iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.80141

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue IV Apr 2026- Available at www.ijraset.com

Vaccination Management System

Rashmi Mishra[1], Prerna Sahane[2], Komal Yadav[3], Muskan Yadav[4], Prof. Pratiksha Shinde[5]
Department of Computer Science (AIML), Bharat College of Engineering, Maharashtra, India

Abstract: ***Managing vaccination records across a country's population is a complex logistical challenge, particularly when paper-based systems remain common. The Vaccination Management System (VMS) is a full-stack, web-based Vaccination Database Management System (VDBMS) built with Python Flask and MySQL that digitizes and streamlines the entire vaccination lifecycle. The system supports three distinct user roles — patient, administrator, and vaccine administrator — each with tailored dashboards and functionality. Core features include appointment scheduling, profile management for multiple family members, digital certificate generation with QR code verification, AI-powered chat support, vaccine inventory management, and real-time analytics. This paper presents the system design, relational database architecture, feature set, and implementation details of the Vaccination Management System, demonstrating how modern web technologies and AI integration can address real-world healthcare management challenges.***

Keywords: Vaccination Management System, Flask, MySQL, AI Chat Support, Digital Certificates, Role-Based Access Control, Healthcare Database, QR Code Verification, Appointment Scheduling

I. INTRODUCTION

Vaccination is one of the most effective tools in public health, yet managing vaccination records, schedules, and appointments remains an administrative burden for healthcare providers and patients alike. In many regions, paper-based records are prone to loss, duplication, and fraud. Meanwhile, patients often struggle to track their own immunization history or that of their family members.

The Vaccination Management System was built to address these challenges head-on. It is a comprehensive, web-based Vaccination Management System that brings the entire vaccination process online — from registration and appointment booking to certificate issuance and real-time AI support. The system was developed using Python Flask for the backend, MySQL for relational data storage, and a responsive HTML/CSS/JavaScript frontend.

The platform is live and publicly accessible, making it a practical, deployable solution rather than a purely academic exercise. This paper walks through the motivation, architecture, features, and implementation of the system, offering insights into how such a system can be adapted for broader healthcare contexts.

II. BACKGROUND AND MOTIVATION

Healthcare digitization has accelerated over the past decade, with Electronic Health Records (EHR) systems becoming standard in hospitals. However, vaccination-specific management software — especially at a community or outpatient level — is still lacking in accessibility and completeness. Existing public health portals are often government-siloed, difficult to integrate, or unavailable in developing regions.

Several studies have highlighted that missed vaccinations often result from poor record-keeping and lack of reminders [1]. Digital vaccination cards and smart scheduling systems have shown promise in improving compliance rates [2]. This system synthesizes these insights into a single, unified platform that is open-source, self-hostable, and extensible.

The integration of AI-powered conversational agents into healthcare portals is also a growing trend [3]. The Vaccination Management System incorporates real-time AI chat support via the Groq API, allowing users to ask questions about vaccines, side effects, and scheduling — reducing the burden on human staff.

III. SYSTEM DESIGN AND ARCHITECTURE

A. System Overview

The Vaccination Management System follows a three-tier architecture: a presentation layer (HTML/CSS/JS templates), a business logic layer (Flask routes and application logic), and a data layer (MySQL database with SQLAlchemy ORM). The system is designed to be deployed on cloud platforms and includes a Gunicorn WSGI server configuration for production use.

©IJRASET: All Rights are Reserved

B. User Roles and Access Control

The system implements role-based access control (RBAC) with three distinct user types:

- Patient / End User: Can register, manage family profiles, book and reschedule appointments, view vaccination history, download digital certificates, and interact with the AI chat assistant.
- System Administrator: Has full oversight of users, vaccination centers, inventory, schedules, and system-wide analytics.
- Vaccine Administrator: A specialized role focused on managing vaccine stock, updating inventory at specific centers, and overseeing appointment fulfillment at their assigned location.

