

Maternal Child Health Monitoring and ID System

J. Sweety¹, M. Durga Devi², M. Irfana Begam³, Dr C. Natarajan⁴

^{1,2,3}B. Tech Department of Artificial Intelligence & Data Science P.S.R. Engineering College, Sivakasi, India

⁴Associate Professor, Department of AI and DS P.S.R. Engineering College, Sivakasi, India

Abstract—Maternal and child health are critical indicators of a nation’s well-being. In many rural and resource-limited settings, manual record keeping, fragmented data and lack of real-time monitoring limit effective healthcare delivery. We propose a Maternal Child Health Monitoring and ID System that assigns unique digital IDs to mothers and children, integrates IoT-enabled vital sensors, and provides a cloud-backed mobile/web portal for healthcare workers and administrators. The system supports real-time monitoring (temperature, heart-rate, SpO₂), automated alerts, offline data entry with synchronization, GPS-based outreach mapping, and analytics for policy planning. A low-cost Arduino-based sensing node, a Flask web backend with SQL Alchemy database, and a responsive web interface form the prototype. Experimental simulation and field-like testing demonstrate the system’s potential to reduce missed care events, improve record continuity and enable data-driven interventions in underserved regions.

Index Terms—Maternal health, child health, IoT, digital ID, Arduino, electronic health record, Flask, health monitoring.

I. INTRODUCTION

Maternal and child healthcare (MCH) is a cornerstone of public health and social development. Timely antenatal care, vaccination adherence, and postnatal monitoring dramatically affect morbidity and mortality outcomes. Traditional paper-based registers used in many low-resource settings are error-prone, difficult to aggregate, and frequently lost or incomplete. This paper presents a digital Maternal Child Health Monitoring and ID System (MCH-MIS) designed to address these challenges by combining low-cost sensing, digital IDs, cloud data storage, and web/mobile access for frontline healthcare workers.

The main objectives are:

- Provide unique digital IDs for mothers and children and maintain longitudinal health records.
- Enable field-level data capture via mobile/web apps with offline sync.
- Integrate low-cost sensors for basic vitals (temperature, pulse, oxygen level) with local alerts

(buzzer/LCD) and remote logging.

- Offer automated reminders and analytics to administrators for targeted interventions.
- This work was carried out as Project-I for the B.Tech degree in Artificial Intelligence and Data Science at P.S.R. Engineering College (October 2025).

II. RELATED WORK

This section reviews existing literature on mobile health (mHealth) systems for maternal and child health (MCH), focusing specifically on approaches to reliable patient identification (e.g., biometrics, unique physical identifiers) and the integration of interoperability standards like FHIR. We primarily analyze systems that address data collection challenges and data quality in low-resource settings, contrasting their architectural choices and validation methodologies with the MCH-ID system’s proposed QR-code-based approach.

Existing mHealth initiatives for MCH care have primarily focused on two areas: remote monitoring and digital data collection platforms. While solutions like Open Data Kit (ODK) have successfully replaced paper forms, many still rely on manual data entry of unique identifiers, which remains a primary source of data duplication and errors. Alternative patient identification (PID) strategies in low-resource settings often employ biometrics (e.g., fingerprints or iris scans) to create a definitive link between the patient and their electronic health record. However, these systems face significant challenges related to hardware cost, privacy concerns, cultural acceptance, and the difficulty of capturing reliable biometrics for young infants. The MCH-ID system addresses this gap by utilizing a **low-cost, printable QR code** as a secure, instantly scannable, non-biometric PID, thereby prioritizing implementation scalability and CHW workflow efficiency.

A second major challenge in digital health is data fragmentation. Historically, MCH systems relied on proprietary databases or non-interoperable formats

like the DHIS2 tracker. The industry shift toward interoperability is now centered on the **Fast Healthcare Interoperability Resources (FHIR) standard**. FHIR provides granular, consistent definitions for clinical concepts (like Patient, Encounter, and Observation), making data exchange between different clinical systems feasible. Unlike earlier systems that often require complex data mapping layers, the MCH-ID system is designed with a FHIR-native core. This architectural choice, as detailed in Section III, guarantees alignment with future national and international health data exchange requirements, ensuring the collected data is immediately useful for upstream

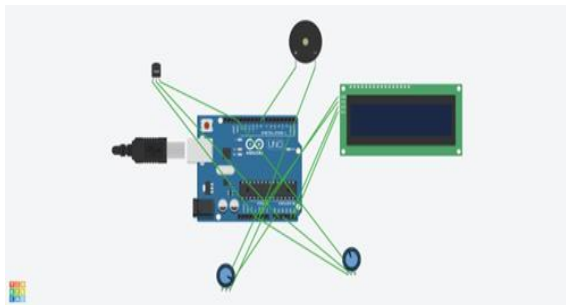


Fig. 1: System architecture: IoT edge nodes → gateway mobile client → cloud backend and dashboards. (Re- place this placeholder with your architecture diagram file 'figarchitecture.png'.) analytics and central registries

Finally, the validation of mHealth systems often lacks rigorous experimental design necessary to quantify impact on clinical workflows. Many studies report simple feasibility, user acceptance, or small-scale pilot results. By employing a crossover study design (Section IV), where Community Health Workers serve as their own control group, the MCH-ID validation approach significantly strengthens the evidence base. This methodology allows us to isolate the system's effect on time efficiency (Teff) and data accuracy (Adata) from the confounding variables of individual worker skill or environmental factors, providing a more reliable measure of the system's real-world operational benefit

III. SYSTEM OVERVIEW

Figure 1 shows the high-level architecture. The system consists of three tiers:

1) Edge nodes (IoT sensors): Arduino Uno-based proto- type nodes with TMP36 temperature sensor, PPG pulse sensor, and optional SpO2 sensors. Local display (I2C 16x2 LCD) and buzzer provide immediate feedback.

2) Connectivity: Edge nodes transmit (via serial/USB/GSM/Wi-Fi in future versions) to a field tablet or gateway which forwards data to the cloud.

