***Project Report***

***Title:*** *HR MANAGEMENT SYSTEM (HRMS)*

Roll Number: 23K-0737, 23K-0838 (Group Leader)

Name(s): M. Hanzala Jamil, Arham Rasheed

* Introduction (Aim or Motivation):

The primary aim of the HR Management System (HRMS) is to automate and streamline the daily operations of the Human Resources department within an organization. Manual HR processes often lead to delays, human errors, data inconsistency, and overall inefficiency. To address these challenges, this project provides a centralized, secure, and efficient platform that manages various HR functions, including employee records, departments and designations, attendance tracking, payroll processing, leave management, complaint handling, recruitment workflows, and monthly HR reporting. The motivation behind developing this system is to enhance organizational productivity, improve data accuracy, and ensure smooth and transparent HR workflows across all departments.

* Background (Research & Project Selection):

During research, it was observed that many organizations, especially small and medium enterprises, still rely on:

* Paper-based HR files
* Manual attendance registers
* Basic spreadsheets
* Unstructured payroll calculations
* Lack of centralized employee data

These outdated methods cause:

* High chances of errors
* Lack of accurate data
* Time-consuming processing
* Difficulty in generating HR reports
* No proper tracking of leaves or payroll history

To address these gaps, an HRMS was selected as the project topic. The system was chosen because:

1. It is practical and used in real organizations.
2. It involves multiple interconnected modules (ideal for full-stack development).
3. It showcases database design skills, backend API development, and frontend UI creation.
4. It has clear objectives and measurable outcomes.

* Project Specification:

The HR Management System is developed using:

**Backend**

* Django 5
* Django REST Framework
* MySQL
* SimpleJWT Authentication
* Python 3.12.4

**Frontend**

* React 19
* Vite
* Redux Toolkit
* TailwindCSS
* React Router

**Functional Modules:**

| **Module** | **Description** |
| --- | --- |
| Employee Management | Add, update, delete, view employees |
| Departments & Designations | HR structure management |
| Attendance | Daily check-in/out, working hours |
| Leave Management | Leave requests, approvals, balance tracking |
| Payroll | Salary generation, history, deductions |
| Complaints | Internal HR complaints |
| Recruitment | Candidate tracking, interview status |
| Monthly Reports | Automated HR summaries |

* Problem Analysis:

Organizations face several HR challenges:

**→ Manual data entry errors**  
Human mistakes in attendance, salaries, and employee records.

**→ No centralized database**  
Hard to retrieve accurate information quickly.

**→ Slow payroll processing**  
Monthly salary generation requires repetitive manual work.

**→ Poor leave tracking**  
Mismanagement of employee leave requests and balances.

**→ Lack of reporting**  
HR reports (new hires, attendance summary, payroll summary) require a lot of manual effort.

**→ Recruitment tracking difficulties**  
Candidate applications cannot be tracked efficiently.

This project addresses all these problems through automation and digitization.

* Solution Design (Project Detail, Functionality & Features):

**System Architecture:**

* **Backend:** Django REST API
* **Frontend:** React UI
* **Communication:** JSON REST APIs
* **Database:** MySQL

**Core Features:**

**1. Employee Management**

* Add/Edit/Delete/View employees
* Assign departments & designations
* Maintain employment history

**2. Attendance System**

* Check-in / Check-out
* Auto calculate working hours
* Monthly attendance reports

**3. Leave Management**

* Employee leave requests
* HR approval/rejection
* Leave balance tracking

**4. Payroll System**

* Automatic salary calculation
* Allowances & deductions
* Payroll history for each month

**5. Recruitment**

* Add candidates
* Track interview process
* Update candidate status

**6. Complaint Management**

* Employee can submit a complaint
* HR reviews and resolves

**7. Monthly Report Generation**

* New hires
* Departures
* Attendance stats
* Payroll summary
* Implementation & Testing Phase:

**Implementation Steps:**

1. **Backend development**
   * Models created for all entities
   * REST APIs built using DRF
   * JWT authentication implemented
   * MySQL schema applied
   * Business logic for attendance, payroll, leaves
2. **Frontend development**
   * UI layout using React + Tailwind
   * Component development: tables, modals, forms
   * API structure created
   * Redux store setup
3. **Integration**
   * Testing API responses
   * Mapping frontend components to backend routes

**Testing Performed:**

* **API testing** (Postman)
* **UI testing** (manual testing)
* **Functional testing**
* **Validation testing**
* **Authentication testing**

All modules were tested to ensure accuracy, error handling, and correct data flow.

* Project Breakdown Structure (Workload distribution with timeline):

**Project Timeline:**

| **Phase** | **Duration** | **Tasks** |
| --- | --- | --- |
| Requirement Analysis | Week 1 | Identify modules, data flow, tools |
| Database Design | Week 2 | ERD, schema, relationships |
| Backend Development | Week 3–5 | Models, APIs, authentication |
| Frontend Development | Week 6–8 | UI pages, components, layouts |
| Integration | Week 9 | Connect frontend with backend |
| Testing | Week 10 | Unit, API, UI testing |
| Final Preparations | Week 11 | Optimization, final reviews |
| Final Report | Week 12 | Documentation |

**Workload Distribution:**

| **Member** | **Responsibilities** |
| --- | --- |
| Developer 1 | Backend API development, database schema, Testing, documentation, integration |
| Developer 2 | Frontend UI, React components, routing, Project management, system design, QA |

* Conclusion (Summary & Discussion):

The HR Management System successfully automates and simplifies HR workflows. The system provides:

* Centralized employee data
* Automated attendance and payroll
* Proper leave and complaint management
* Efficient recruitment handling
* Accurate monthly reporting

The project showcases strong implementation of full-stack development, combining Django REST APIs with a modern React frontend.  
The removal of the insurance module simplified the architecture while keeping the project focused on the core HR needs.

Overall, the HRMS significantly improves organizational efficiency, reduces human error, and modernizes the HR department’s operations.























