C. Database Architecture

The relational database is implemented in MySQL and designed with normalization in mind. Key entities include Users, Profiles (family members linked to a user), Vaccines, Vaccination Centers, Schedules, Appointments, Certificates, and Inventory. Foreign key constraints enforce referential integrity throughout.

The database follows a third normal form (3NF) design to eliminate redundancy. An Entity-Relationship (ER) diagram and a full relational schema have been published alongside the codebase. The schema supports one-to-many relationships between users and profiles, and many-to-many relationships between vaccines and centers (mediated through inventory and schedule tables).

D. Technology Stack

Layer	Technology	Purpose
Backend	Python Flask	Web framework, routing, session management
Database	MySQL + SQLAlchemy	Relational data storage and ORM
Authentication	Flask-Login + Werkzeug	User sessions and password hashing
PDF Generation	ReportLab	Digital certificate creation
QR Code	qrcode library	Certificate verification codes
AI Chat	Groq API	Real-time conversational AI support
Frontend	HTML, CSS, JavaScript	Responsive user interface
Production Server	Gunicorn	WSGI server for deployment
Environment	python-dotenv	Secure configuration management

TABLE I
Vaccination Management System Technology Stack

IV. FEATURES AND FUNCTIONALITY

A. Patient-Facing Features

Patients begin by creating an account and setting up a profile. One of the system's most user-friendly features is support for multiple family profiles under a single account — a parent, for instance, can manage vaccination records for all their children without creating separate logins. Each profile tracks demographic details, vaccination history, and upcoming appointments independently.

Appointment booking is driven by real-time availability: the system checks vaccine inventory levels and existing schedule slots before confirming a booking. Patients receive a summary of their appointment and can reschedule or cancel with a few clicks. A vaccination history view provides a complete timeline of past immunizations, making it easy to share records with healthcare providers. Upon completion of vaccination, patients can download a digitally generated PDF certificate. Each certificate contains a unique QR code that links back to the system for verification — a critical feature for travel and institutional requirements.

B. AI Chat Support

A standout feature of the Vaccination Management System is its integrated AI chat assistant, powered by the Groq API. Available from within the patient dashboard, the chatbot can answer questions about specific vaccines (indications, contraindications, side effects), help users understand their vaccination schedule, and provide general health guidance.

This reduces the need for patients to contact clinic staff for routine queries, freeing up human resources for more complex cases.
The chatbot is designed to be informative but appropriately cautious — it does not replace medical advice but serves as an accessible first point of information. Conversations are handled in real time, and the interface is simple enough for users with limited technical literacy.

C. Administrator Features

System administrators have access to a comprehensive dashboard that provides an overview of the entire platform. They can add, edit, or deactivate vaccination centers; manage the vaccine catalogue; create and modify schedules; and monitor appointment load across locations. The admin panel also includes user management, allowing administrators to review accounts and resolve issues.
Analytics are built into the admin view, offering visibility into appointment volume, vaccine utilization rates, and center-level performance. These insights support operational planning, such as adjusting inventory levels or opening additional appointment slots during peak demand.

D. Vaccine Administrator Features

The vaccine administrator role is purpose-built for on-ground staff at vaccination centers. Rather than exposing them to the full system, they see a focused dashboard for managing their center's vaccine stock, confirming appointments, and recording completed vaccinations. This separation of concerns keeps the interface clean and reduces the risk of accidental data modification.

V. SECURITY IMPLEMENTATION

Security is treated as a first-class concern in the system. All user passwords are hashed using Werkzeug's PBKDF2-based password hashing before storage — plain-text passwords are never persisted. Session management is handled by Flask-Login, with secure cookie settings to prevent session hijacking.
Database connections are encrypted using SSL/TLS, with support for certificate-based authentication when using managed cloud database providers such as Aiven.io. SSL certificates are stored as Base64-encoded environment variables, avoiding hardcoded credentials in source code.
The application also implements CSRF protection to guard against cross-site request forgery attacks, and role checks are enforced server-side on every protected route — client-side UI restrictions alone are never relied upon. The MIT-licensed codebase is publicly available for security review and community contributions.