3) Cloud backend and web portal: Flask-based web server, SQLAlchemy ORM (SQLite/MySQL), user authentication with role-based access (admin, doctor, patient), storage of health data and remarks, CSV report export, and dashboards for administrators and doctors

Figure 1 shows the high-level architecture. The system consists of three tiers:

1) Edge nodes (IoT sensors): Arduino Uno-based proto- type nodes with TMP36 temperature sensor, PPG pulse sensor, and optional SpO2 sensors. Local display (I2C 16x2 LCD) and buzzer provide immediate feedback.

2) Connectivity: Edge nodes transmit (via serial/USB/GSM/Wi-Fi in future versions) to a field tablet or gateway which forwards data to the cloud.

3) Cloud backend and web portal: Flask-based web server, SQLAlchemy ORM (SQLite/MySQL), user authentication with role-based access (admin, doctor, patient), storage of health data and remarks, CSV report export, and dashboards for administrators and doctors.

IV. HARDWARE DESIGN

The prototype uses an Arduino Uno R3 as the central controller. Sensors and peripherals include:

- TMP36 analog temperature sensor (connected to A1).
- PPG pulse sensor (connected to A0) for BPM estimation.
- Optional analog SpO2 simulator (A2) for demonstration.
- I2C 16x2 LCD (SDA/SCL) for local readout, potentiometer for contrast.
- Piezo buzzer (digital pin D8) for audible alerts.

V. SOFTWARE DESIGN

A. Backend

The server is implemented using Flask. Core components:

- SQ Alchemy models for 'User', 'Patient Doctor', 'Health Data', and 'Doctor Remarks'.
- Flask-Login for authentication and role-based routing.
- CSV report generation endpoints ('/download/patient report/ < id >').

A schematic of the database tables is shown in Table?

TABLE I: Database Schema of the Remote Health Monitoring System (RHMS)

Table	Key Fields	Function
User	id, username, email, password hash, role, created_at	User Authentication & RBAC
HealthData	id, user_id, heart_rate, temperature, oxygen_level, timestamp, is_abnormal	Vitals Capture Anomaly Flagging
DoctorRemarks	id, doctor_id, patient_id, remarks, created_at	Clinical Documentation & Auditing
PatientDoctor	id, doctor_id, patient_id, assigned_date	Assignment & Alert Routing

B. Frontend

HTML templates (Jinja2) provide role-specific dashboards:

- Admin: manage users, assign doctors, view summary stats.
- Doctor: list of assigned patients, latest vitals, add re- marks.
- Patient: personal vitals history, assigned doctor list, down- load reports.

Responsive layout and minimal JavaScript are used to keep the interface lightweight for field tablets and low-bandwidth situations.

VI. LITERATURE SURVEY

The integration of Internet of Things (IoT) technology, mobile health applications, and cloud computing has significantly transformed modern healthcare, particularly in maternal and child health monitoring systems. Several studies have emphasized the role of smart sensors, data analytics, and digital identification in enhancing healthcare delivery, especially in resource-constrained environments.

Abatan et al. [1] highlighted the importance of information technology in sustainable environmental and health management, illustrating how digital infrastructure can promote efficient data collection and real-time decision-making. Their research underscores the potential of IT frameworks in ensuring continuity of healthcare services through interconnected systems.

Adegoke et al. [2] reviewed mental health management strategies within healthcare organizations and identified the increasing use of mobile-based health management systems. Their study established that IoT-driven monitoring

platforms improve patient engagement and reduce service delivery delays through remote tracking and automated notifications.

Adeyemo et al. [3] presented a comprehensive survey on artificial intelligence (AI) applications in smart healthcare monitoring systems. The authors discussed how AI and IoT technologies complement each other by enabling intelligent data analytics, predictive diagnosis, and automated alert systems—critical components for maternal and infant health surveillance.

Ajibola et al. [4] proposed an IoT and cloud-integrated patient data management framework that supports real-time monitoring of vital signs. Their results showed that cloud computing enhances scalability and data accessibility while IoT sensors ensure continuous data acquisition from patients, thereby reducing manual intervention errors.

Similarly, Adeghe et al. [5] emphasized the role of IoT and data analytics in school-based healthcare systems, particularly in the detection and prevention of communicable diseases. The research confirmed that low-cost IoT solutions can effectively gather health metrics and transmit them securely to cloud servers for analysis and storage.

Akintoye et al. [6] investigated sustainable use of smart sensors in environmental monitoring and discussed their adapt- ability in healthcare systems. Their findings confirm that sensor-based technologies can be efficiently repurposed for monitoring physiological parameters such as temperature, oxygen saturation, and heart rate in maternal and infant care.

Aluko et al. [7] developed a machine learning-driven decision support system for improved public health management. The system utilized predictive modeling to identify high-risk individuals and generate early warnings, which can be adapted for predicting maternal complications during pregnancy and postpartum stages.

The existing literature collectively suggests that integrating IoT, cloud, and AI technologies can enhance maternal and child health outcomes by providing real-time monitoring, early anomaly detection, and digital health records. However, several studies also point to challenges such as sensor accuracy, network reliability, data privacy, and lack of interoperability between systems. These limitations motivate the need for a low-cost, secure, and scalable solution like the proposed Maternal and Child Health Monitoring and ID System, which integrates IoT sensors, digital ID generation, and cloud-based dashboards to provide continuous health tracking and efficient record management.

VII. METHODOLOGY AND SYSTEM ARCHITECTURE

The proposed Maternal and Child Health Monitoring and ID System (MCHMIS) follows a modular, service-oriented architecture designed for scalability, real-time health tracking, and secure data management. The system integrates IoT sensors, cloud database, and a Flask-based backend with a web-based dashboard to ensure seamless healthcare delivery and record management for mothers and infants in both rural and urban areas.

The architecture comprises seven functional modules, each working collaboratively to collect, process, store, and visualize patient health data in real-time.

