WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

MEDICARE DIGITAL PRESCRIPTION SYSTEM

Ms. Gayathri V[#1], Lakshnas Sree M[#2], Lakshitha S[#3], Krithin T[#4], Hema S[#5]

[#1] Assistant Professor, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.
[#2] Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.
[#3] Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.
[#4] Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.
[#5] Student, Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology, India.

ABSTRACT

Healthcare facilities may manage patient registration, consultation records, prescriptions, medicine availability, and laboratory services through paper-based or disconnected workflows. Such arrangements can make record retrieval and coordination between departments more difficult. This paper presents the design and implementation approach of the Medicare Digital Prescription System, a web application developed as a student software project to organize selected hospital workflows in one platform. The application uses a React-based user interface, a Node.js backend, and a Supabase/PostgreSQL data layer. The core workflow covers patient registration with a unique Patient ID, doctor access, patient search, and digital prescription handling. The expanded administration workflow includes pharmacy information such as medicine name, availability, quantity, dosage, company, and brand, as well as laboratory information such as test name, availability, cost, and procedure details. The paper describes the problem context, related work, system architecture, modules, implementation approach, and a test plan. The design emphasizes role-oriented access, structured records, data minimization, and the need to protect sensitive health information. No clinical outcome, user-study result, or quantitative performance improvement is claimed because verified measurements were not available for this draft. The system should be treated as a project prototype until security, usability, reliability, and institutional compliance have been evaluated.

KEYWORDS—digital prescription, healthcare information system, patient registration, pharmacy management, laboratory services, web application, PostgreSQL.

I. INTRODUCTION

Digital health applications are useful only when the information they manage is accurate, retrievable, and available to appropriate users. Patient registration, consultation, prescription preparation, medicine dispensing, and laboratory testing are related activities, yet they may be recorded by different departments or through separate tools. Staff may need to repeat data entry or consult more than one record source to complete a routine task. The World Health Organization's global digital health strategy emphasizes sustainable, coordinated digital health systems that support health services and responsible use of health information [1].

The Medicare Digital Prescription System is a student-developed web application intended to bring selected administrative and prescription-related workflows into one interface. Its scope includes patient registration, unique patient identification, doctor login and patient search, digital prescription handling, and hospital-side management of pharmacy and laboratory service information. It is not intended to replace clinical judgement or to claim that digitization alone improves patient outcomes.

Problem Statement

When patient details, prescriptions, pharmacy information, and laboratory catalogues are stored separately, authorized staff may experience delays locating information, inconsistent updates, and limited visibility across departments. The project addresses this coordination problem by providing a shared application structure with role-oriented screens and a central data layer. The extent to which these benefits are achieved must be established through documented testing rather than assumed from the presence of software features.

Objectives and Contributions

The objectives are to:
(i) support digital patient registration and unique patient identification;
(ii) provide doctor-facing access for patient search and prescription operations;
(iii) organize pharmacy records and medicine metadata;

(iv) maintain laboratory service information, including availability, cost, and procedure details; and
(v) define security and testing considerations for a healthcare-oriented web application. The project contribution is an integrated workflow design that connects these functions in a modular application.

II. LITERATURE REVIEW

Digital health systems support the electronic management and exchange of healthcare information. The World Health Organization's *Global Strategy on Digital Health 2020–2025* emphasizes coordinated digital health development, governance, interoperability, and alignment with healthcare requirements [1]. These principles are relevant to systems that organize patient records, prescriptions, pharmacy information, and laboratory services.

Electronic prescribing systems help manage prescription information digitally. Their effectiveness depends on accurate patient identification, reliable prescription records, appropriate access controls, and integration with existing clinical workflows. Vejdani et al. reviewed requirements associated with electronic prescription systems, highlighting the importance of functional and technical considerations in system development [2].

Healthcare interoperability enables different applications to exchange information using consistent structures. HL7 Fast Healthcare Interoperability Resources (FHIR) provides standardized resources and exchange methods for healthcare data [3]. Although standards-based exchange is a potential enhancement for the Medicare project, it should not be described as an implemented feature unless it is supported by the actual application.

