

EMPLOYEE MANAGEMENT SYSTEM

A Web-Based HR Automation Platform

PRATHMESH PARAG DHOMANE

Guided By – MR. S. S. BHIDE

MCA Department, P. E. S. Modern College of Engineering, Pune, India

ABSTRACT

Human resource management in growing organizations often suffers from inefficiencies rooted in manual, paper-based processes — leading to data inconsistencies, payroll errors, and delayed decision-making. To address these challenges, the Employee Management System (EMS) is developed as a comprehensive, role-based, web-based HR automation platform built using the Django framework and Python.

EMS centralizes all HR operations — including employee onboarding, attendance tracking, leave management, payroll processing, task assignment, performance appraisal, and notification management — under a single unified digital platform. The system supports three distinct user roles: Admin, HR Manager, and Employee, each with fine-grained access controls. A relational SQLite3 database stores all records with integrity and consistency, while Django's Model-View-Template (MVT) architecture ensures a clean, scalable codebase.

The platform provides analytical dashboards with real-time graphical metrics, role-based redirection upon login, exportable PDF reports for attendance and performance appraisals, and automated payroll computation. All twelve test cases executed during system validation passed successfully.

Keywords: Employee Management, Django, Python, Payroll Automation, Attendance Tracking, Role-Based Access Control, Leave Management, Performance Appraisal, HR Dashboard

INTRODUCTION

Efficient human resource management is a cornerstone of organizational productivity. As companies grow, the complexity of managing employee records, attendance logs, leave approvals, payroll disbursement, and performance evaluations increases exponentially. Organizations that rely on manual processes — physical registers, spreadsheets, and printed memos — face escalating risks of data redundancy, calculation errors, unauthorized access, and operational delays.

The Employee Management System addresses these challenges by providing a centralized digital platform that automates all core HR workflows. By eliminating the need for physical record-keeping, the system ensures that employee data is always accurate, accessible, and secure. HR managers can approve leaves with a single click, payroll is computed automatically from salary components, attendance is tracked in real time, and management gains instant visibility through an analytical dashboard.

Developed using Django 4.x (Python) with an SQLite3 database for development and PostgreSQL recommended for production, EMS operates as a browser-accessible web application requiring no client-side installation. The system serves three user types — Admin, HR Manager, and Employee —

through tailored dashboards and access rights. The project was developed as part of an industrial internship at TechQuadra Software Solutions, Nagpur, providing real-world exposure to full-stack web development and software engineering principles.

LITERATURE REVIEW

- 1. Sommerville (2016):** Outlined key principles of software engineering including requirement analysis, feasibility study, system design, and modular development. EMS follows these principles rigorously, from DFD and ERD creation to black-box testing.
- 2. Pressman (2014):** Emphasized the role of structured SDLC methodologies in producing maintainable and scalable software. EMS adopts a phased development approach aligned with Pressman's practitioner model.
- 3. Date (2004):** Introduced normalization and relational database theory. EMS's data dictionary and table specifications reflect normalized database design with properly constrained foreign key relationships.
- 4. Silberschatz, Korth & Sudarshan (2019):** Provided foundational database system concepts applied in designing EMS's nine relational tables — Employee, Department, Attendance, LeaveRequest, Payroll, Task, Performance, Notification, and SystemSettings.
- 5. Django Software Foundation (2024):** Django's ORM, MVT architecture, session management, and template rendering engine form the backbone of EMS's backend. Built-in security features (CSRF protection, password hashing via pbkdf2_sha256) are directly utilized.
- 6. Fowler (2004):** UML diagrams guide EMS's system design — Use Case, Class, Sequence, Activity, Component, and Deployment diagrams are produced as documentation artifacts.
- 7. HR Technology Research (General):** Studies on existing HR tools such as SAP HCM, Zoho People, and BambooHR highlight high licensing costs and steep learning curves as key pain points for SMEs. EMS resolves these using entirely open-source technologies at zero licensing cost.
- 8. ReportLab (2024):** The ReportLab PDF generation library is used within EMS to produce downloadable attendance reports and performance appraisal PDFs.

SYSTEM ARCHITECTURE

EMS follows a modular, layered architecture based on Django's Model-View-Template (MVT) pattern, ensuring clean separation of concerns, maintainability, and scalability.

1. Presentation Layer

Built with HTML5, CSS3, and Vanilla JavaScript, this layer provides responsive dashboards for all three roles, role-based navigation sidebars, dynamic form rendering via Django Template Language (DTL), and client-side feedback including error messages and confirmations.

2. Application Logic Layer

Implemented in Python 3.x using Django 4.x, this layer handles session-based authentication and role verification, CRUD operations for all entities, payroll calculation (Basic, HRA = 25%, Allowance = 15%, Deduction = 10%, Net Pay), leave lifecycle management, task status transitions, performance appraisal computation, and notification dispatch.

3. Data Access Layer

Django's ORM (app.models) abstracts database operations across nine models: Employee, Department, Attendance, LeaveRequest, Payroll, Task, Performance, Notification, and SystemSettings. Foreign key constraints enforce relational integrity throughout.