VI. DEPLOYMENT AND AVAILABILITY

The Vaccination Management System is deployed and publicly accessible at https://vaxincare.onrender.com/. The production deployment uses Gunicorn with four worker processes, bound to port 10000, with a 120-second request timeout — a configuration suited for moderate traffic loads typical of a community-scale vaccination program.
The application is cloud-agnostic and can be self-hosted on any Linux server with Python 3.x and MySQL. The repository includes a detailed setup guide covering local development (using the Flask development server) and production deployment (using Gunicorn). Database initialization is handled through SQL scripts provided in the repository (Create_tables.sql and Populate_all.sql), making it straightforward to spin up a fresh instance.
For teams requiring a managed MySQL instance, the documentation includes instructions for configuring SSL connections with Aiven's cloud database service. Environment variables are used throughout to keep sensitive credentials out of the codebase.

VII. RESULTS AND DISCUSSION

This Vaccination Management System demonstrates that a small development team can build a production-grade healthcare management system using widely available open-source tools. The system successfully covers the full vaccination management workflow — from patient onboarding through to certificate issuance — in a single, integrated platform.
The multi-role architecture ensures that each category of user — patient, clinic staff, and administrator — interacts only with the features relevant to their responsibilities. This not only improves usability but also reduces the surface area for accidental errors or unauthorized data access.
The AI chat integration, while not a replacement for clinical guidance, meaningfully reduces the information gap that many patients experience when navigating vaccination programs. Early feedback from the deployed system suggests that users find the chatbot particularly helpful for understanding which vaccines are due and what to expect post-vaccination.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Limitations of the current system include the absence of push notifications or email reminders for upcoming appointments, and the lack of integration with external EHR systems. These represent natural avenues for future development.

VIII. CONCLUSIONS

The Vaccination Management System addresses a genuine gap in healthcare infrastructure by providing a complete, open-source, web-based vaccination management system that is immediately deployable. By combining a well-designed relational database, secure authentication, role-based access control, digital certificate generation, and AI-powered support, the platform offers a level of functionality typically associated with enterprise healthcare software — but at zero licensing cost.

The system is built with extensibility in mind. Future work will explore the addition of appointment reminder notifications, integration with national immunization databases, multilingual support for broader accessibility, and a mobile application for patients in low-connectivity environments.

This system is a step toward a world where every person has immediate, reliable access to their vaccination history and can interact with the healthcare system on their own terms. The codebase is open for contributions at https://github.com/dattaaaaa/vaxcare.

