



AIORI 2.0 HACKATHON

Report on

Problem Statement 11

“QR-to-Database Real-Time Interaction System (QRCONNECT)”

Team Members

Atul Dattatray Dubal (BE ECS)

Yogesh Balasaheb Kalel (BE ECS)

Under the supervision of

Dr. Mansi Subhedar (HOD ECS)



Department of Electronics & Computer Science
Mahatma Education Society's

Pillai HOC College of Engineering and Technology

Pillai HOCL Educational Campus, HOCL Colony, Rasayani Tal.

Khalapur, Dist. Raigad-410207

University of Mumbai

● List of Contents

| | |
|---|----|
| 1. Problem Description | 3 |
| 2. Solution Proposed | 5 |
| 3. Optimization Proposed by the Team..... | 7 |
| 4. Project Timeline | 9 |
| 5. References..... | 10 |
| 6. Conclusion..... | 11 |

Chapter 1 Problem Definition

- **Widespread Use of QR Codes**

QR codes are commonly used for **authentication, attendance tracking, digital payments, and event check-ins**, making them a key part of modern digital workflows.

- **Fragmented and Inefficient Workflows**

Current QR-based systems often require users to go through multiple steps such as:

Scan → Open link → Input or Upload data → Submit.

This process is slow, repetitive, and reduces overall user adoption.

- **Limitations of Existing Solutions**

Present solutions like **WhatsApp Web** rely on QR scanning but are **closed, single-purpose, and not adaptable** for general use. They cannot be applied across multiple industries or customized for diverse needs.

- **Need for a Unified Framework**

There is a strong need for a **unified, secure, and real-time QR-to-database interaction system** that eliminates inefficiencies and provides a seamless user experience.

2.Key Requirements of the Proposed System

1. Seamless Two-Way Communication

Direct interaction between mobile devices and backend databases, enabling instant verification and data exchange.

2.Real-Time Dynamic Updates

Web interfaces and dashboards should update instantly using **real-time technologies** (e.g., WebSockets, Server-Sent Events) without requiring manual page refresh.

3.Cross-Industry Adaptability

A general framework that can be easily adapted to various sectors such as:

- **Education** → attendance tracking, exam authentication, access to study material.
- **Enterprises** → employee logins, meeting check-ins, secure workspace entry.
- **Healthcare** → patient record access, appointment confirmations, lab test verification.
- **Event Management** → ticket scanning, live seat allocation, entry validation.

4. Security and Privacy

Strong encryption, role-based access, and secure session handling to ensure safe data exchange.

Chapter 2 Solution Proposed

A **universal QR interaction framework** designed to be **secure, real-time, and API-first**, enabling seamless communication between mobile devices, backend databases, and web interfaces.

