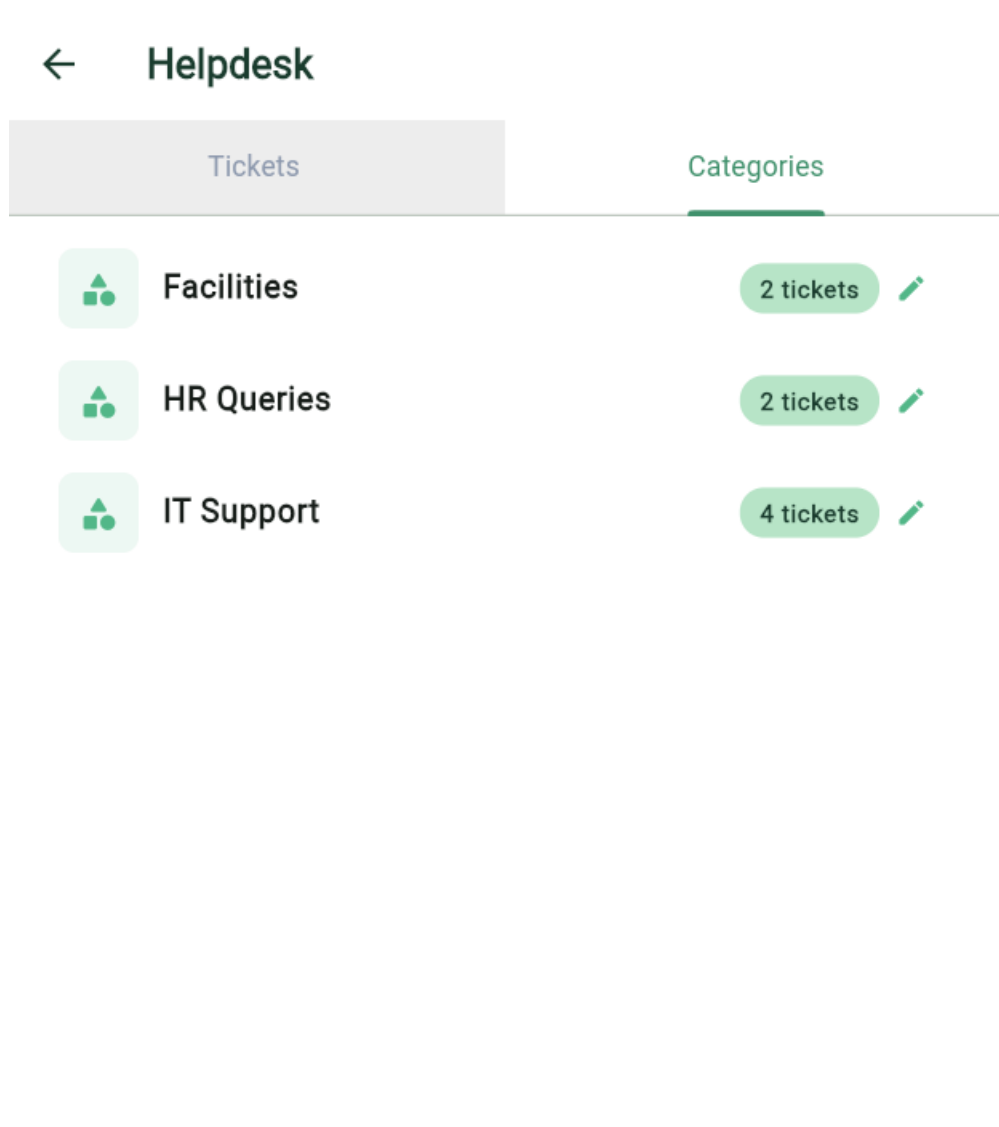


The Offboarding feature supports the employee separation process by organizing and tracking all required exit procedures. Through the mobile application, users can monitor offboarding tasks, review clearance requirements, and ensure that all necessary steps are completed efficiently and in a timely manner.

Helpdesk

Figure 33: Mobile Helpdesk





The Helpdesk feature allows employees to submit support requests and track their status directly through the mobile application. It facilitates efficient communication between employees and support teams, ensuring that issues are addressed promptly and that users receive timely updates on their requests.