Application programming interface (API) security is another important consideration. The OWASP API Security Top 10 identifies risks such as broken object-level authorization, authentication weaknesses, and excessive data exposure [4]. Accordingly, healthcare applications should validate requests, enforce permissions on the server, and restrict access to sensitive records.

Research Gap and Motivation: The sources above address digital health strategy, electronic prescribing, interoperability, and API security. The Medicare project brings patient registration, digital prescriptions, pharmacy information, and laboratory service information together within a student-developed web application. Its contribution is the integration of these modules at prototype level. A comparative evaluation is still needed to establish whether the application improves task completion, record accuracy, or user experience.

ASPECT	SEPARATE OR MANUAL WORKFLOWS	MEDICARE PROJECT SCOPE
PATIENT IDENTIFICATION	RECORDS MAY BE STORED SEPARATELY	REGISTRATION AND UNIQUE PATIENT ID
PRESCRIPTION HANDLING	PAPER-BASED OR SEPARATE RECORDS	DIGITAL PRESCRIPTION WORKFLOW
PHARMACY INFORMATION	SEPARATE MEDICINE RECORDS	MEDICINE DETAILS AND AVAILABILITY
LABORATORY INFORMATION	SEPARATE TEST CATALOGUES	LABORATORY SERVICE DETAILS
ACCESS CONTROL	DEPENDS ON THE EXISTING PROCESS	BACKEND AUTHORIZATION SHOULD BE VERIFIED

TABLE I. CONCEPTUAL COMPARISON OF DISCONNECTED WORKFLOWS AND THE PROPOSED SYSTEM.

This table describes the project's intended scope; it is not the result of a measured experiment.

III. EXISTING SYSTEM

In conventional healthcare workflows, patient details, prescriptions, medicine information, and laboratory services may be maintained using paper records or separate digital applications. This can require healthcare staff to switch between different records to retrieve information. Updating information in separate systems may also create difficulties in maintaining consistency and accessing the latest details.

Manual prescription handling can make prescription retrieval and record management more time-consuming. Similarly, pharmacy information, including medicine availability and dosage details, and laboratory information, such as test availability and cost, may be maintained separately.

The limitations of these workflows motivate the development of an integrated digital platform that organizes related healthcare information in a structured manner. However, the actual impact of an integrated system must be established through testing rather than assumed.

IV. PROPOSED SYSTEM

The Medicare Digital Prescription System is a web-based application designed to organize patient registration, digital prescription management, pharmacy information, and laboratory services through a common platform. The system uses a React-based frontend, a Node.js backend, and Supabase with PostgreSQL for authentication and database operations, subject to the final deployment configuration.

The main modules are:

1. Patient Registration: Registers patient details and associates a unique Patient ID with the patient record.
2. Doctor Login and Patient Search: Provides authenticated doctor access and supports patient lookup.
3. Digital Prescription Management: Organizes prescription records associated with patients and authorized doctors.
4. Pharmacy Management: Supports medicine information such as name, brand, company, dosage, quantity, and availability, where implemented.
5. Laboratory Management: Organizes laboratory information such as test name, availability, cost, and procedure details, where implemented.

The backend should validate incoming requests and enforce access permissions before accessing sensitive information. The proposed system aims to centralize related information and provide a structured workflow for authorized users. Functional and security testing is necessary to verify its reliability and determine its practical benefits.

V. SYSTEM IMPLEMENTATION

Development Approach and Technology Stack

The project follows a modular web-application approach. React is used for the client-side interface, Node.js provides the server-side API layer, and Supabase with PostgreSQL is used as the intended authentication and relational data platform. The user interface communicates with backend endpoints, while the data layer stores structured records. The project source and deployment workflow use GitHub for version control, Vercel for frontend hosting, and Render for backend hosting. These platform names describe the project setup; the final author team should verify the current deployment and each feature's operational status before publication.

Technologies Used

- **Frontend:** React.js, Vite, HTML, CSS, JavaScript
- **Backend:** Node.js, Express.js
- **Database:** Supabase PostgreSQL
- **Authentication:** Doctor login and user authentication
- **Deployment:** Vercel (frontend), Render (backend)
- **Version Control:** Git and GitHub
- **API Communication:** REST APIs

FIG. 1. WORKFLOW OF THE MEDICARE PRESCRIPTION SYSTEM.

