WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Police Duty Rostering Management System Using MERN Stack and MongoDB

Mohmmad Asra Begum [1], K.Jayasri [2]

[1]Student, School of Informatics, Department of MCA, Aurora Deemed University, Hyderabad
[2]Assistant Professor, School of Engineering, Department of CSE, Aurora Deemed University, Hyderabad
[1]mohmmadasra1812@gmail.com, [2]jayasri@aurora.edu.in

ABSTRACT

Police departments handle numerous responsibilities, ranging from patrolling to special duty assignments. Traditionally, duty allocation is managed manually through registers and spreadsheets, which often results in errors, delays, and lack of transparency. To overcome these challenges, this paper proposes a Police Duty Rostering Management System (PDRMS) developed using the MERN (MongoDB, Express.js, React.js, Node.js) stack. The system enables administrators to efficiently allocate duties, track shifts, monitor resources, and generate reports, while police staff can access their duty schedules through a web interface. The system ensures fairness in rostering, reduces manual errors, and increases accountability. Architectural and data flow diagrams illustrate the system design, and frontend results show how the interface improves usability.

Keywords: Police Duty Management, Rostering System, MongoDB, MERN Stack, Duty Allocation, Scheduling

1. Introduction

Efficient police duty allocation is vital to maintain law and order in any community. With increasing responsibilities such as crime prevention, traffic management, VIP security, and emergency response, scheduling duties for police officers has become complex. Manual systems are prone to duplication, biased allocations, and inefficiencies.

A Police Duty Rostering Management System (PDRMS) provides a digital solution that automates the process of assigning duties and shifts. This system integrates modern web technologies to improve efficiency, fairness, and transparency. It allows the administration to dynamically schedule duties while ensuring that the workload is balanced. Officers can log in to check their assigned duties, apply for leave, or request shift changes.

This paper presents the design and implementation of a PDRMS using the MERN stack with MongoDB as the database, enabling a scalable, secure, and user-friendly solution.

2. Literature Review

Existing systems for police duty management are either manual or rely on basic desktop applications. Manual registers often result in:

- Difficulty in tracking leave and attendance.
- Lack of real-time updates for officers.
- High chances of human errors and favoritism in duty allocation.

Previous attempts at digital duty management have mostly used relational databases (MySQL/Oracle). However, these are less flexible for handling complex and dynamic rostering requirements. With the rise of NoSQL databases like MongoDB, systems can handle large amounts of unstructured duty and officer data more efficiently.

The MERN stack offers a full-stack JavaScript solution, making it easier to integrate frontend (React) with backend (Node.js, Express) while ensuring smooth communication with MongoDB. This approach improves speed, scalability, and maintainability.

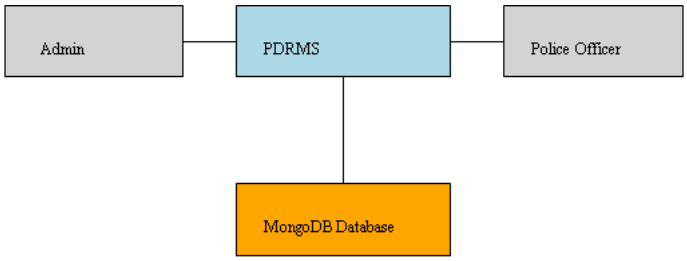
3. Proposed System

The proposed PDRMS automates duty scheduling and provides officers with an accessible platform.

Objectives:

1. Automate allocation of duties and shifts.
2. Provide a web-based interface for administrators and officers.
3. Maintain officer records including shifts, leaves, and performance.
4. Ensure transparency and reduce errors in scheduling.
5. Generate real-time duty reports and history logs.

3.1 System Architecture



Description:

- Frontend (React + Vite): Provides an interactive web interface for both administrators and police officers.
- Backend (Node.js + Express): Handles business logic, duty assignment algorithms, and API endpoints.
- Database (MongoDB): Stores officers, duties, shifts, attendance, and reports.
- Authentication: Secure login using JWT (JSON Web Token) for role-based access (Admin vs Officer).3.2 Data Flow Diagram (DFD)

Level 0 DFD: Shows the interaction between Admin, Officer, and System.

Level 1 DFD: Explains internal processes like duty assignment, leave request, and report generation.

4. System Design and Implementation

4.1 Database Design (MongoDB)

Collections include:

- Officers → officerID, name, rank, availability, leave status.
- Duties → dutyID, type (patrol, traffic, event), location, shift time.
- Roster → mapping of officerID to dutyID with shift timing.
- Reports → history of duty allocations and attendance.