Results and Discussion

CoreHR set out to make HR management easier for both companies and their employees by giving them a single platform with the features they actually need. By the time the project was finished, the system had reached the goals that were set at the beginning, with results that go beyond what was originally planned in some areas.

The system has made daily HR work much faster. Things that used to take days, like getting a leave request approved or calculating an employee's payslip, now happen in seconds. Real-time communication through notifications keeps everyone in the loop without anyone having to send manual updates. The ability to clock in and out from a phone has made attendance tracking much more reliable than the paper sheets that used to be the standard.

From the employee side, the experience has improved a lot. Instead of waiting for someone in HR to answer their questions, they can ask the chatbot or check their own dashboard. They can see exactly how many leave days they have left, what their last payslip looked like, and how their goals are progressing. The mobile app means they can do all of this from anywhere, not just when they are at their desk.

From the manager and administrator side, the dashboard provides a clear picture of the whole organization at a glance. Pending approvals are visible immediately. Reports give insight into attendance trends, payroll costs by department, and recruitment activity, all with charts that make patterns easy to spot. The smart insights engine even surfaces things that managers might have missed, like employees absent for several days in a row or contracts about to expire.

The project tackled a number of real challenges along the way. Building a system that needed to feel consistent across web and mobile, while supporting two languages and a complex permission model, was not simple. The technical decisions, particularly around the permission system, the AI features, and the offline-friendly mobile architecture, all required careful planning and testing. Some problems, like the Flutter web rendering issues and the field name mismatches between the mobile app and the backend, took longer to solve than expected.

Looking forward, there is still a lot that could be improved or added. The database could be migrated from SQLite to PostgreSQL to support larger organizations with more concurrent users. The polling-based notification system could be replaced with WebSockets for instant delivery. The chatbot, which currently uses keyword matching, could be connected to an actual large language model like Claude for natural language understanding. Document management could be added so

employees can upload contracts, ID copies, and certificates. Email integration could send automated notifications for things like leave approvals and contract expirations.

Overall, CoreHR demonstrates that it is possible for a small team, even a single developer, to build a serious HR platform using open-source technologies. The system is not just an academic exercise. It is something that could realistically be deployed in a small or medium organization tomorrow and provide real value from day one.

Conclusion

CoreHR was developed to be a complete Human Resource Management System that handles the full employee lifecycle through 12 integrated modules, gives users a modern web interface with bilingual support, provides a dedicated mobile app with role-specific experiences, and enforces a strong security model with granular permissions at every level. By the end of the project, all of these goals were met.

The project shows that it is genuinely possible to build a production-quality HR platform using free, open-source technologies. The combination of Node.js, React, Flutter, and SQLite turned out to be a powerful but lightweight stack that can be deployed on minimal infrastructure while still offering features that compete with commercial platforms costing thousands of dollars per year.

One of the most important achievements is the permission system. By going beyond simple role-based access and offering 32 individual permissions that can be customized per user, CoreHR makes it possible for organizations to apply the principle of least privilege properly. Each user gets access to exactly what they need and nothing more. The fact that these permissions are enforced at three independent levels, the backend API, the web frontend, and the mobile app, gives the system real defense in depth against unauthorized access.

The AI features, even without using large language models, show what intelligent automation can do for HR work. The smart insights engine surfaces information that would otherwise be buried in data. The chatbot answers questions instantly that would normally require contacting HR. The smart leave suggestions help employees make better choices about their time off. These features add a layer of intelligence that sets CoreHR apart from more traditional HR systems.

The mobile app is another major achievement. It is not just a smaller version of the web interface. It is a thoughtful experience designed specifically for mobile use, with different screens and navigation depending on whether the user is an employee or a manager. The employee dashboard with its clock status, leave balance, and goals creates a real personal HR hub that an employee can carry around in their pocket. The manager dashboard provides oversight of the organization on the go.

In the end, CoreHR proves that with the right tools and a clear vision, one developer can build a system that addresses real problems in HR management. This is more than a graduation project. It is a foundation for something that could actually be used in real organizations to improve how they handle their most important resource: their people.