4. Data Layer

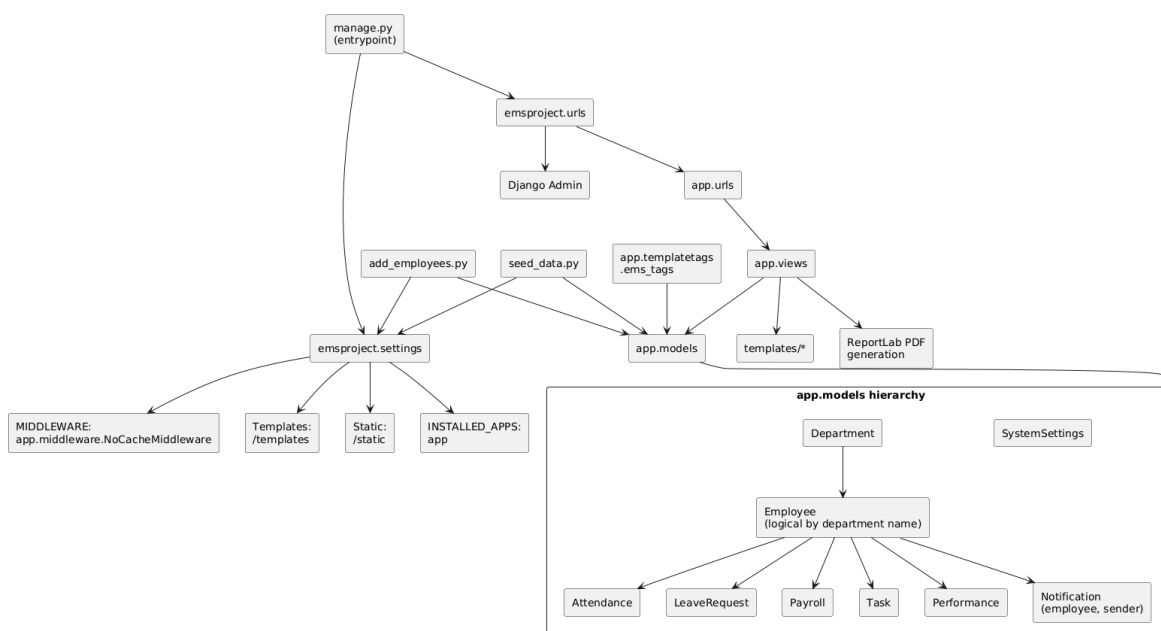
SQLite3 is used during development — lightweight and zero-configuration. PostgreSQL is recommended for production environments requiring concurrent multi-user access and enterprise-grade durability.

5. Deployment Architecture

Development: Django built-in server on localhost:8000. Production: Nginx (Reverse Proxy) → Gunicorn (WSGI) → Django Application → PostgreSQL, hosted on Linux (Ubuntu/Windows Server).

MODULE SPECIFICATIONS

EMS is structured into eleven functional modules organized in a hierarchical architecture:

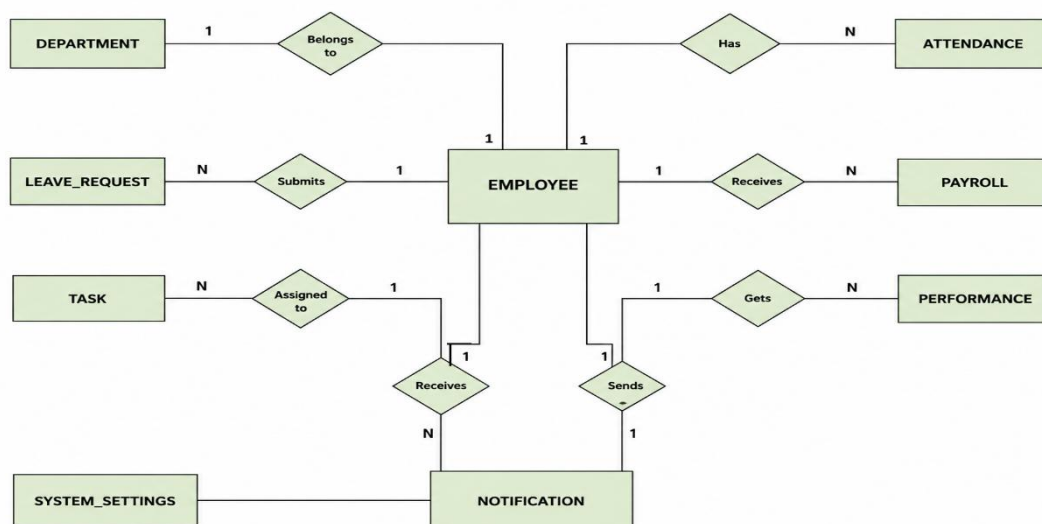


Module	Description	Key Functions
Authentication	Login, register, logout with session management and attempt limiting	login_page(), login_submit(), logout_view()
Employee Management	Full CRUD on employee profiles with role and status tracking	employee_list(), employee_add(), employee_edit(), delete_employee()
Department Management	Create/update/delete departments with headcount and open positions	department_list(), add_department(), edit_department()
Attendance	Mark daily attendance with time-in/out; export monthly PDF report	attendance(), mark_attendance(), export_attendance_report()
Leave Management	Submit, approve, and reject leave requests with status tracking	leave_requests(), add_leave(), update_leave_status()
Payroll	Automated monthly payroll computation and payment status management	payroll(), run_payroll(), update_payroll_status()
Task Management	Assign and track tasks with priority levels and Kanban-style progress	tasks(), add_task(), update_task_status(), delete_task()
Performance	Quarterly appraisals with self-review, manager review, and PDF export	performance(), start_review(), generate_appraisal()
Notifications	Send targeted in-app notifications (General, Salary, Project)	notifications(), send_notification()
Dashboard	Real-time HR analytics with charts for headcount, attendance, leaves, payroll	Role-specific admin/HR/employee dashboard views
System Settings	Company-wide config: currency, timezone, 2FA, audit log toggles	settings_view(), update_settings()

DATA DESIGN

Entity Relationship Diagram

The ERD captures nine core entities and their relationships. Employee is the central entity, linked to Department (N:1), Attendance (1:N), LeaveRequest (1:N), Payroll (1:N), Task (1:N), Performance (1:N), and Notification (1:N as both recipient and sender). SystemSettings maintains a global configuration relationship with Notification.