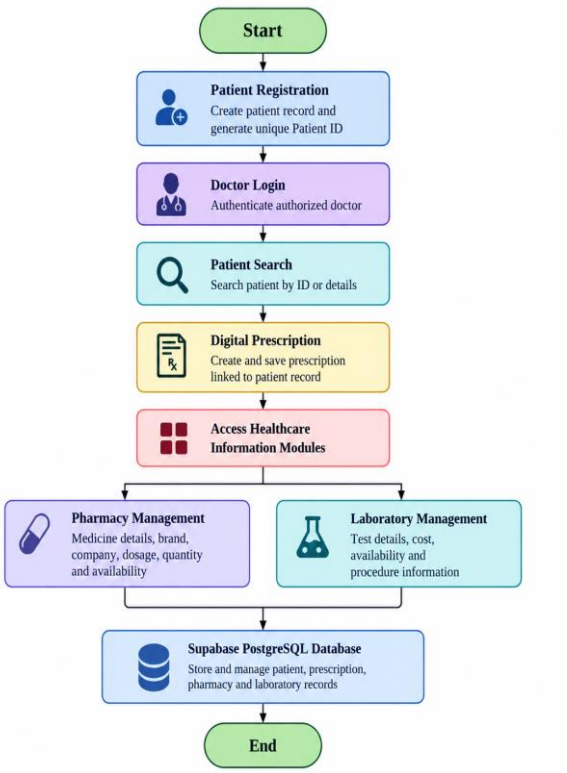


Fig. 1. Workflow of the Medicare Prescription System

Main Modules

A. Patient Registration

The patient registration module provides a structured workflow for collecting and maintaining patient details. During registration, the system captures the required information and assigns or associates a unique Patient ID to support patient identification and record retrieval. Input validation helps prevent missing mandatory values and reduces errors caused by incomplete submissions. The registered information is stored in the connected database and can be retrieved through the application's supported functions. The unique identifier also helps doctors locate the correct patient record during prescription creation. Appropriate access restrictions are important to protect patient information. For demonstrations and testing, synthetic patient data should be used instead of real personal or medical information.

B. Doctor Access and Patient Search

The doctor access module provides an authentication interface through which authorized doctors can access the functions available to them. After successful sign-in, doctors can search for registered patients using supported identifiers, such as the Patient ID. The system retrieves matching information through the backend and presents the relevant patient record for further action. This module connects patient identification with prescription management and helps organize interactions between doctors and patient records. Authentication and authorization are essential because sensitive information should not be accessible to every application user. The backend should verify permissions for sensitive operations rather than relying solely on interface restrictions.

C. Hospital Registration

The hospital registration module enables hospitals or healthcare facilities to register their institutional details within the Medicare Prescription System. The registration process collects the required information and stores it in the connected database for subsequent access and management. Hospital registration provides an organizational foundation for connecting hospital-related activities with doctor access, digital prescriptions, pharmacy management, and laboratory services. Validating the submitted information helps reduce incomplete or incorrectly formatted records. The registered hospital information can be used by the application to identify the relevant healthcare facility and organize associated services. Access to registration and institutional record management should be restricted to authorized users. Where multiple hospitals are supported, each facility should be associated with the appropriate records and permissions to prevent unauthorized cross-facility access.

D. Hospital Dashboard

The hospital dashboard provides a centralized interface through which authorized hospital users can access the hospital's available management functions. It serves as a navigation point for hospital-related information and the services supported by the application. Depending on the implemented features, users can access hospital details and manage or navigate to registered pharmacy and laboratory services. The dashboard connects the relevant user interface with backend services so that supported information can be retrieved or updated through the application. Organizing related functions in one location simplifies navigation and helps maintain a consistent workflow across the hospital module. Access to hospital-specific information should be restricted to authorized users, and sensitive operations should be verified by the backend. The dashboard represents the functions actually implemented in the system and should not be described as providing clinical analytics or real-time monitoring unless those features are available.

