REPORT

Visitor Management System (VMS) Documents

By
Ram Krishna [BT23CSE026]
Lakshay Batra[BT23CSE023]

Semester: 5 **Branch: CSE**

Course Name: Database Management System [CLS 301]

Under the guidance of
DR. Rahul Semwal

Department of Computer Science and Engineering Indian Institute of Information Technology Nagpur - 441108 (India)

1. **Introduction:**

   Manually managing visitor information can be laborious, ineffective, and prone to mistakes in many organizations. The goal of this project, called Visitors Management System, is to use a database-driven application to digitize and expedite the process of recording and managing visitor information.

   Manual visitor logging is often inefficient. The "Visitors Management System" project addresses this using a modern web application built on **Supabase**. Leveraging its underlying **PostgreSQL** database, the system streamlines visitor registration, tracks check-in/check-out times, and maintains a secure digital log.

   The system features an admin login, visitor registration, and reporting tools. This browser-compatible solution is suitable for offices, schools, etc., requiring controlled access, and utilizes the Supabase platform for its backend database and hosting needs. The core strength lies in the robust PostgreSQL database managed by Supabase for effective visitor record management.

2. **Problem Statement:**

   What problem does the project solve?

   This project tackles the shortcomings of manual visitor logging, such as disorganization, significant time investment, and the inability to track visits in real-time. By implementing a web-based digital platform, it replaces outdated paper logbooks, enabling efficient registration, comprehensive management, and precise tracking of visitor data. The system's dynamism and dependability are further enhanced by features including visitor clearance protocols, departmental routing, and the capability to modify records.

3. Objective:
   - To enable visit record management and filtering through departmental or purpose of visit for grouping classification to simplify visitor record management.
   - To allow PHPMailer integration for alerting notifications on visitor confirmations and approvals.
   - To make the system accessible from any web browser by deploying it online through InfinityFree.
   - To ensure staff with little to no training can operate the system by maintaining a user friendly and simple interface.
   - To shift manual log book and paperwork registration into an automated process for easier registration flow.

4. Software Requirements:

- **Node.js** (JavaScript runtime for development and running the app)
- **Vite** (Frontend build tool and dev server)
- **React.js** (Frontend UI library)
- **Email JS** (Email API)
- **React Router** (Routing and navigation)
- **Tailwind CSS** (Styling the UI)
- **Zustand** (State management)
- **Lucide-react** (Icon library)
- **Web Browser** (Chrome, Firefox, Edge, etc.)
- **(Optional) Git** (Version control)
- **(Optional) VS Code** (Recommended code editor)
- **Supabase** (Postgres Development Platform)