Data Dictionary — Employee Entity (app_employee)

Field	Type	Constraint	Description
emp_id	VARCHAR(20)	NOT NULL, UNIQUE	Human-readable ID (e.g., EMP-001)
email	VARCHAR(200)	UNIQUE, NOT NULL	Login credential
password	VARCHAR(200)	NOT NULL	pbkdf2_sha256 hashed password
role	VARCHAR(100)	NOT NULL	Admin / HR Manager / Employee
salary	DECIMAL(10,2)	DEFAULT 0	Monthly salary in INR
department	VARCHAR(100)	NOT NULL	Department name
status	VARCHAR(50)	NOT NULL	Active / On Leave / Remote / Inactive
joined_date	DATE	AUTO	Date of joining (auto-populated)

Data Dictionary — Payroll Entity (app_payroll)

Field	Type	Description
basic	DECIMAL(10,2)	Employee base salary
hra	DECIMAL(10,2)	House Rent Allowance (25% of basic)
allowance	DECIMAL(10,2)	Additional allowances (15% of basic)
deduction	DECIMAL(10,2)	Tax/PF deductions (10% of basic)
net_pay	DECIMAL(10,2)	Net = basic + hra + allowance - deduction
status	VARCHAR(20)	Paid / Pending / Processing

WORKING OF THE SYSTEM

1. User Authentication Flow

Users register with name, email, password, and role. On login, the system checks for an existing session and redirects to dashboard if found; otherwise it validates the attempt count (max 5 failures), verifies email in the Employee table, confirms the hashed password, validates the role tab selection against the assigned role, creates a session with user_id, user_role, user_name, and user_initials, then redirects to the appropriate role dashboard.

2. Leave Request Workflow

Employees navigate to /leave/, fill in leave type (Sick / Casual / Earned), from/to dates, and reason. The system validates that to_date >= from_date, calculates the number of days, creates a LeaveRequest record with status Pending, and returns a success response. HR Managers can then approve or reject pending requests from the same interface.

3. Payroll Processing Workflow

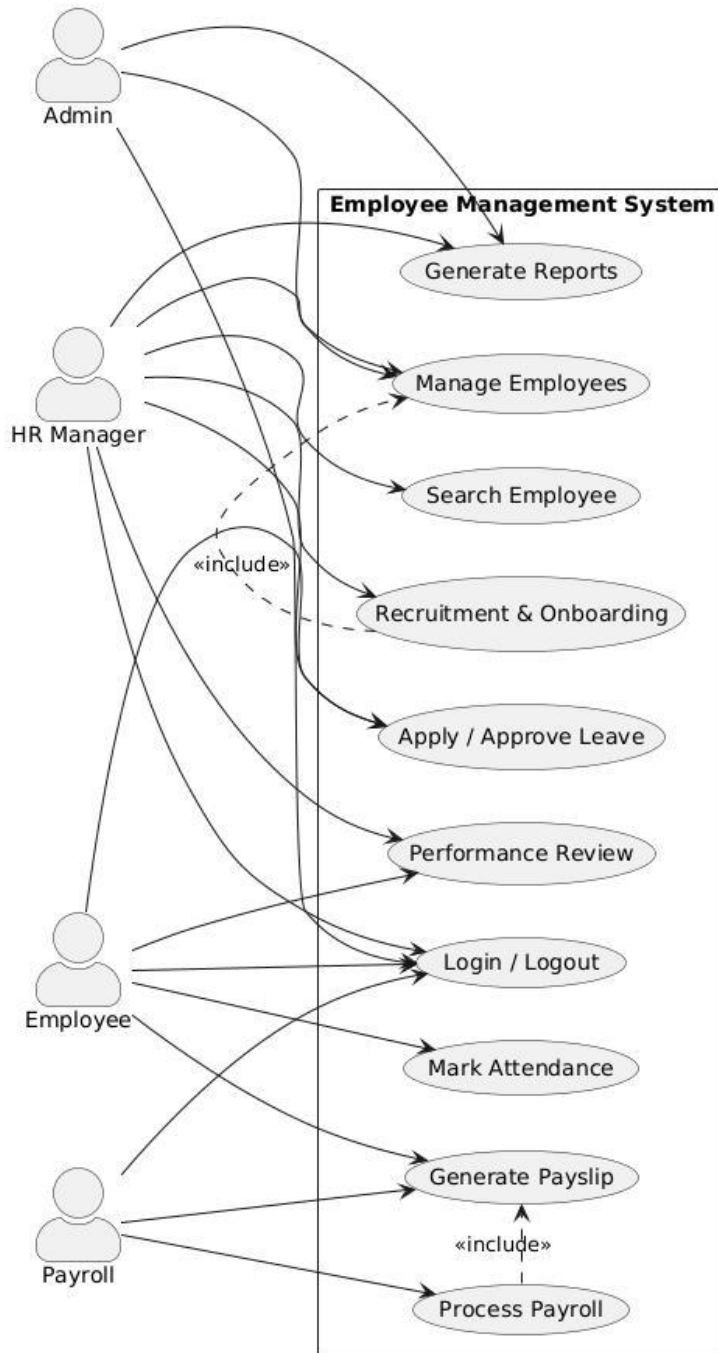
When Admin navigates to /payroll/ and runs payroll for an employee, the system fetches employee.salary as basic, then computes: HRA = basic × 25%, Allowance = basic × 15%, Deduction = basic × 10%, and Net Pay = basic + HRA + Allowance - Deduction. If a Payroll record already exists for the employee and month, the existing record is returned unchanged. If new, a record is created with status Pending. Admins can later update the payment status to Paid, Processing, or Pending.