E. Digital Prescription Management

The digital prescription module supports the electronic creation and retrieval of prescription records. A prescription is associated with the relevant patient and authorized prescriber, helping maintain a structured relationship between patient information and prescription details. Storing these records electronically provides an alternative to maintaining prescriptions only in handwritten or disconnected records. The module communicates with the backend to process supported operations and store information in the database. Proper access controls are necessary to prevent unauthorized viewing or modification. In a production environment, prescription creation and updates should be traceable through audit records. Clinical decisions, medication selection, dosage, and prescription approval remain the responsibility of qualified healthcare professionals.

F. Pharmacy Management

The pharmacy management module provides a structured method for maintaining and retrieving medicine information. Depending on the implemented fields, the catalogue includes medicine name, brand, company, dosage, quantity or count, and availability status. Authorized hospital or pharmacy users can maintain these details through the available application functions. Organizing medicine information in a common database helps users retrieve relevant details and review the availability recorded by the system. This module supports medicine-information management alongside the prescription workflow. However, maintaining a medicine catalogue does not automatically guarantee that physical stock is accurate. A complete inventory-management implementation would additionally require validated stock transactions, update history, and safeguards against inconsistent quantity records.

G. Laboratory Management

The laboratory management module supports laboratory registration and the maintenance of diagnostic test information. The catalogue can contain test names, availability status, cost, and procedure details according to the implemented functionality. These records provide a structured reference for the laboratory services represented in the application. Integrating laboratory information into the same platform helps organize it alongside patient, prescription, pharmacy, and hospital records. The stored details can be maintained and retrieved through the application's supported interfaces. Laboratory test ordering, specimen tracking, test-result entry, and clinician review are separate workflows and should be considered additional capabilities unless they have been implemented and tested. The module manages laboratory service information and does not itself perform diagnostic analysis.

H. Administration and Service Registration

The administration module supports the management of hospital-related information and the registration of associated healthcare services where these functions are implemented. It provides a pathway for organizing institutional details and coordinating pharmacy and laboratory service information within the application. The relevant registration functions help maintain structured records that can be accessed through the supported interfaces. Administrative operations should be restricted to authorized users to reduce the risk of unauthorized changes. If the system supports multiple hospitals or healthcare facilities, institution-level permissions are important to ensure that users access only the information relevant to their responsibilities. This module complements the hospital dashboard by supporting the administrative functions available within the system.

I. Database and Record Management

The database module provides persistent storage for the application's supported records. The system uses Supabase PostgreSQL to maintain structured information associated with patients, doctors, prescriptions, hospitals, pharmacy entries, and laboratory services. Relationships between records help connect the information required by different modules. The backend processes application requests and communicates with the database to create, retrieve, or update records. Structured storage supports consistent record organization and reduces dependence on separate manual files. Appropriate database permissions, validation rules, and access restrictions are important for protecting sensitive information and maintaining data integrity. The database layer serves as the shared information foundation for the connected application modules.

J. API and System Integration

The API integration module connects the frontend with backend services and database operations. The frontend is developed using React and Vite, while Node.js and Express.js provide the backend API layer. REST APIs allow the frontend to send requests and receive responses for supported operations, including patient registration, patient search, prescription management, hospital functions, and pharmacy and laboratory services. The backend processes requests and communicates with the PostgreSQL database. Separating the user interface, backend logic, and data storage helps organize the application into maintainable components. Correct API configuration, input validation, error handling, and server-side permission checks are important for reliable communication between the components.

K. Authentication and Access Control

The authentication and access-control module helps ensure that users sign in before accessing protected application functions. Access restrictions should be applied according to the user's role and the operation being requested. For example, doctor-specific functions should be available only to appropriately authorized users, while hospital registration and administrative operations should be restricted to authorized administrators. Sensitive actions must be checked by the backend because hiding a function in the frontend alone does not provide sufficient protection. Secure session handling, appropriate database permissions, and protection against unauthorized requests are important considerations. Authentication and unauthorized-access checks were included in the system's testing process. Further security assessment, including detailed permission reviews and audit logging, could strengthen the application.