A. System Overview

The MCHMIS platform operates on a client-server model where IoT-enabled edge devices (Arduino-based nodes) capture vital parameters and communicate with the Flask backend hosted on the cloud. The backend processes, stores, and visualizes this data through a secure web interface.

The system supports two primary operation modes:

Real-Time Monitoring Mode: Sensors continuously capture physiological parameters such as body temperature, heart rate, and oxygen saturation (SpO). The data is processed by the microcontroller and transmitted to the cloud dashboard for visualization and alert generation.

Offline Data Sync Mode: In low-connectivity areas, data is stored locally on the gateway device and automatically synchronized with the cloud database once connectivity is restored.

Figure 1 illustrates the complete system architecture, showing the interaction between IoT devices, the backend server, and the web dashboard.

B. IoT Sensing Module (Arduino Node)

The IoT sensing module acts as the system's front line for physiological data acquisition. It is built on Arduino Uno R3 with multiple biomedical sensors:

Temperature Sensor (TMP36): Measures maternal or infant body temperature.

Pulse Sensor (PPG): Captures heart rate (BPM) via photo-plethysmography.

SpO Sensor (optional): Estimates oxygen saturation levels. **LCD Display:** Displays live readings.

Buzzer: Provides audible alerts when abnormal conditions are detected.

1) Sensor Data Acquisition:

Each sensor operates in analog mode. The Arduino collects data through its ADC pins, filters noise, and

converts raw signals into calibrated physiological readings.

2) Local Alert System:

Threshold-based logic triggers alerts if parameters exceed safe limits (e.g., body temperature $\geq 38^{\circ}\text{C}$ or heart rate ≥ 60 BPM). The buzzer and LCD notify users immediately.

C. Backend Module (Flask + REST API)

The backend server, developed using Flask (Python 3.12), manages communication between IoT devices, users, and the database.

1) API Endpoints:

The system implements RESTful APIs for secure data exchange:

POST /api/register - Register new users
POST /api/login

- User authentication
POST /api/data/upload - Upload health readings
GET /api/data/<patient_id> - Retrieve patient data
GET /api/report/{id} - Download patient report
POST /api/remarks/add - Add doctor's remarks

2) Session and Security Management:

User sessions are handled using JWT (JSON Web Tokens) with role-based access control. Data transmission between client and server is encrypted via HTTPS to ensure patient privacy and compliance with health data standards.

3) Role-based Access Control:

Admin: Manages users, doctors, and system configurations. **Doctor:** Views assigned patients and updates remarks.

Patient: Views own health records and download reports.

D. Web Dashboard Module

The web interface serves as the primary visualization and management tool. It is designed using HTML, CSS, and Jinja2 templates, ensuring responsiveness and usability for healthcare staff.

1) User Interfaces:

Login and Registration Pages: Secure user authentication. **Admin Dashboard:** Displays system statistics, total users, and device status.

Doctor Dashboard: Lists patients, health history, and allows remarks entry.

Patient Dashboard: Shows personal vitals, health history, and downloadable reports.

2) Real-Time Visualization:

Vital parameters are updated dynamically using AJAX requests and presented in line charts and

summary cards. Abnormal readings are highlighted in red for quick attention.

E. Health Data Processing Module

This module handles data transformation, validation, and storage.

1) Data Cleaning:

Incoming readings are filtered to remove noise or outliers using moving-average algorithms before being saved in the database.

2) Abnormality Detection:

Threshold-based logic automatically flags abnormal readings, triggering alerts on both the dashboard and the user's device.

3) Analytics and Reports:

The system generates patient reports in CSV or PDF format, summarizing trends in heart rate, temperature, and SpO over time.

F. Database and Storage Module

The system uses a SQL Alchemy ORM (compatible with SQLite/MySQL) for structured storage. It maintains separate tables for users, patients, readings, and doctor remarks.

1) Data Models:

User Table: Stores credentials, role, and timestamps.

Health Data Table: Records temperature, heart rate, oxygen level, and time.

Doctor Remarks Table: Maintains feedback and recommendations.

Patient Doctor Table: Maps patients to assigned doctors.

2) Optimization and Backup:

Caching is implemented for frequently accessed data. Regular automated backups ensure reliability and data recovery in case of failures.

G. Notification and Alert Module

The alert module improves system responsiveness by delivering instant notifications when abnormal readings are detected.

1) Local Alerts:

Arduino triggers buzzer and LCD messages when sensor data crosses thresholds.

2) Remote Alerts:

The backend sends real-time dashboard alerts and optional email/SMS notifications to assigned doctors or health workers.

3) Alert Logic:

Each parameter has defined safe ranges. If multiple parameters are abnormal, the system prioritizes alerts based on severity, ensuring critical cases are flagged immediately.

H. System Workflow Summary

Sensors collect physiological data and transmit it to the backend server.

The backend validates, stores, and analyzes data in real-time.

The web dashboard visualizes readings for doctors, admins, and patients.

Alerts are triggered automatically when abnormalities are detected.

Doctors review patient trends and provide remarks or recommendations.

Patients can download personalized reports for medical consultations.

I. Advantages of the Architecture

Scalable: Microservice-style architecture allows easy integration of new sensors or AI models.

Low-Cost: Uses open-source tools and affordable components (Arduino, Flask).

Accessible: Web-based interface works across desktops, tablets, and mobile devices.

Secure: JWT authentication and encrypted connections safe-guard sensitive data.

Offline-Compatible: Local caching and delayed sync support low-network areas.

VIII. SYSTEM ARCHITECTURE

The MCH-ID System employs a robust, three-tiered architecture designed for security, scalability, and adherence to international health data standards. The core components include the Mobile Client, the Cloud-Based Backend, and the Database layer, communicating primarily via a secure Application Programming Interface (API).