UML DIAGRAMS

Use Case Diagram

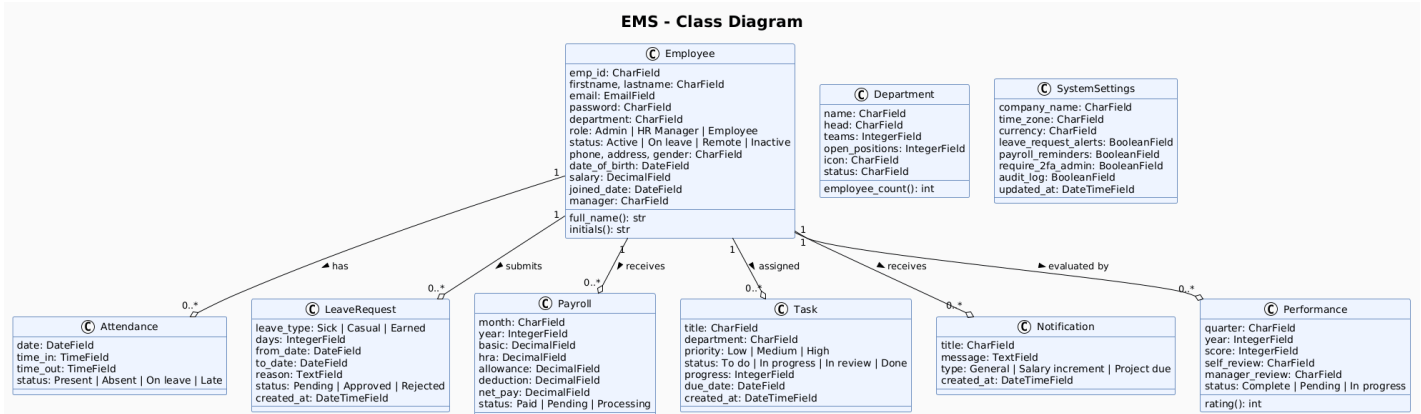
The Use Case Diagram shows the interaction of three actors — Admin, HR Manager, and Employee — with the system. Admin accesses all use cases. HR Manager handles employee management, attendance, leave approval, task assignment, and performance reviews. Employee accesses login/logout, mark attendance, apply leave, view payslip, and view tasks. A Payroll sub-system includes the Generate Payslip and Process Payroll use cases.

Employee Management System - Simple Use Case Diagram



Class Diagram

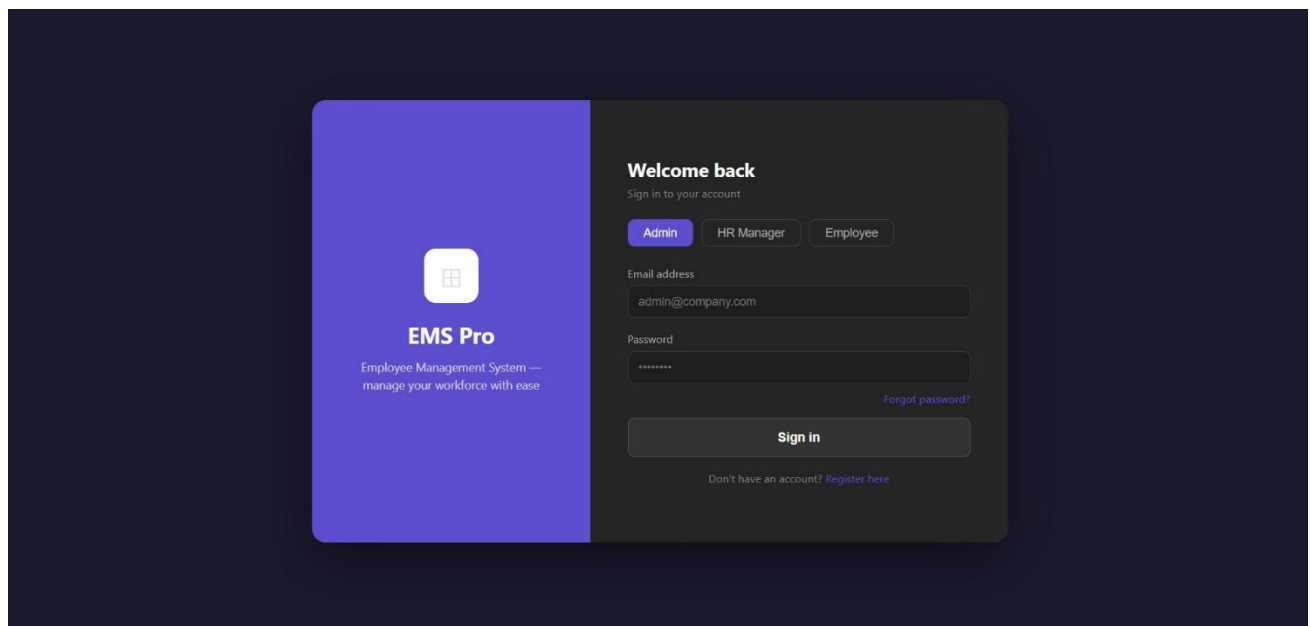
The Class Diagram represents all nine Django model classes — Employee, Department, Attendance, LeaveRequest, Payroll, Task, Notification, Performance, and SystemSettings — with their fields, data types, and association lines (has, submits, receives, assigned, evaluated by).