L. Deployment and Application Availability

The deployment module concerns making the application accessible through its hosted components. The frontend is deployed using Vercel, the backend using Render, and GitHub is used for source-code management. The backend connects to the Supabase PostgreSQL database to process supported application operations. Separate deployment of the frontend and backend requires correct API endpoints, environment configuration, and database connection settings. Integration and deployment checks were performed to verify communication between the deployed components and the operation of the implemented modules. Deployment-related issues encountered during testing were addressed, and the application operated as expected during the reported checks. Continued monitoring, maintenance, backups, and security reviews would be important considerations for long-term production use.

Data Model and Information Flow

A relational model can represent patients, users, doctors, prescriptions, hospitals, pharmacies, medicines, laboratories, and laboratory tests as related entities. Primary keys and foreign keys should preserve record relationships, while indexes may support common lookups. A prescription should reference both a patient and an authorized prescriber. Pharmacy and laboratory records should be associated with the correct institution where institution-level separation is required. The actual schema must be checked against the current database before the paper is finalized; this description is a logical design, not a claim that every listed entity exists in the deployed schema.

The expected information flow begins when a patient is registered and receives a unique identifier. An authorized doctor searches for the patient and performs prescription-related operations. Hospital administration maintains the relevant pharmacy and laboratory catalogue data. Users access only the functions and records permitted by their roles. Data moves between the browser, backend API, and database over the configured application interfaces.

VI. ADVANTAGES

1. Reduces paperwork: Replaces manual prescription writing and paper-based record management with digital records.
2. Easy patient registration: Registers patients digitally and generates a unique Patient ID for identification.
3. Quick access to patient details: Doctors can search for patients using their Patient ID or phone number and view their relevant details.
4. Digital prescription management: Doctors can create and store prescriptions electronically, making them easier to access and manage.
5. Improves prescription readability: Digital prescriptions reduce problems caused by unclear or illegible handwriting.

6. Organized medical history: Helps maintain patient and prescription records in a structured manner for easier retrieval.
7. Separate user portals: Provides dedicated interfaces for patients, doctors, and hospitals, supporting their different needs.
8. Pharmacy support: Helps organize medicine-related information and supports pharmacy operations.
9. Laboratory support: Includes laboratory-related information, such as test availability, cost, and procedure details.
10. Improves efficiency: Reduces time spent searching for records and managing prescriptions manually.
11. Better accessibility: A web-based interface allows authorized users to access the system through supported devices and browsers.
12. Supports secure management: Authentication and access controls provide a foundation for protecting sensitive medical records.

VII. TESTING PLAN AND DISCUSSION

A. Functional Testing

Functional testing was performed after deployment to verify the operation of the Medicare Prescription System's implemented modules. The testing process focused on checking whether the application functions behaved as expected and whether the frontend, backend, and database communicated correctly.

Patient registration was tested to verify that patient details could be submitted and that a unique Patient ID was generated. Doctor authentication and patient search were checked to confirm that authorized doctors could access the application and retrieve the relevant patient records. The digital prescription module was tested to verify prescription creation and record handling.

The pharmacy module was tested to check the management and retrieval of medicine information, including medicine name, brand, company, dosage, quantity, and availability. The laboratory module was tested to verify laboratory registration and the management of test details, including test name, availability, cost, and procedure information.

B. Integration and Deployment Testing

Integration testing focused on communication between the React frontend, Node.js/Express backend, and Supabase PostgreSQL database. The application was checked to ensure that frontend requests reached the backend correctly and that database operations supported the implemented functions.

Deployment testing was conducted with the frontend hosted on Vercel and the backend hosted on Render. Configuration settings, API communication, and database connectivity were checked during the verification process. Identified deployment and integration issues were addressed, and the application operated as expected after testing.

C. Security Testing

Security testing focused on authentication and unauthorized access. Login functionality was checked, and access restrictions were tested to verify that unauthorized users could not perform protected operations. These checks support the need for controlled access to patient and prescription information. Further security assessment, including database permission reviews, audit logging, and vulnerability testing, can strengthen the system before use with real patient data.

D. Discussion

The testing process verified the functionality of the implemented modules and their integration in the deployed environment. Connecting patient registration, prescription management, pharmacy information, and laboratory services within one application provides a structured approach to healthcare information management.

