**Kathmandu University**

**Department of Computer Science and Engineering**

**Dhulikhel, Kavre**



**A Project Report**

**on**

**“KU Health-Care”**

**[Code No: COMP 207]**

**(For partial fulfillment of II Year/ II Semester in Computer Engineering)**

**Submitted by**

**Saimon Neupane (27)**

**Risham Raj Byahut (43)**

**Parikchit Sen (47)**

**Keshav Raj Sharma (48)**

**Prabesh Sharma (49)**

**Submitted to**

**Mr. Suman Shrestha**

**Department of Computer Science and Engineering**

**Submission Date: 2025/08/06**

**Bona fide Certificate**

**This project work on**

**“KU Health-Care”**

**is the bona fide work of**

**“Saimon Neupane” (27)**

**“Risham Raj Byahut” (43)**

**“Parikchit Sen” (47)**

**“Keshav Raj Sharma” (48)**

**“Prabesh Sharma” (49)**

**who carried out the project work under my supervision.**

**Project Supervisor**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Rabindra Bista**

**Associate Professor,**

**Department of Computer Science and Engineering**

**Date: 2025/08/06**

# Acknowledgement

We would like to express our heartfelt gratitude to our Project Supervisor, Dr. Rabindra Bista, for his constant support, thoughtful guidance, and valuable insights throughout the development of the KU Health Care management system. His mentorship played a vital role in shaping our approach to solving complex real-world problems and in ensuring the successful execution of our project.

We are also sincerely thankful to the Department of Computer Science and Engineering (DoCSE) for providing us with the opportunity and platform to work on this impactful project. The development of KU Health Care has been a transformative experience—allowing us to explore concepts in healthcare technology, database design, and user-centric software development.

This project has significantly enhanced our technical and analytical skills and strengthened our ability to work as a cohesive team. From designing structured information flows to integrating automated communication systems, every phase of this journey has deepened our understanding of how technology can improve healthcare service delivery.

We deeply appreciate the support, encouragement, and resources that made this project possible and meaningful.

# Abstract

This project introduces a comprehensive healthcare management system, KU Health Care, designed to streamline patient care workflows and administrative processes within clinical environments. As healthcare institutions face increasing demands for efficiency and coordination across departments, this integrated platform addresses the critical challenges of patient management and appointment scheduling. Our solution employs a centralized database architecture to create a cohesive ecosystem that connects patients, receptionists, doctors, and laboratory staff through automated notification systems and structured information pathways. The platform features robust patient registration capabilities, appointment allocation, and department-specific management tools. By incorporating laboratory test requisition workflows and prescription management, the application eliminates traditional bottlenecks in patient care delivery. The system's architecture supports both clinical decision-making and administrative processes through features such as automated notifications. Beyond mere record-keeping functionality, the platform fosters a collaborative healthcare environment where patient information flows securely between stakeholders. With an emphasis on improving patient experience and operational efficiency, KU Health Care establishes a foundation for enhanced healthcare service delivery, ultimately transforming how medical institutions manage the complex interplay between administrative requirements and quality patient care.

# **Keywords**: *Healthcare Management, Patient Registration, Appointment Scheduling, Laboratory Integration*Table of Content

# List of Figures

# Acronyms/ Abbreviation

**API**: Application Programming Interface

**SQL**: Standard Query Language

**REST**: Representational State Transfer

**ID**: Identification

**UI**: User Interface

**HTTP**: Hypertext Transfer Protocol

**MVC**: Model View Controller

**JSON**: JavaScript Object Notation

**SSD**: Solid State Drive

**UHD**: Ultra High Definition

**SQL**: Standard Query Language

# Chapter 1: Introduction

In the evolving landscape of digital healthcare solutions, building integrated platforms that cater to institutional needs has become increasingly essential. KU Health Care embodies this vision as a comprehensive healthcare management system developed using Java and modern web technologies. This project aims to streamline administrative workflows and enhance coordination among patients, receptionists, doctors, and laboratory staff through a centralized, user-friendly interface. By implementing structured information flow, automated notifications, and robust data handling, KU Health Care delivers a seamless and efficient experience for all stakeholders. Emphasizing scalability, reliability, and accessibility, the system is designed to improve healthcare service delivery while reducing operational burdens commonly found in clinical environments.