5. How does the Model Work?

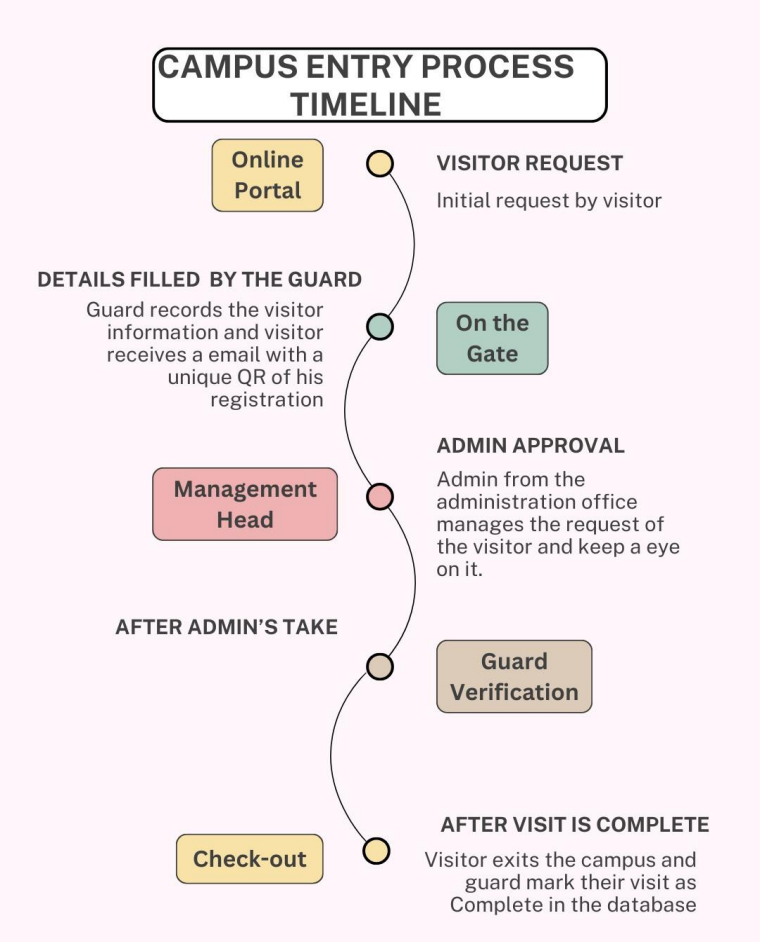
User Roles and Workflow

1. Visitor
    - Submit a request via college portal or to the guard on the visit:
        - Personal details(name, contact, e-mail)
        - Purpose of visit(meeting faculty, admission inquiry, etc)
        - Date/time of visit and leave.
    - Receive automated email with a unique QR code registration.

2. Admin
    - Review and approve/reject requests via email notification.
    - Manages staff/faculty/department access permission.

3. Staff/Faculty
    - Pre-register expected visitors (e.g. guest lectures, parent).
    - Receives notifications when their visitor arrives.

4. Process Flow



5. Key Features

1. Admin Dashboard
    - User Management: Add/remove staff, faculty, or departments.
    - Approval System: Bulk-approve/reject requests with comments.
    - Reports: Generate logs for audits (e.g., daily visitors, frequent guests).
    - Emergency Lockdown: Instantly revoke all visitor access if needed.

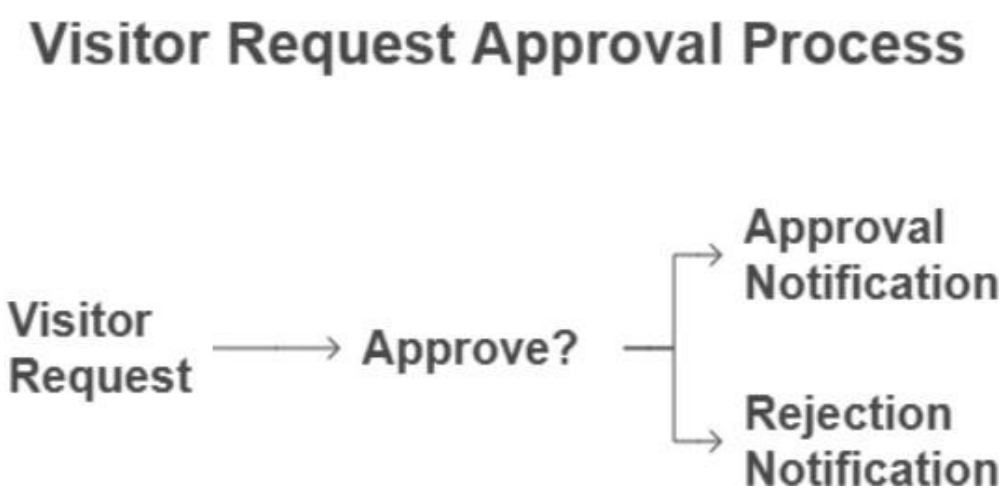
2. Automated Notification:
    - **Admin**: Alters for pending request.
    - **Staff**: Notifies when their visitor arrives.

3. Visitor Status Tracking
    - **Pending**: Request submitted, awaiting admin action.
    - **Approved**: entry permitted

- **Rejected**: Reason provided via SMS/email.

4. Approval Process
   - ●Visitor submits a request with details.
   - ●Admin receives SMS/email:
   - ●Admin action triggers SMS/email to visitor:
     a. Approved: Entry granted.
     b. Rejected: States reason.