- **Solution Architecture And Design**

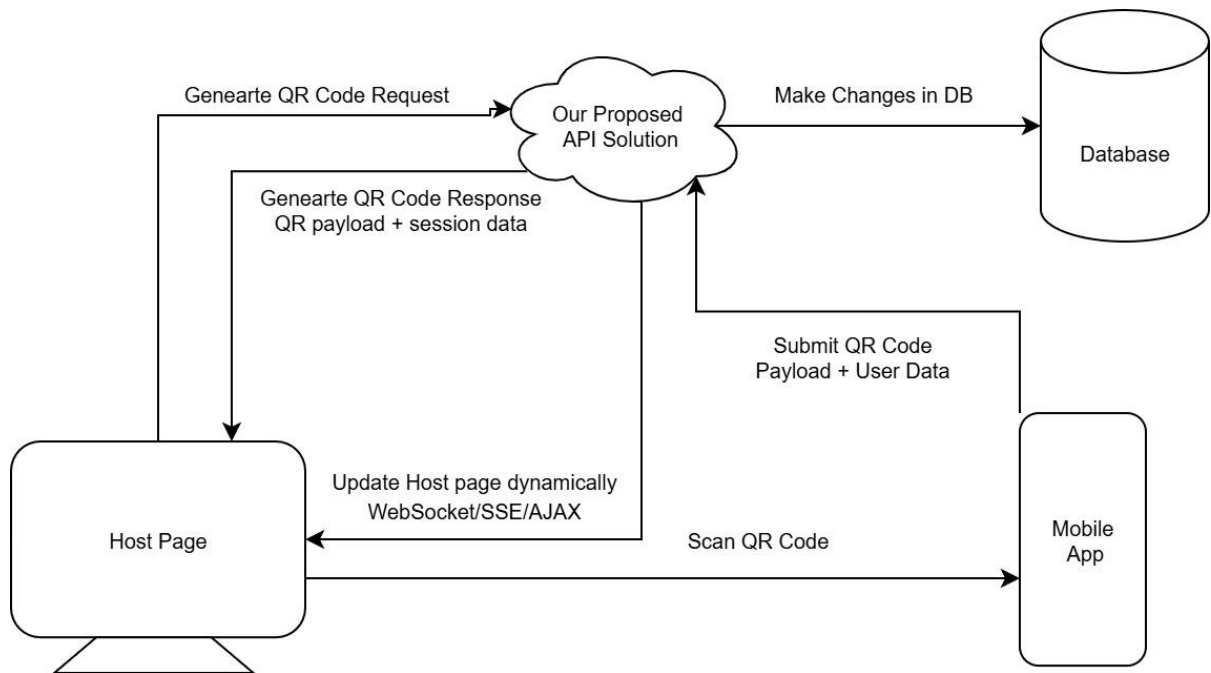


Fig-1.1 Solution Architecture And Design

- **Workflow**

1. **QR Code Generation**

The host web interface generates **unique, secure session-based QR codes** for each user or session.

2. User Scan & Auto-Submission

The user scans the QR code. The phone **automatically opens a secure link** and **submits the required payload** (e.g., user identity, attendance, ticket data) without manual input.

3. Backend Verification & Database Update

The backend verifies the submitted payload using **JWT tokens, rotating QR codes, and authenticity checks**. Upon successful verification, the data is securely **written to the database**.

4. Real-Time Dashboard Update

The host web interface updates **instantly** using **WebSocket or Server-Sent Events (SSE)**, eliminating the need for manual page refresh.

5. API-First Design

Built as an **API-first platform**, making it easy to integrate with **third-party systems** across multiple industries.

● Why This Solution? (Highlights)

1. **Zero-Touch Capture** → Scanning alone completes the process, eliminating manual input.
2. **Secure** → Strong authentication with JWT, rotating QR codes, and payload verification.
3. **Real-Time** → Web dashboards update instantly without manual refresh.
4. **Cross-Industry Applicability** → Can be adapted to Education, Enterprises, Healthcare, and Event Management.
5. **API-First Design** → Flexible and easy integration with external platforms.

Chapter 3 Optimization Proposed By The Team

1. Universal QR Interaction Framework

Instead of just login, support attendance, event check-in, payments, visitor logs — one platform, many use cases.

2. Security Upgrade

Use rotating QR (time-based, like OTP) → prevents screenshots / replays. Add device binding (WebAuthn on phone) → ensures the right person scanned. Add encryption at QR level → payload isn't plain session ID, but encrypted.

3. Real-time Features Beyond Insert DB

Host page not only shows “scanned,” but also displays user identity, role, payment status instantly. Multiple host pages subscribe to the same session (distributed live dashboards).

4. Offline / Edge Mode

If phone is offline, scanned payload is signed + stored locally → syncs later → still verifiable. Useful in rural/low-connectivity areas.

5. Open API for Developers

Instead of just a demo, provide a small SDK: POST /api/scan and GET /api/session so anyone can plug this into their app (schools, events, enterprises).

● **A Comparative Analysis of Legacy and Proposed QR Interaction Systems**

| No. | Feature | Problem Statement Requirements | Our Proposed Solution |
|-----|--------------------|--|---|
| 1. | Workflow | Multi-step: Scan → Open Link → Fill Form → Submit. | Zero-Touch: Scan and the action is complete. |
| 2. | Security | Static QR code is vulnerable to screenshot/replay attacks. | Dynamic Rotating QR Codes and JWT-signed payloads prevent unauthorized use. |
| 3. | Data Sync | The host page requires a manual refresh to see updates. | Instantaneous real-time updates via WebSockets/SSE. No refresh needed. |
| 4. | Reliability | Fails completely in offline or poor network conditions. | Offline caching ensures data is captured and synced later. |
| 5. | Intégration | Closed, single-purpose system. Difficult to integrate. | API-first design allows for simple, flexible integration with any platform. |