SYSTEM SCREENS

Login Page

The login screen presents a role-selector tab (Admin / HR Manager / Employee), email and password fields, a Sign In button, and a Register link for new users.



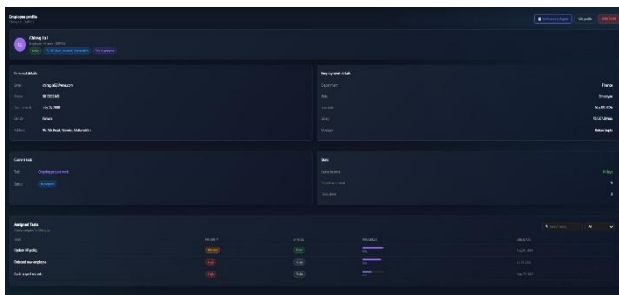
Admin Dashboard

Post-login, Admins see key HR metrics: total employees (80), present today (63), on-leave count (3), and payroll pending (16), with quick-action buttons (Add Employee, Approve Leaves, Run Payroll, View Tasks), a Department Distribution pie chart, and an Attendance Trend line chart for the last 7 days.



Employee List & Profile Page

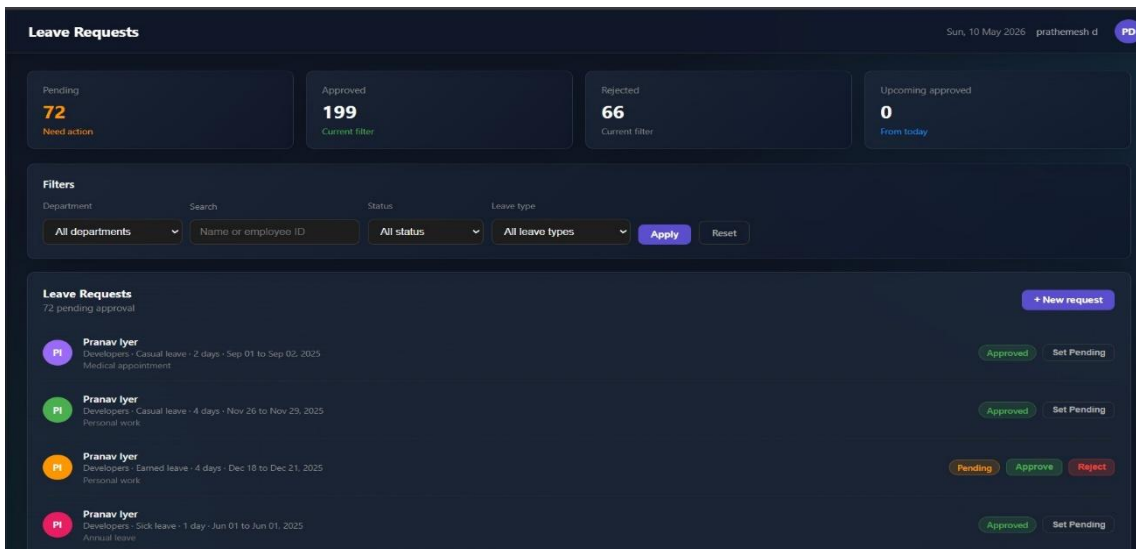
The Employees page shows a searchable, filterable table of all employees with columns for Name, ID, Department, Role, Current Task, Task Status, Employment Status, Joined Date, and Actions (View / Edit / Delete). Metric cards at the top summarize Total (80), Active (63), On Leave (3), and Remote (14).



ID	Name	Department	Role	Current Task	Task Status	Employment Status	Joined Date	Actions
1	Pranav Iyer	Developers	Software Engineer	Developing new feature	In Progress	Active	2023-01-15	View Edit Delete
2	Pranav Iyer	Developers	Software Engineer	Testing new feature	Completed	Active	2023-01-15	View Edit Delete
3	Pranav Iyer	Developers	Software Engineer	Reviewing code	Pending	Active	2023-01-15	View Edit Delete
4	Pranav Iyer	Developers	Software Engineer	Implementing bug fix	In Progress	Active	2023-01-15	View Edit Delete
5	Pranav Iyer	Developers	Software Engineer	Designing new feature	On Hold	Active	2023-01-15	View Edit Delete