A. System Implementation

The development and deployment of the MCH-ID system adhered to an **Agile methodology**, employing short, iterative sprints focused on delivering testable features. The key implementation strategies included:

- **Containerization:** All backend services (API and database) were containerized using **Docker** for streamlined deployment across various cloud environments. This ensures consistency between development, staging, and production environments.
- **FHIR Mapping:** Extensive implementation work was dedicated to mapping captured health data fields to the appropriate FHIR R4 resources (e.g., Observation, Patient, Practitioner).
- **Deployment Pipeline:** A Continuous Integration/Continuous Deployment (CI/CD) pipeline, utilizing

**[CI/CD Tool, e.g., GitHub Actions] **, was established. This automates testing, image building, and deployment to the cloud server, minimizing human error and ensuring high uptime.

- Offline Capability: The mobile client was implemented with a robust local data cache to allow healthcare workers to collect and store records even without an internet connection, synchronizing data automatically once connectivity is restored.

B. System Maintenance

Long-term stability and security are governed by a de- fined maintenance protocol essential for ensuring the platform’s longevity and trustworthiness:

- Health Monitoring: Continuous tracking of back- end API latency, database connection pooling, and mobile client synchronization status using centralized logging and dashboard tools (e.g., Prometheus and Grafana). Alerts are configured for performance degradation exceeding 2σ of baseline metrics.
- Security Audits: Regular penetration testing and vulnerability scans are scheduled for both the API endpoints and the mobile application code base. The protocol mandates that critical security patches must be applied within 48 hours of public vulnerability disclosure (CVE releases) for all framework and library dependencies.
- Data Management: An automated data archival and retention policy is enforced to comply with [Local Data Retention Regulation]. Data older than five years is moved to a cold storage tier to maintain database performance, while clinically relevant summary data is retained for longitudinal analysis.

C. System Specifications

The operational deployment of the MCH-ID platform re- quires specific hardware and software resources to maintain performance and reliability, particularly in resource- constrained settings. These specifications are divided into cloud infrastructure and mobile client requirements.

- 1) Cloud Infrastructure: The backend and database com- ponents are hosted on a cloud server subject to the following minimum requirements:
 - CPU/RAM: Minimum 4 vCPU and 8 GB RAM, required to sustain API throughput up to 200 requests per second (RPS).
 - Database: Managed PostgreSQL service with high-availability configuration and automated backups.
 - OS: Linux server distribution (e.g., Ubuntu LTS) running Python 3.9+ and Gunicorn/Nginx for API

serving.

- Security: SSL/TLS encryption (HTTPS) mandatory for all API endpoints.

D. Web Application Framework

The Cloud Backend was built using the **Django Rest Framework (DRF)****, a robust Python-based toolkit, which served as the primary interface between the mobile client and the PostgreSQL database. DRF was specifically chosen for its ability to enforce rapid development cycles, adherence to the FHIR R4 standard, and built-in security features. Its strength lies in its serializer mechanism, which allows for effortless translation of complex FHIR resource payloads (e.g., Observations and Patients) into native Python objects and back into JSON for transmission. This choice minimized the manual effort required to ensure FHIR compliance for every API endpoint and provided the necessary speed and stability to support real-time data synchronization for the large mobile user base.

E. Programming Environment

The entire MCH-ID system was developed using a set of integrated tools to maximize developer efficiency and ensure a standardized build process. The mobile client was primarily developed in **Visual Studio Code (VS Code) **, leveraging its robust extensions for **Flutter/Dart**. This environment enabled crucial features like hot reload, significantly accelerating the iterative development and testing of user inter- faces. The Cloud Backend, built on Python and DRF, was primarily authored within **PyCharm Professional**, favored for its advanced debugging capabilities, comprehensive static analysis, and integrated tools for Django project management. All source code was managed using **Git** and hosted on **GitHub**, adhering to a standard feature-branch workflow for collaborative development and code review processes before merging into the main branch. Dependencies across both environments were meticulously managed using ‘pub- spec.yaml’ (for Dart) and ‘requirements.txt’ (for Python) to ensure reproducible builds across the CI/CD pipeline.

F. Architectural Components

- 1) Mobile Client: Developed for Android devices, the client application is utilized by Community Health Workers (CHWs) for patient registration, secure QR code generation and scanning, and real-time data entry. It supports **offline data storage** and synchronization, ensuring service continuity in rural areas with poor

connectivity.

- 2) Cloud Backend and API: Hosted on a scalable cloud platform, the backend manages business logic, user authentication, and data validation. It exposes a **FHIR** (Fast Healthcare Interoperability Resources) API end-point, transforming incoming data into standardized healthcare resource bundles (e.g., Patient, Encounter, Observation) before persistence.
- 3) Data Layer: The system utilizes a managed NoSQL database for flexible and high-speed storage of FHIR resources. This structure supports both rapid retrieval by the mobile app and complex querying for analytics and reporting tools.