The modular architecture also allows individual functions to be maintained and extended independently. However, the reliability of the system depends on accurate data entry, appropriate access permissions, and consistent database operations. Although the application operated as expected during testing, quantitative performance evaluation, formal usability studies, and clinical validation would be required to establish its effectiveness in real-world healthcare environments.

VIII. LIMITATIONS AND FUTURE WORK

The paper has several limitations. First, it does not include a controlled comparison with a paper-based workflow. Second, quantitative performance measures and formal usability results were not supplied for this draft. Third, the deployed status of each requested feature must be verified against the current code and production environment. Fourth, interoperability with external hospital, pharmacy, or laboratory systems has not been established. Finally, security review and clinical validation are outside the evidence currently available.

Future work may include structured usability evaluation; stronger audit logs for sensitive operations; improved pharmacy stock transaction history; laboratory request and result workflows; standardized data exchange using suitable healthcare standards; automated security testing; and documented load, reliability, and recovery tests. Any extension involving real patient data should be preceded by institutional approval and a privacy and security assessment.

IX. CONCLUSION

This paper describes the Medicare Digital Prescription System, a student web application project focused on patient registration, unique patient identification, doctor access, digital prescription handling, pharmacy information, and laboratory service catalogues. The described architecture combines a React frontend, Node.js backend, and Supabase/PostgreSQL data layer. The paper outlines the project motivation, related work, main modules, data

model considerations, security concerns, and a test plan. The contribution is an integrated workflow design rather than a demonstrated clinical outcome. Before publication, the authors should verify implementation details, execute and report actual tests, complete author and institution information, and ensure that all claims are supported by project evidence.

TABLE I. FUNCTIONAL TESTING RESULTS

Test Area	Test Activity	Actual Result
Registration	Test valid and invalid patient details.	Patient registration and input validation worked as expected.
Authentication	Test valid and invalid login attempts.	Authentication worked as expected for the tested cases.
Authorization	Test restricted actions using unauthorized access.	Access restrictions worked as expected in the tested cases.
Patient Search	Search for an existing patient and a nonexistent Patient ID.	Patient search returned the expected results.
Prescription	Create and retrieve a prescription linked to a patient and doctor.	Prescription creation and retrieval worked as expected.
Pharmacy	Create or update medicine details and availability.	Medicine details and availability management worked as expected.
Laboratory	Create or update test availability, cost, and procedure details.	Laboratory and test-catalogue management worked as expected.
Deployment/Database	Check API connectivity, configuration, and database communication.	Frontend, backend, and database integration worked as expected.

X.REFERENCES

[1] World Health Organization, Global Strategy on Digital Health 2020–2025. Geneva, Switzerland: WHO, 2021. [Online]. Available: https://www.who.int/publications/i/item/9789240020924

[2] N. Vejdani et al., "Requirements of electronic prescription systems: A scoping review," BMC Medical Informatics and Decision Making, vol. 23, art. no. 18, 2023, doi: 10.1186/s12911-023-02115-5.

[3] HL7 International, "FHIR Overview." [Online]. Available: https://www.hl7.org/fhir/overview.html. Accessed: Oct. 26, 2026.

[4] OWASP Foundation, "OWASP API Security Top 10—2023." [Online]. Available: https://owasp.org/API-Security/editions/2023/en/0x11-t10/. Accessed: Oct. 26, 2026.

[5] World Health Organization, "Global Strategy on Digital Health 2020–2027," WHO, 2025. Available: https://www.who.int/publications/i/item/9789240116870

[6] Health Level Seven International, "FHIR Release 5: Overview," HL7 International. Available: https://hl7.org/fhir/overview.html

[7] OWASP Foundation, "OWASP API Security Top 10 – 2023," 2023. Available: https://owasp.org/API-Security/editions/2023/en/0x11-t10/

[8] Meta Platforms, "React Documentation: Reference Overview." Available: https://react.dev/reference/react

[9] Express.js, "Express.js Guide." Available: https://expressjs.com/guide/

[10] Supabase, "Database: PostgreSQL Documentation." Available: https://supabase.com/docs/guides/database/overview

[11] Health Level Seven International, "FHIR: Fast Healthcare Interoperability Resources." Available: https://www.hl7.org/fhir/