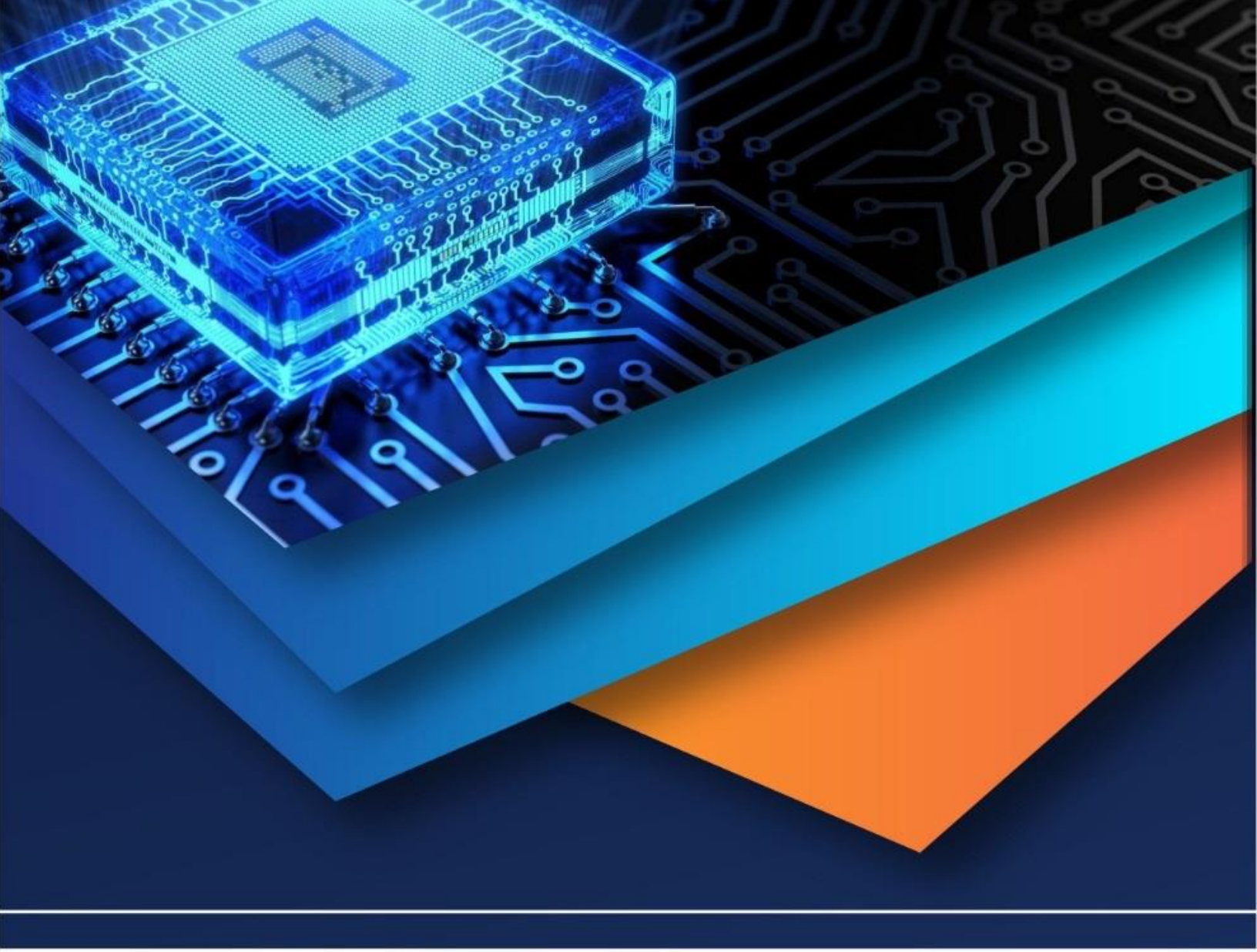
IX. ACKNOWLEDGMENT

The authors express sincere gratitude to Prof. Pratiksha Shinde, Department of Computer Science (AIML), Bharat College of Engineering, Maharashtra, for her expert guidance, continuous encouragement, and valuable feedback throughout this project. The authors also thank the open-source community behind Flask, SQLAlchemy, ReportLab, and the qrcode library for providing the building blocks that made this system possible. Gratitude is extended to Groq for their API access enabling the AI chat functionality. This project was developed as part of academic coursework at Bharat College of Engineering, Maharashtra, India.

REFERENCES

[1] World Health Organization, "Vaccination and trust: How concerns arise and the role of communication in mitigating crises," WHO Europe Policy Summary, 2017.
[2] R. Dombkowski, K. Lantz, and G. Freed, "Role for Health Information Exchange in Immunization Registries," Am. J. Manag. Care, vol. 10, pp. 67–74, 2004.
[3] A. Laranjo et al., "Conversational agents in healthcare: a systematic review," J. Am. Med. Inform. Assoc., vol. 25, no. 9, pp. 1248–1258, Sep. 2018.
[4] M. Grinberg, Flask Web Development, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.
[5] ReportLab Inc., "ReportLab PDF Library User Guide," [Online]. Available: https://www.reportlab.com/docs/reportlab-userguide.pdf
[6] Groq Inc., "Groq API Documentation," [Online]. Available: https://console.groq.com/docs
[7] Vaccination Management System GitHub Repository. [Online]. Available: https://github.com/dattaaaaa/vaxcare

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)