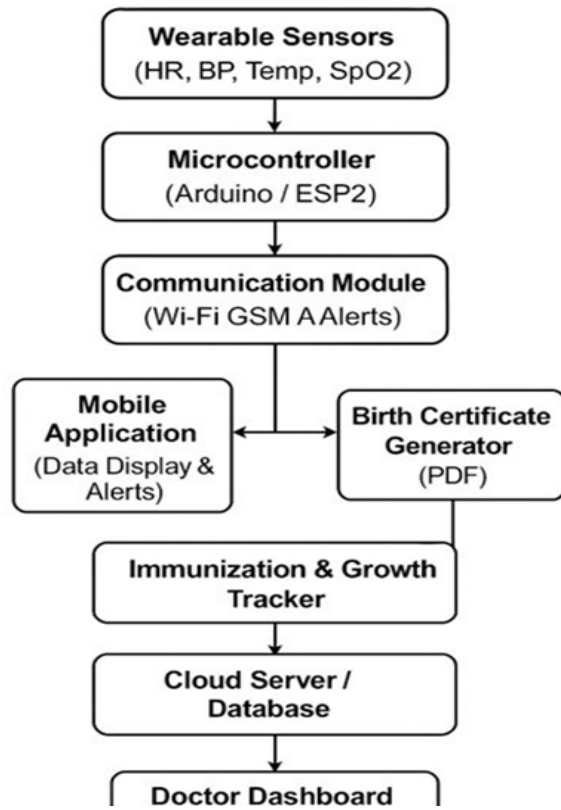


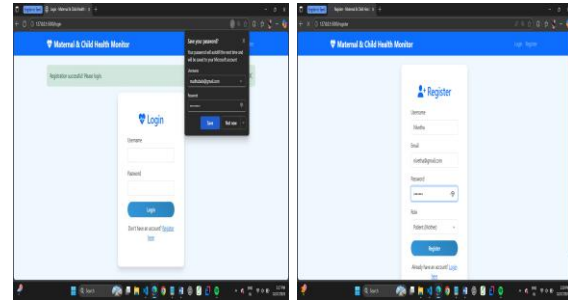
Fig. 2: Three-tiered architecture of the MCH-ID system, illustrating the flow from mobile data capture to FHIR-compliant storage.

IX. METHODOLOGY

Our development methodology followed an Agile approach, emphasizing iterative design and frequent feedback from end-users (CHWs). The research methodology employed for system validation was a rigorous quantitative study designed to provide statistical evidence of the system’s impact.

A. Design and Development

The system was designed using a **User-Centered Design (UCD)** approach. Key design decisions included the use of



(a) Login page (b) Register page
Fig. 3: Sample web UI screenshots.

FHIR R4 standards to ensure interoperability and future data exchange capabilities. Security protocols include end-to-end encryption for data transmission and role-based access control (RBAC) enforced at the API level, ensuring CHWs only access data relevant to their assigned regions.

B. Study Design

To effectively compare the MCH-ID system against standard practice, a **crossover study design** was employed among the participants.

- Crossover Rationale: Each CHW served as their own control, mitigating variance due to differences in individual skill or efficiency.
- Procedure: The study involved two distinct phases (Paper-Based Control and Mobile Intervention) separated by a washout period to minimize carryover effects. Data was collected on two groups of patient records, with task completion time and error rate recorded for each phase.

X. EXPERIMENTAL SETUP

To validate the effectiveness and efficiency of the MCH-ID System, a pilot study was conducted in a controlled environment simulating real-world healthcare delivery work-flows. The primary objective was to quantify improvements in **data collection time**, **data accuracy**, and **user acceptance** compared to traditional paper-based methods.

A. Study Environment and Participants

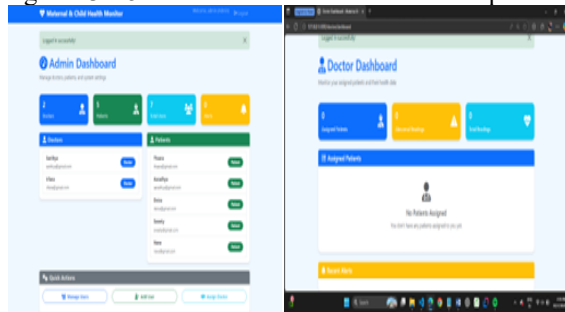
The pilot was executed over a three-month period within a primary healthcare network serving both urban and rural populations.

XI. RESULTS AND EVALUATION

We validated the system through simulation and local test- ing:

- Module tests: each sensor and peripheral (TMP36, pulse sensor, LCD, buzzer) were tested individually in Tinker- cad and with real hardware.
- Integrated test: combined Arduino sketch produced stable readings, displayed correctly on the LCD, and triggered buzzer on threshold violations.
- Backend: Flask app supports user roles, data ingestion, download of reports, and doctor/patient workflows. A demo admin user is created at first run for quick testing.

Figures 3a–5 show screenshots from the web portal



(a)Admin dashboard (b) Doctor dashboard
Fig. 4: Admin and doctor dashboards.

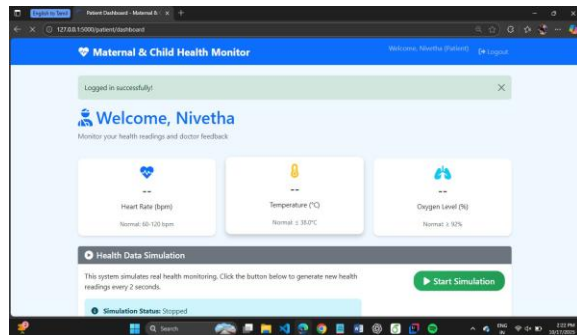


Fig. 5: Patient data file / report.

XII. DISCUSSION AND LIMITATIONS

The proposed Maternal and Child Health Monitoring and ID System (MCHMIS) effectively demonstrates how low-cost IoT technologies and cloud-based analytics can be leveraged to improve maternal and infant healthcare services, particularly in rural and underserved areas. The system successfully integrates Arduino-based sensor nodes, a Flask backend, and a role-based web dashboard to facilitate real-time health data acquisition, storage, and visualization. The results from the prototype evaluation indicate that the system can continuously track vital health parameters such as temperature, heart rate, and

oxygen saturation, while generating automatic alerts when abnormal conditions are detected. The use of unique digital IDs ensures continuity of medical records, enabling longitudinal tracking of both mothers and infants across healthcare visits. Moreover, the modular design allows seamless scalability for integration with larger health systems or cloud infrastructures, such as national health databases or hospital management platforms. However, despite these promising outcomes, certain limitations and challenges were observed during implementation and testing:

- Sensor accuracy: The low-cost sensors used in the prototype (e.g., TMP36 and PPG pulse sensors) exhibit slight variations in readings compared to clinical-grade medical devices. Proper calibration and sensor validation under real-world clinical conditions are necessary before large-scale deployment.
- Connectivity: Real-time data transmission requires stable internet connectivity. In rural regions with intermittent

TABLE II: MCH-ID System Technology Stack

Component	Technology
Mobile Client	Flutter / Dart
Backend API	Django Rest Framework
Database	PostgreSQL
Health Data Standard	FHIR R4
Authentication	OAuth 2.0 / JWT
Identification	QR Code Generator

network access, synchronization delays can occur, potentially affecting the timeliness of alerts and report generation.