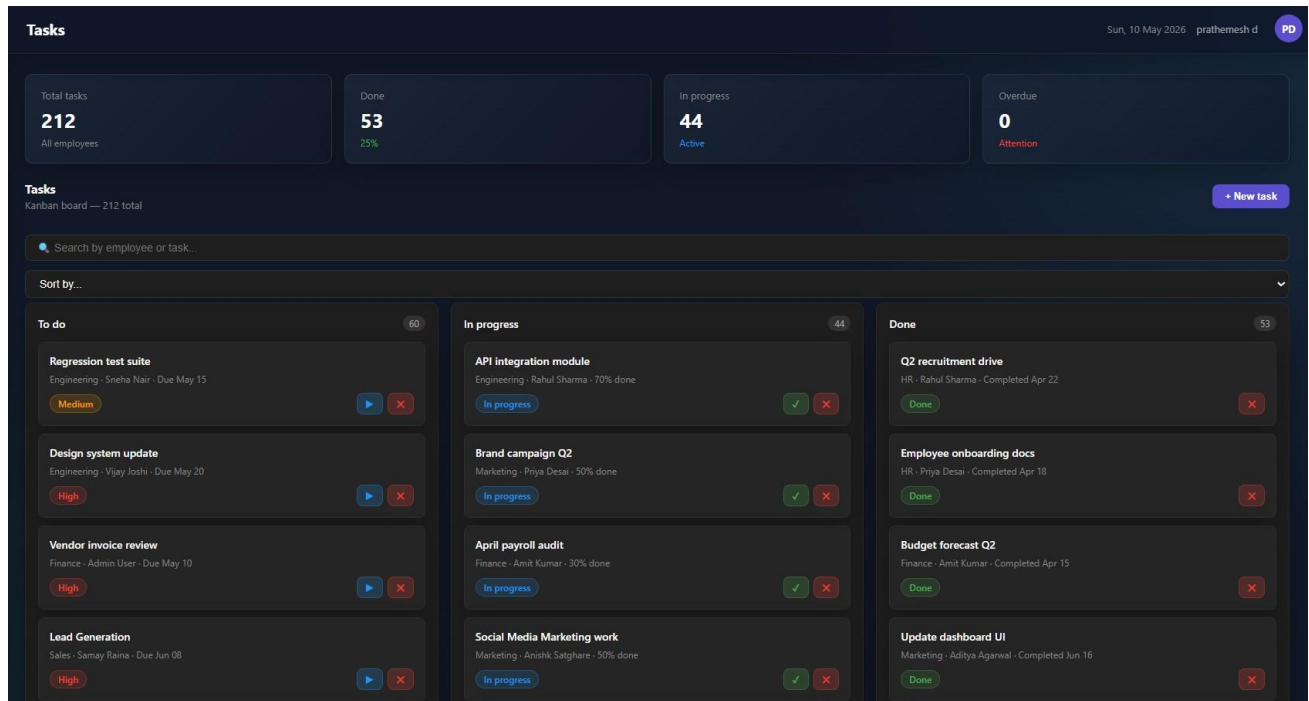
Leave Request Management

The Leave Requests page displays summary counters (Pending: 72, Approved: 199, Rejected: 66), filter controls by department, status, and leave type, and a list of requests with Approve / Reject action buttons per entry.



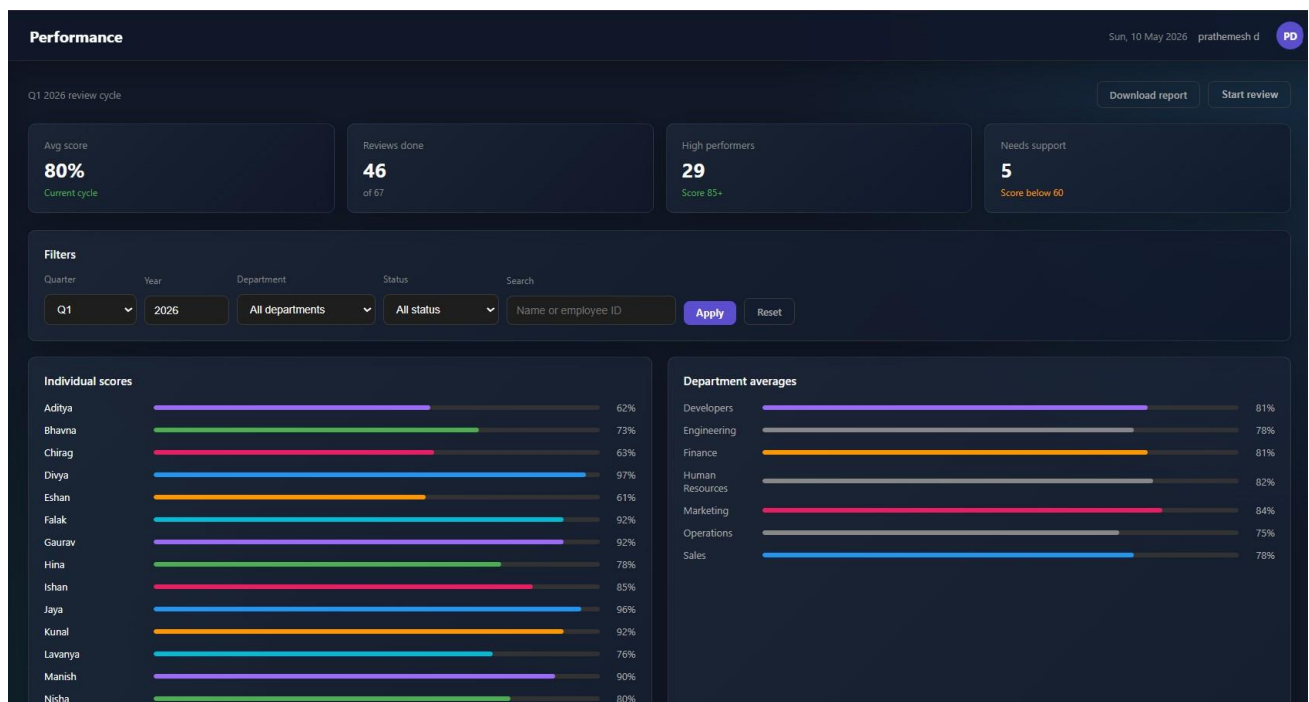
Task Management (Kanban Board)

Tasks are displayed in a Kanban-style board with columns: To Do (96), In Progress (44), and Done (53). Each card shows the task title, assigned employee, department, priority badge (Low / Medium / High), and a progress percentage indicator.