Chapter 4 Project Timeline

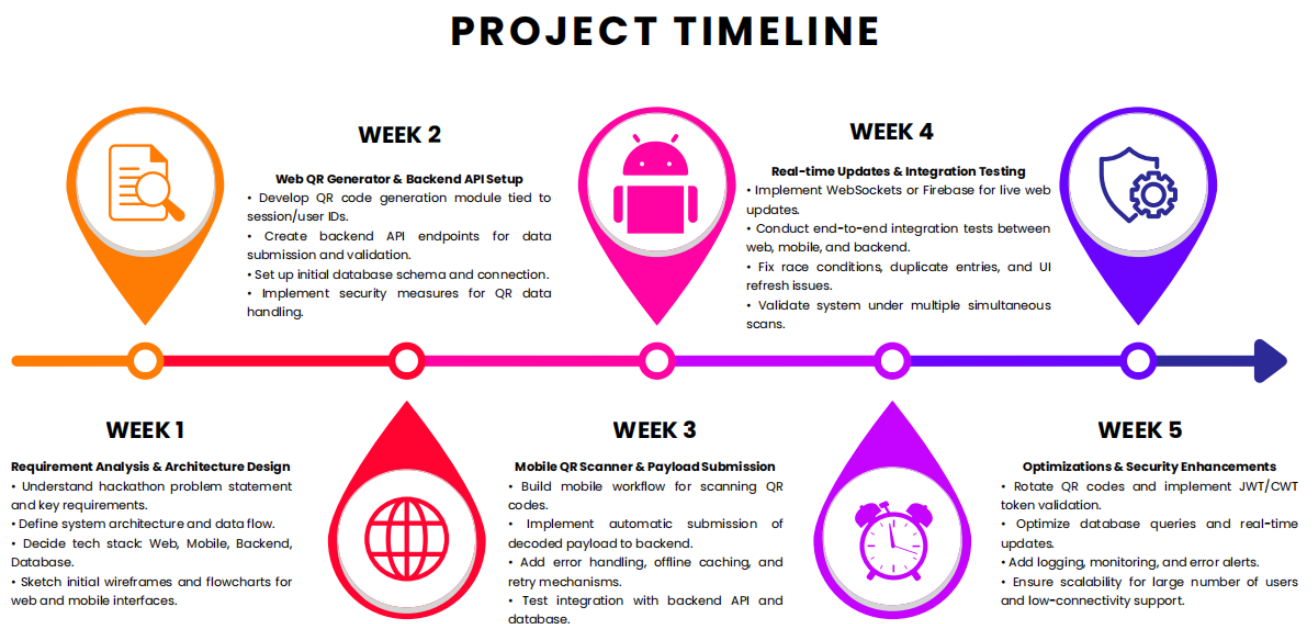


Fig-4.1 Project timeline of QRCONNCT

Chapter 5 Reference

1. Hardt, D. (2012). *The OAuth 2.0 Authorization Framework* ([RFC 6749](#)). Internet Engineering Task Force (IETF).
2. ISO/IEC. (2015). *Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification* ([ISO/IEC 18004:2015](#)).
3. Jones, M., Bradley, J., & Sakimura, N. (2015). *JSON Web Token (JWT)* ([RFC 7519](#)). Internet Engineering Task Force (IETF).
4. Jones, A. (2018). *CBOR Web Token (CWT)* ([RFC 8392](#)). Internet Engineering Task Force (IETF).

Chapter 6 Conclusion

QRCONNECT project successfully addresses the widespread inefficiencies found in conventional QR code systems. By creating a universal, secure, and real-time interaction framework .It eliminates the fragmented, multi-step workflows that hinder user adoption. The system's core innovations—such as **zero-touch data submission**, instant dashboard updates via **WebSocket/SSE**, and a robust **API-first architecture** establish a new standard for seamless digital interaction. Key security upgrades, including the use of **rotating QR codes**, **JWT validation**, and **device binding**, ensure that data exchange is both rapid and secure. QRCONNECT serves as a versatile and scalable foundation adaptable across numerous industries, from education and healthcare to enterprise management.