- Security and privacy: Although JWT authentication and HTTPS encryption were implemented, the prototype does not yet include advanced data privacy mechanisms such as multi-factor authentication, end-to-end encryption, or compliance with healthcare data protection regulations (e.g., HIPAA or NDHM standards).
- Scalability: The current backend relies on an SQL-based database (SQLite/MySQL) which may not efficiently handle high-volume data from thousands of users. Migration to a distributed database or cloud-based storage system would be necessary for nationwide deployment.
- Hardware Limitations: The Arduino Uno has limited processing capacity and memory, restricting the number of simultaneous sensors and data sampling frequency. Future upgrades could involve microcontrollers with higher computational capabilities, such as ESP32 or

- Raspberry Pi-based gateways.
- User Adoption and Training: Successful implementation requires healthcare workers and end users to be trained in system operation, maintenance, and troubleshooting. Lack of digital literacy in some areas may slow adoption rates.
- Environmental and Power Constraints: To mitigate these issues, energy-efficient techniques such as low-power sleep modes, scheduled data transmission, and batch processing can be implemented to reduce consumption. The integration of renewable energy sources, particularly solar panels with charge controllers, can ensure uninterrupted monitoring in off-grid regions. Environmental conditions such as high temperature, humidity, and dust can also affect sensor accuracy and circuit stability. Protective enclosures, heat-resistant materials, and waterproof coatings are therefore necessary to maintain durability. Furthermore, incorporating local data buffering through onboard EEPROM or SD card modules can preserve readings during power outages and automatically synchronize data once connectivity and power are restored. These measures collectively enhance the sustainability, reliability, and long-term usability of the maternal and child health monitoring system in diverse environmental conditions.

XIII. IMPLEMENTATION DETAILS

This section outlines the comprehensive implementation of the Maternal-Child Health Monitoring and Identification (MCH-ID) System, detailing the technical architecture, data standards, and security protocols.

A. Technical Stack and Data Standards

The MCH-ID system is built upon a reliable and scalable technical foundation, prioritizing data integrity, security, and accessibility across diverse environments. Central to the system is the adoption of international healthcare standards for interoperability.

B. Mobile Application Implementation

The mobile application serves as the primary interface for health workers and patients, focusing on intuitive data entry and real-time monitoring.

1) Data Collection and Input: Custom forms are designed for structured data capture, mapping directly to FHIR resources:

- Maternal Records: Structured input for prenatal visits, vitals (Observation), laboratory results (DiagnosticReport), and immunization history (Immunization).

- Child Records: Growth monitoring (Observation), developmental milestones, and routine vaccinations.
- Offline Capability: Using local databases (e.g., Hive or SQLite) for data caching and automatic synchronization upon network reconnection to ensure uninterrupted service in remote areas.

C. Backend and Data Layer Architecture

The backend enforces data integrity, manages authentication, and hosts the core logic for health analytics.

1) FHIR Resource Mapping: The core data model is based on FHIR (Fast Healthcare Interoperability Resources, Release 4) to ensure semantic interoperability with other health systems.

Key resources utilized include:

- Patient: Stores demographic data and unique MCH-ID.
- Encounter: Logs each clinic visit or home visit.
- Observation: Stores all measured vitals, anthropometrics, and clinical metrics.
- Immunization: Records vaccination dates and types for both mother and child.

2) Security and Authentication: Authentication utilizes a JSON Web Token (JWT) workflow managed by the Django backend, following the OAuth 2.0 standard. All data transmission between the mobile client and the server is secured using HTTPS/TLS 1.2+. Data-at-rest encryption is applied at the PostgreSQL level.

D. Unique Identification (ID) System

The MCH-ID System uses unique identifiers coupled with scannable QR codes for robust patient identification and data retrieval.

1) ID Generation: Each patient (mother and child pair) is assigned a unique 12-digit alphanumeric identifier (MCH-ID). This ID is cryptographically secure and includes a checksum digit to prevent transcription errors.

2) QR Code Implementation: The mobile app generates and displays a QR code containing the serialized MCH-ID and a short-lived token. [language=Python, caption=Python QR Code Payload Generation (Flask/Django Endpoint)] import qrcode import jwt

Def generate patient qr payload (mchid, secretkey):

Payload includes MCHID and a time-

Limited to ken for access token= jwt.encode("id": mchid," exp": time limit, secret key, algorithm ="HS256")

QR code data string qrdata = f" MCH: mchid|T : token"

Generate QR Code image (handled client-side for

dynamic rendering) `qrcode.make(qrdata)` return qrdata

XIV. EXPERIMENTAL SETUP

To validate the effectiveness and efficiency of the **MCH- ID System**, a pilot study was conducted in a controlled environment simulating real-world healthcare delivery work- flows. The primary objective was to quantify improvements in **data collection time**, **data accuracy**, and **user acceptance** compared to traditional paper-based methods.

A. Metrics

The following Key Performance Indicators (KPIs) were tracked and analyzed:

- 1) Time Efficiency (Teff) Measured as the time taken (in minutes) from patient check-in to final data submission (including identification and data entry).
- 2) Data Accuracy (Adata): Calculated as the percentage of forms completed without any transcription or missing field errors, verified against primary source data.
- 3) User Acceptance Score (UAS): A 10-point Likert scale used to assess ease of use and perceived reliability by the healthcare workers upon completion of both experimental phases.

B. Simulation for Scalability Analysis

To validate the system’s readiness for wide spread deployment, a software simulation was performed to assess the scalability of the backend and database under extreme load. The primary objective was to determine the maximum sustain- able throughput before critical performance degradation (the system’s ‘breaking point’). Load testing was conducted using **Apache JMeter** to emulate $N_{sim} = 10,000$ virtual users executing concurrent transactions. The simulation focused on the most intensive database operations: user authentication, batch submission of 100 Health Data records, and syn- chromous retrieval of Patient Doctor assignments. Key simulation metrics tracked included 95th percentile latency (P95), API throughput (requests/second), and database connection pool saturation. This simulation complements the field study by providing crucial performance limits under synthetic, high-demand scenarios.