Performance Appraisal Page

The Performance page shows average score (80%), review completion (46 of 87), high performers count (29 with score ≥ 83), and needs-support count (5 below 80), with individual employee score bars and department average comparison bars.



Payroll Module

The Payroll page lists all employees with Basic, HRA, Allowance, Deduction, Net Pay, and Payment Status. A Run Payroll button opens a modal to select an employee and process their salary. Total payroll for May 2026 is shown as ₹40,15,513.

Payroll Sun, 10 May 2026 prathmesh d PD

May 2026 Run payroll

Total payroll: ₹4015513 (May 2026)

Paid: 45 (Processed)

Pending: 16 (Awaiting)

Avg salary: ₹55771 (Per month)

Payroll summary — May 2026

EMPLOYEE	DEPT	BASIC	HRA	ALLOWANCE	DEDUCTION	NET PAY	STATUS	PAYSLIP
RS Rahul Sharma	Developers	₹75000	₹18750	₹11250	₹7500	₹97500	Paid	Download
PD Priya Desai	HR	₹65000	₹16250	₹9750	₹6500	₹84500	Paid	Download
AK Amit Kumar	Finance	₹55000	₹13750	₹8250	₹5500	₹71500	Pending	Mark Paid
SN Sneha Nair	Marketing	₹70000	₹17500	₹10500	₹7000	₹91000	Pending	Mark Paid
VJ Vijay Joshi	Developers	₹52000	₹13000	₹7800	₹5200	₹67600	Paid	Download
AA Aditya Agarwal	Marketing	₹23878	₹11939	₹7163	₹3042	₹39938	Processing	Download
BC Bhavna Chavan	Operations	₹29537	₹14768	₹8861	₹4425	₹48741	Processing	Download
CL Chirag Lal	Finance	₹25604	₹12802	₹7681	₹3398	₹42688	Paid	Download

REPORTS

EMS generates three exportable reports:

1. Attendance Report (PDF): Monthly summary per employee showing Employee Name, Time In, Time Out, and Status (Present / Absent / Late / On Leave). Includes header with date, total records, and status breakdown.

Attendance Report

Date: 07 May 2026
 Generated on: 10 May 2026, 08:09 PM
 Total records: 67
 Present: 56 | Absent: 3 | On leave: 4 | Late: 4

Employee	Time In	Time Out	Status
Aakash Trivedi	08:17 AM	06:52 PM	Present
Aditya Agarwal	08:58 AM	06:53 PM	Present
Ankit Sharma	08:08 AM	06:11 PM	Present
Aryan Pandey	09:41 AM	06:44 PM	Present
Bhavna Chavan	09:26 AM	05:00 PM	Present
Bhumi Dubey	-	-	Absent
Chetan Joshi	-	-	On leave

2. Payroll Report: Monthly payroll summary table with Basic, HRA, Allowances, Deductions, Net Pay, and Payment Status per employee — generated from the Payroll module.

3. Performance Appraisal Report (PDF): Annual/quarterly report showing Organisation Summary (Department, Employees, Avg Score) and individual appraisal data with competency ratings, review status, and overall score out of 100.

EMS Pro
Annual Performance Report — 2026
 Generated on: 10 May 2026, 08:11 PM
 Total Employees: 80

Organisation Summary

Department	Employees	Avg Score
Developers	17	66/100
HR	3	29/100
Finance	14	77/100
Marketing	14	79/100
Sales	8	75/100
Operations	10	75/100
Engineering	11	77/100
Human Resources	3	86/100

TESTING

The system was validated using Black Box Testing and Functional Testing approaches across all modules. Both positive (valid input) and negative (invalid input) test cases were executed. All 12 test cases passed successfully.

TC No.	Module	Test Scenario	Expected Output	Result
TC01	Login	Valid credentials	Redirect to Dashboard	Pass
TC02	Login	Invalid password	Error: Invalid credentials	Pass
TC03	Employee	Add with all valid fields	Employee record created	Pass
TC04	Employee	Duplicate email entry	Error: Email already exists	Pass
TC05	Attendance	Mark attendance for today	Attendance record saved	Pass
TC06	Leave	Submit leave request	Leave with Pending status	Pass
TC07	Leave	Approve leave request	Status updated to Approved	Pass

TC No.	Module	Test Scenario	Expected Output	Result
TC08	Payroll	Run payroll for a month	Payroll records generated	Pass
TC09	Task	Add and assign task	Task created and assigned	Pass
TC10	Performance	Generate appraisal report	Appraisal PDF generated	Pass
TC11	Notification	Send notification to employee	Notification visible to recipient	Pass
TC12	Settings	Update system settings	Settings saved and reflected	Pass