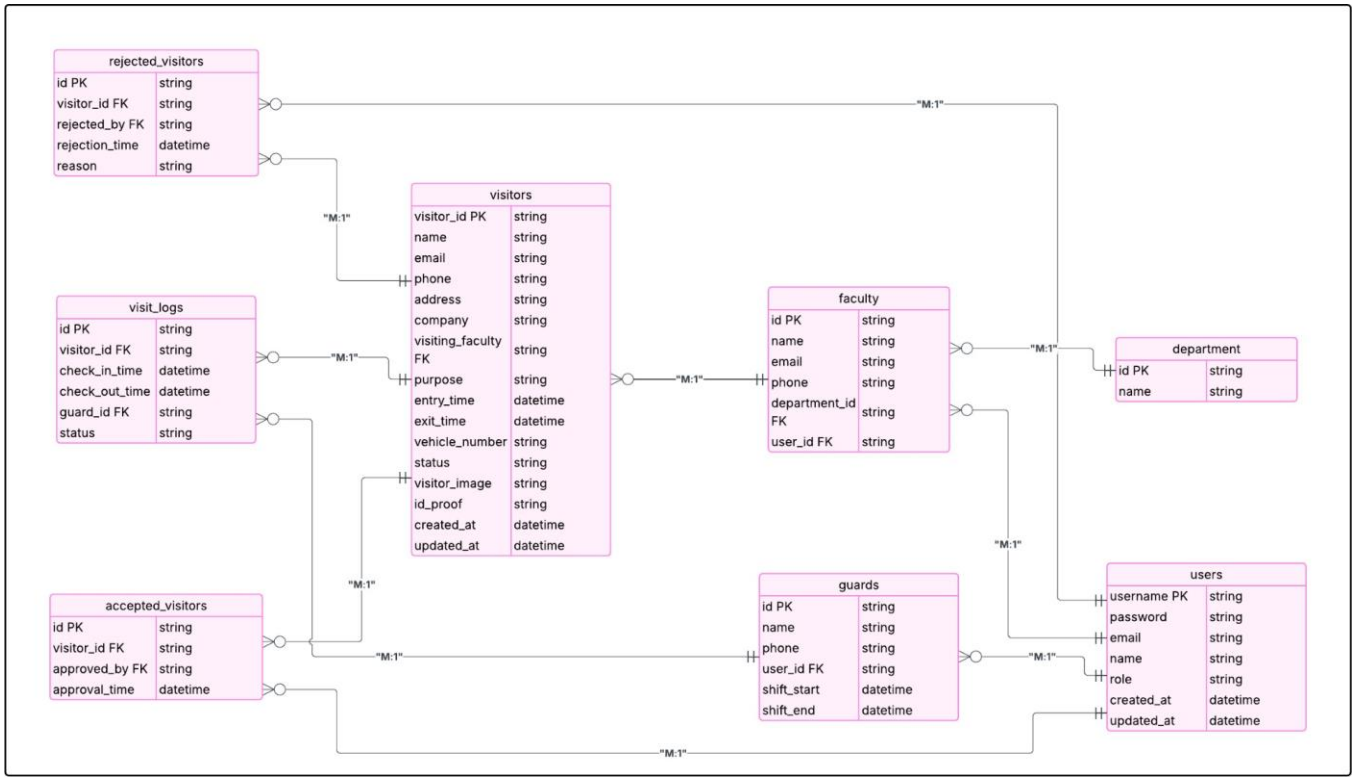
.



6. Benefits for IIIT Nagpur
- Safety: Mitigate risks of unauthorized access in a remote location.
- Efficiency: Reduce manual paperwork and waiting time.
- Accountability: Digital trail for all visitor activities.
- Scalability: Adaptable for future expansions (new departments, events).

Note: This system integrates seamlessly with IIIT Nagpur's existing infrastructure while prioritizing security and user convenience.

ER DIAGRAM



From Table	To Table	Relationship Type	Description / Key Used
`faculty`	`department`	Many-to-One (M:1)	`faculty.department_id` → `department.id`
`faculty`	`users`	Many-to-One (M:1)	`faculty.user_id` → `users.username`
`guards`	`users`	Many-to-One (M:1)	`guards.user_id` → `users.username`
`visitors`	`faculty`	Many-to-One (M:1)	`visitors.visiting_faculty` → `faculty.id`
`accepted_visitors`	`visitors`	Many-to-One (M:1)	`accepted_visitors.visitor_id` → `visitors.visitor_id`
`accepted_visitors`	`users`	Many-to-One (M:1)	`accepted_visitors.approved_by` → `users.username`
`rejected_visitors`	`visitors`	Many-to-One (M:1)	`rejected_visitors.visitor_id` → `visitors.visitor_id`
`rejected_visitors`	`users`	Many-to-One (M:1)	`rejected_visitors.rejected_by` → `users.username`