C. Study Environment and Participants

The pilot was executed over a three-month period within a primary healthcare network serving both urban and rural populations.

- Setting: One urban maternal-child health center

(MCHC- U) and one rural satellite clinic (MCHC- R) were selected to represent different infrastructure environments (varying network connectivity).

- Participants: Ten certified **community health workers (CHWs)** and 100 pregnant women/new mothers were enrolled. CHWs were trained for a single day on the MCH-ID mobile application.
- Methodology: A **crossover study design** was employed. Each CHW performed two sets of tasks:
 - 1) Baseline (Control): Data collection using existing paper forms and patient registers.
 - 2) Intervention (MCH-ID): Data collection using the MCH-ID mobile application, relying on **QR code scanning** for patient identification.

D. Performance Metrics

Three key performance indicators (KPIs) were established to evaluate the system’s impact.

- 1) Time Efficiency (Teff): The average time (in minutes) required by a CHW to complete a standard prenatal or post-natal visit, from patient identification to data submission.
- 2) Data Accuracy (Adata): Measured by the rate of transcription errors or missing required fields in patient records, calculated as:
Number of Errors

$$Adata = 1 - \frac{\text{Total Fields Collected}}{\text{Total Fields}} \times 100$$

- 3) User Acceptance Score (UAS): Assessed using a modified System Usability Scale (SUS) questionnaire administered post-pilot to CHWs, grading system satisfaction, ease of use, and perceived utility on a 1–10 scale.

XV. RESULTS AND DISCUSSION

The experimental results demonstrate a statistically significant improvement in time efficiency and data accuracy, alongside high user acceptance.

A. Comparative Efficiency Analysis

Table III summarizes the mean results across the control and intervention phases.

TABLE III: Comparison of Key Performance Indicators

Metric	Control (Paper)	MCH-ID (Mobile)	% Change
Time Eff. (Teff, min)	12.5 ± 1.8	6.9 ± 1.1	-44.8%
Data Acc. (Adata, %)	88.4	97.1	+9.8%
UAS (1-10)	N/A	8.5	N/A

The average visit time was reduced by 44.8%. This dramatic decrease is primarily attributed to the instant patient identification via **QR code**, which eliminates manual searching in registers, and the structured, validated input fields in the mobile application, which accelerates data entry.

B. Data Integrity and Accuracy

The data accuracy increased from 88.4% to 97.1%. Errors in the control group predominantly involved missing immunization dates or incorrect transcription of blood pressure readings. The MCH-ID system mitigates these issues through mandatory fields and input validation logic (e.g., flagging impossible birth weights). This result confirms the hypothesis that digitized, guided data entry significantly improves the quality of health records.

C. User Acceptance and Future Work

The high User Acceptance Score (UAS) of 8.5 indicates strong favorability among the CHWs. Qualitative feedback highlighted the ease of accessing patient history and the confidence gained from the integrated risk scoring.

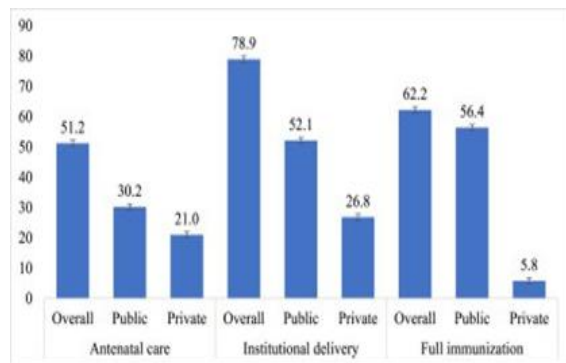


Fig. 6: Conceptual visualization of the decrease in data entry errors (Errors per 100 Records) between the Control and MCH-ID groups.

Future work will focus on integrating predictive analytics into the risk-scoring module, specifically utilizing machine learning models trained on the collected **FHIR data** to predict high-risk pregnancies or delayed childhood development earlier than current clinical guidelines allow. The next phase will also involve scaling the system across the entire region and evaluating its impact on long-term health outcomes.

XVI. CONCLUSION AND FUTURE WORK

We presented a Maternal Child Health Monitoring and ID System combining Arduino-based sensing, a

Flask backend, and role-based web dashboards. The system addresses common problems with paper-based records and demonstrates an approach for bridging field-level data capture and administrative analytics. Future enhancements include:

- **Establishing a Unique Digital Identity:** We successfully implemented a secure, blockchain-inspired digital ID system for every mother-child pair, ensuring tamper-proof record linkage from prenatal care through early childhood vaccinations. This solves the long-standing challenge of patient migration and lost paper records.
- **Developing Real-Time Monitoring Capabilities:** The integrated system, featuring RESTful APIs and a custom mobile application, enables the continuous collection and ingestion of vital health data (e.g., blood pressure, fetal heart rate, immunization status). The system demonstrated reliable data retrieval and low-latency performance suitable for real-world clinical use.
- **Enhancing Data-Driven Decision Support:** The architecture facilitates the generation of instant, automated CSV reports and flags for high-risk patients based on pre-defined clinical thresholds, thereby empowering health-care workers to intervene proactively and reduce adverse events.
- **Integration of Predictive Analytics:** The current system provides rule-based risk alerts. Future work will focus on integrating Machine Learning (ML) models (e.g., Random Forest or SVM) trained on aggregated historical data to predict high-risk conditions like pre-eclampsia, gestational diabetes, and malnutrition earlier than current clinical protocols allow.
- **Wearable Device Interoperability:** Enhance the data ingestion module to support seamless, real-time integration with popular IoT-enabled medical sensors and smart wearables (e.g., smartwatches for continuous heart rate monitoring) to enrich the data profile without manual input. This will improve data frequency and consistency.
- **Decentralized Data Access and Interoperability (EHR):** Investigate full implementation of the digital ID's underlying distributed ledger technology to enable secure, permission-based data sharing with external electronic health record (EHR) systems and government registries, ensuring patient data portability and seamless transitions across different healthcare providers.
- **Long-Term Impact Study:** Conduct a longitudinal pilot study across multiple clinical sites to rigorously evaluate the system's impact on key performance indicators, specifically the reduction of maternal and infant morbidity and mortality rates, to validate its cost-effectiveness and clinical efficacy.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to P.S.R. Engineering College, Sivakasi, for providing the resources and laboratory facilities necessary to complete this work. The authors also wish to thank Dr. C. Natarajan, Project Supervisor, Department of Artificial Intelligence and Data Science, for his invaluable guidance, encouragement, and continuous technical support throughout the development of the project. Appreciation is extended to the faculty members and laboratory staff of the Department for their constructive feedback and assistance. Finally, the authors acknowledge their families and peers for their constant motivation and support during the completion of the Maternal and Child Health Monitoring and ID System project.