APPLICATIONS

1. Small and medium enterprises (SMEs) replacing spreadsheet-based HR processes
2. IT companies managing distributed development teams
3. Educational institutions managing faculty and administrative staff
4. Manufacturing organizations with large, shift-based workforces
5. Startups requiring low-cost HR infrastructure with immediate deployment
6. NGOs and non-profit organizations with volunteer and staff management needs

ADVANTAGES

- Centralized Data Management — Single relational database eliminates redundancy and inconsistency
- Role-Based Access Control — Three-tier access (Admin / HR Manager / Employee) ensures data privacy
- Zero Licensing Cost — Built entirely on open-source technologies (Django, Python, SQLite/PostgreSQL)
- Automated Payroll — Formula-driven computation eliminates manual calculation errors
- Real-Time Analytics — Instant HR metrics via graphical dashboard without manual report generation
- PDF Report Export — Attendance and performance reports downloadable in standard PDF format
- Browser-Based Access — No client-side installation required; cross-platform (Windows, macOS, Linux)
- Scalable Architecture — Django + PostgreSQL setup supports hundreds of concurrent users in production
- Audit Logging — System settings include configurable audit log for all admin actions
- Modular Design — Each HR function is an independent module, simplifying maintenance and extension

LIMITATIONS

- No native mobile application; browser-dependent access on smartphones
- Biometric device integration for auto-attendance marking is not yet implemented
- Email and SMS notification dispatch (external channels) not included in the current version
- No multi-company or multi-branch support in the current version
- Tax computation follows a simplified formula; Indian income tax slab calculations (Form 16, EPFO) are not automated
- Real-time WebSocket-based push notifications are not implemented; users must refresh for updates
- The Applicant Tracking System (recruitment module) is planned but not yet developed

CONCLUSION

The Employee Management System successfully demonstrates how open-source web technologies can replace inefficient manual HR processes with a robust, automated, and user-friendly digital platform. Developed using Django 4.x and Python 3.x, the system automates eleven core HR functions — from employee onboarding to quarterly performance appraisals — under a single web application accessible through any modern browser.

The centralized relational database eliminates data redundancy and ensures consistency across all HR operations. Role-based access control enforces data privacy and operational security. Automated payroll computation, real-time attendance tracking, digital leave approval workflows, task management with priority tracking, and exportable PDF reports collectively represent a significant improvement over paper-based systems.

All twelve system test cases passed successfully, validating both functional correctness and graceful error handling. The project was developed as an industrial training deliverable at TechQuadra Software Solutions, Nagpur, providing hands-on experience in full-stack web development, relational database design, software engineering documentation, and UML-based system modeling.

The system is production-ready with a recommended deployment stack of Nginx → Gunicorn → Django → PostgreSQL on Linux, and provides a strong foundation for future enhancements including mobile applications, biometric integration, AI-powered attrition prediction, REST API exposure, and multi-branch support.

FUTURE ENHANCEMENTS

7. Mobile Application — Native iOS/Android app using React Native or Flutter with GPS-based attendance for remote employees
8. Biometric Integration — Fingerprint/face recognition terminal integration for automatic attendance marking
9. Email & SMS Notifications — Automated alerts via Twilio (SMS) and SendGrid (email) for payroll, leaves, and performance reminders
10. AI-Powered HR Insights — Machine learning models for attrition risk prediction, top performer identification, and payroll forecasting
11. Employee Self-Service Portal — Document upload/download (payslips, offer letters, tax forms), policy library, and internal job postings

12. Multi-Company / SaaS Support — Tenant-level isolation enabling the system to serve multiple organizations
13. Recruitment Module (ATS) — Job postings, candidate tracking, interview scheduling, and direct onboarding integration
14. Enhanced Payroll with Tax — Automatic income tax slab computation, Form 16 generation, EPFO and ESI portal integration
15. REST API Layer — Django REST Framework (DRF) API for integration with SAP, Oracle HCM, Tally, and other ERP systems
16. Training & Development Module — Track certifications, courses, and generate Individual Development Plans from appraisal data

REFERENCES

- [1] Django Software Foundation. Django Documentation v4.2. <https://docs.djangoproject.com/en/4.2/> (Accessed: May 2026)
- [2] Python Software Foundation. Python 3.x Documentation. <https://docs.python.org/3/> (Accessed: May 2026)
- [3] SQLite Consortium. SQLite Documentation. <https://www.sqlite.org/docs.html> (Accessed: May 2026)
- [4] MDN Web Docs. HTML5 and CSS3 Reference. <https://developer.mozilla.org/> (Accessed: May 2026)
- [5] Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition. Addison-Wesley Professional.
- [6] Sommerville, I. (2016). Software Engineering, 10th Edition. Pearson Education.
- [7] Pressman, R. S. (2014). Software Engineering: A Practitioner's Approach, 8th Edition. McGraw-Hill Education.
- [8] Django REST Framework. <https://www.django-rest-framework.org/> (Accessed: May 2026)
- [9] Date, C. J. (2004). An Introduction to Database Systems, 8th Edition. Pearson Education.
- [10] Silberschatz, A., Korth, H. F., and Sudarshan, S. (2019). Database System Concepts, 7th Edition. McGraw-Hill Education.
- [11] ReportLab Inc. ReportLab PDF Generation Library. <https://www.reportlab.com/> (Accessed: May 2026)
- [12] TechQuadra Software Solutions. Company Profile and Internship Documentation, Nagpur, 2026.