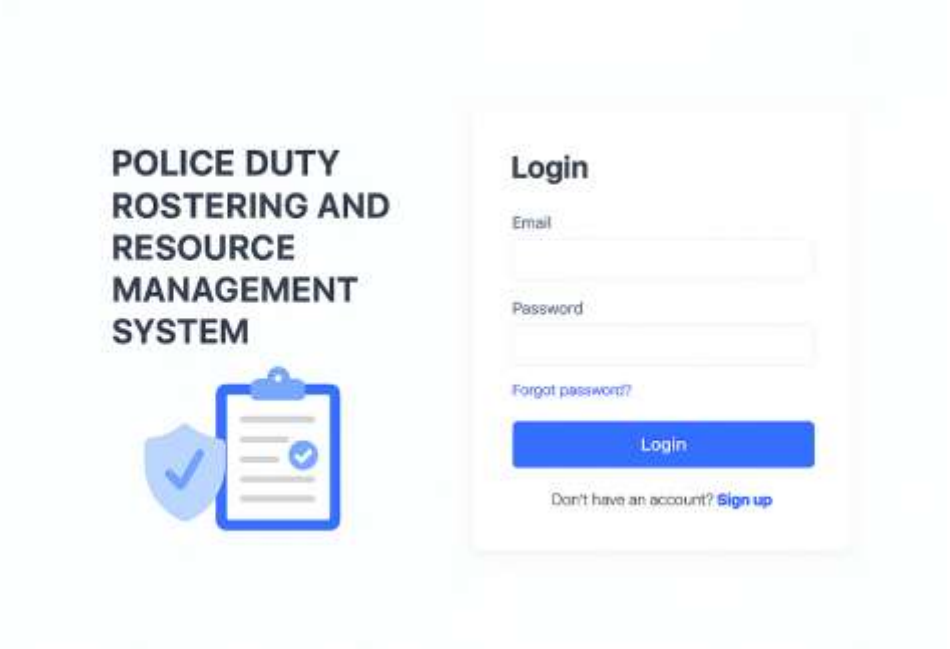
4.2 Backend (Express + Node.js)

- RESTful APIs for duty assignment, roster retrieval, and leave management.
- Middleware for authentication and logging.

4.3 Frontend (React + Vite)

- Admin Dashboard: Create/Edit duties, assign officers, generate reports.
- Officer Dashboard: View assigned duties, request leave, check history.
- Notifications: Duty reminders via dashboard alerts.

6. Frontend Results



Duty Allocation Form – Assigns an officer to a specific duty

Officer Shifts Schedule

For the Week of:

Department Name:

Monday	9:00 A.M.	11:00 A.M.	12:00 P.M.	1:00 P.M.	4:00 P.M.	Sick?
John W	Patrol	Patrol	Traffic Control	Desk Duty	Patrol	1
Sarah M	Traffic Control	Traffic Control	Desk Duty	Patrol	Traffic Control	7
David R	Desk Duty	Desk Duty	Patrol	Traffic Control	Desk Duty	7
Emily T	Patrol	Patrol	Traffic Control	Desk Duty	Patrol	7
Michael K	Patrol	Traffic Control	Patrol	Patrol	Sick	7

Wednesday	9:00 A.M.	11:00 A.M.	12:00 P.M.	1:00 P.M.	4:00 P.M.	Sick?
John W	Patrol	Patrol	Traffic Control	Desk Duty	Patrol	1
Sarah M	Traffic Control	Traffic Control	Desk Duty	Patrol	Traffic Control	7
Anna G	Sick	Sick	Sick	Sick	Sick	7

Example Interfaces:

1. Admin Dashboard – Displays officer list, duty slots, and allocation controls.
2. Duty Allocation Form – Assigns an officer to a specific duty.
3. Officer Dashboard – Shows upcoming duty schedule, leave requests, and past records.
4. Report Page – Generates PDF/Excel reports of duty allocation.

6. Results and Discussion

The developed system successfully eliminates challenges of manual rostering. Key outcomes include:

- Time Efficiency: Duties are assigned within seconds compared to manual registers.
- Fairness: Automated allocation prevents favoritism.
- Transparency: Officers can directly view duties without relying on verbal communication.
- Scalability: MongoDB allows storage of thousands of duty records efficiently.

7. Conclusion and Future Work

This paper presented a Police Duty Rostering Management System built on the MERN stack. The system streamlines duty scheduling, improves accountability, and provides user-friendly dashboards.

Future Enhancements:

- AI-based duty allocation using workload prediction.
- Mobile app integration for notifications.
- Integration with biometric attendance systems.

References

1. M. Stonebraker, "SQL Databases vs NoSQL Databases," *Communications of the ACM*, vol. 53, no. 4, 2010.
2. A. Kumar and S. Reddy, "Automated Duty Scheduling System for Public Safety Organizations," *International Journal of Computer Applications*, 2020.
3. MongoDB Inc., "MongoDB Documentation," [Online] Available: https://www.mongodb.com/docs/
4. Facebook Inc., "React – A JavaScript library for building user interfaces," [Online] Available: https://react.dev/