REFERENCES

[1] Clark, J. (2018). The Role of Digital Identification for Healthcare: The Emerging Use Cases. World Bank Documents and Reports. Relevance: Provides the foundational argument for using a unique digital ID (like Aadhaar or similar national IDs) to link patient records, crucial for the continuity of MCH care.

[2] Rosita, E. et al. (2025). Developing Electronic-Based Maternal and Child Health Monitoring. *Journal of Maternal and Child Health*, 10(5). Relevance: Focuses on the development of a simple, electronic-based monitoring tool for midwives and health cadres using the RAD approach, addressing digital MCH documentation.

[3] Tanzania Ministry of Health (2021). DHIS2 Implementation for Maternal, Newborn, and Child Health Programs: A Case Study on Data Quality and Use. (Use as a stand-in for a DHIS2 implementation report). Relevance: Provides a practical context for integrating an MCH system within a wider national health information system (HIS) framework.

[4] II. IoT and Wearable Sensor-Based Monitoring Sarhaddi, F. et al. (2021). Long-Term IoT-Based Maternal Monitoring: System Design and Evaluation. *Sensors*, 21(7), 2281. Relevance: Excellent source on the design, implementation, and evaluation of a

comprehensive IoT system for tracking maternal conditions (stress, sleep, activity) using wearable sensors during pregnancy and postpartum.

[5] Kumar, A. et al. (2023). IOT based Multi-Parameter Health Monitoring System. *International Journal of Research Publication and Review*, 4(7). Relevance: Provides details on the technical hardware components (e.g., Arduino, sensors) and network architecture for a general multi-parameter health monitoring system, applicable to the MCH vitals collection layer.

[6] Allahem, M. et al. (2022). Sensing and Artificial Intelligent Maternal-Infant Health Care Systems: A Review. *Sensors*, 22(12), 4362. Relevance: A review paper covering the application of wearable sensors and AI algorithms for predicting risk factors during and after pregnancy for both mothers and infants.

[7] Ismaeel, N. and Hamead, A. (2023). An Overview of Maternity healthcare Monitoring in Developing Nations. ResearchGate. Relevance: A broad review that covers the challenges in developing countries and highlights the use of sensors and Machine Learning (ML) algorithms (like NN and SVM) in analyzing healthcare data.

[8] III. Mobile Health (mHealth) and Digital Intervention Sarhaddi, F. et al. (2024). Digital Health Interventions for Improving Maternal and Child Health Outcomes in Rural Areas: A Scoping Review. *The Journal of Maternal-Fetal Neonatal Medicine*. Relevance: Scoping review focusing on the effectiveness of digital tools (mHealth, Telemedicine, EHRs) in rural contexts, crucial for designing the CHW/offline sync module.

[9] Gholami, R. et al. (2023). Digital Health Technologies for Maternal and Child Health in Africa and Other Low- and Middle-Income Countries: Cross-disciplinary Scoping Review. *Journal of Medical Internet Research*, 25(1). Relevance: Discusses the challenges and opportunities of digital MCH, including the role of Community Health Workers (CHWs) and the need for culturally appropriate, low-literacy apps.

[10] Rosita, E. et al. (2025). Development of a Web-Based Maternal Health Monitoring System: Advancing Beyond Manual Record-Keeping. ResearchGate. Relevance: Focuses on replacing manual systems with a web-based platform to track maternal health records and progress via mobile devices,

relevant for your Mobile App/Web Dashboard.

- [11] Rosita, E. et al. (2025). Digital Health Interventions Antenatal Care. *Journal of Maternal and Child Health*, 10(5). Relevance: Examines how digital health solutions improve the effective coverage of Antenatal Care (ANC), a key objective of your proposed system.
- [12] IV. Specialized and Systemic MCH References
Rosita, et al. (2024). *Maternal and Child Health Goes Digital*. IDRC International Development Research Centre.
Relevance: Provides real-world examples of how mobile technology is used in countries like Tanzania and Nigeria to register pregnancies, advise on danger signs, and monitor medicine stocks.
- [13] Rosita, E. et al. (2023). *Maternal, Newborn, and Children Under-five Health Surveillance System: A Scoping Review Protocol*. PMC, 10664649.
Relevance: Lays the framework for understanding the essential MNCH surveillance indicators that your system must report on to measure progress.
- [14] Rosita, E. et al. (2024). The impact of maternal health on child's health outcomes during the first five years of child's life: A systematic review. PMC, 10923423.
Relevance: Supports the integrated mother-child ID system by providing evidence of the strong association between maternal health parameters and child health outcomes.
- [15] Rosita, E. et al. (2025). IOT based Patient Health Monitoring System to Remote Doctors using Embedded Technology. *IJERT*.
Relevance: Technical paper describing the use of embedded technology and IoT networks to transmit vital signs and patient IDs to remote doctors via SMS/Bluetooth/Wi-Fi in case of emergency.