## Background

Healthcare institutions are increasingly adopting digital tools to enhance operational efficiency and to respond to the growing demand for patient-centered care. Traditionally, the management of patient data, appointments, prescriptions, and test results was handled manually through physical records or disjointed software systems. These methods are often prone to human error, data duplication, communication lags, and difficulty in accessing records when needed urgently.

While several hospitals and clinics have adopted partial digitization by implementing isolated systems—such as electronic patient registration or digital appointment scheduling—these solutions often operate independently and fail to offer integrated management. The lack of interoperability between systems leads to fragmented care delivery and administrative overload.

Moreover, in many clinical environments, patients are still required to fill out physical forms during each visit, and interdepartmental coordination relies heavily on verbal or manual handovers. Delays in report generation, misplacement of documents, and miscommunication between departments are not uncommon, directly impacting the quality and speed of healthcare services.

KU Health Care aims to resolve these inefficiencies by offering an all-in-one platform where stakeholders can communicate and operate using the same structured and secure information ecosystem. This will not only enhance the patient experience but also empower healthcare professionals with the tools needed to provide accurate and timely care.

## Objectives

The primary goal of this project is to design and implement a digital healthcare management system that supports both clinical and administrative workflows through an integrated web-based application. The specific objectives of the system are outlined as follows:

* To develop a reliable and centralized healthcare management platform that enhances the efficiency of patient care and administrative operations.
* To create a user-friendly web application that allows for easy navigation by patients, doctors, receptionists, and lab staff.
* To automate routine processes such as appointment scheduling, doctor assignment, and lab report management through intelligent notification systems.
* To ensure secure and accurate maintenance of patient data, including registration details, consultation records, prescriptions, and laboratory reports.

## Motivation and Significance

The motivation behind KU Health Care arises from the inefficiencies in traditional hospital workflows, where different departments often work in isolation, leading to delays, communication gaps, and poor patient experience. Existing systems are either limited in scope or lack integration, making healthcare management cumbersome.

Our project aims to address these challenges by developing a centralized platform that connects patients, receptionists, doctors, and laboratory staff. Through automation and streamlined communication, the system enhances coordination, reduces administrative burden, and improves the overall quality of patient care.

This solution is significant as it promotes a more organized, efficient, and tech-driven approach to healthcare, especially in environments still reliant on partial or manual systems.

# Chapter 2: Related Works

## 2.1 Epic Systems

Epic provides a comprehensive healthcare management platform with integrated workflows across departments, similar to your centralized approach. Its strength lies in connecting patient records seamlessly between reception, clinical staff, and laboratories through a unified database architecture.

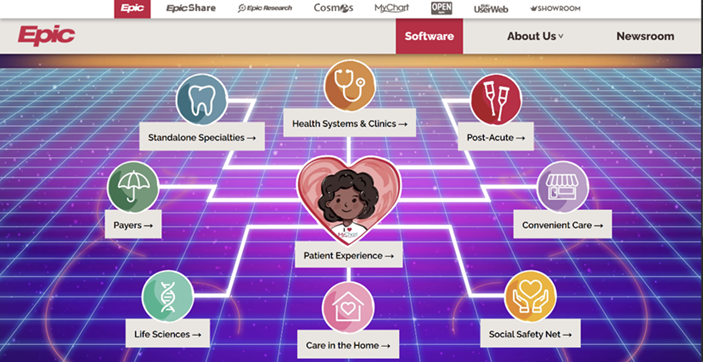


Figure 2.1.1 Epic System Software

## 2.2 OpenMRS

OpenMRS is an open-source electronic medical record system designed primarily for use in resource-constrained environments. It offers a modular architecture that enables customization according to institutional needs, making it suitable for small clinics, research environments, and university health centers. OpenMRS supports patient registration, clinical observations, encounter tracking, and data reporting, which aligns well with KU Health Care’s goals of building a flexible and scalable health management platform tailored to a specific community.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 2.2.1 OpenMRS Online Demo

## 2.3 Bahmni

Bahmni is an open-source hospital information system built on top of OpenMRS, designed specifically for low-resource healthcare environments. It integrates electronic medical records, laboratory management, pharmacy, billing, and radiology into a unified platform. Bahmni provides an online demo environment that allows users to explore its full suite of clinical workflows in a real-time setting. Its modular and user-friendly design makes it a valuable reference for developing scalable and comprehensive health care solutions such as KU Health Care.

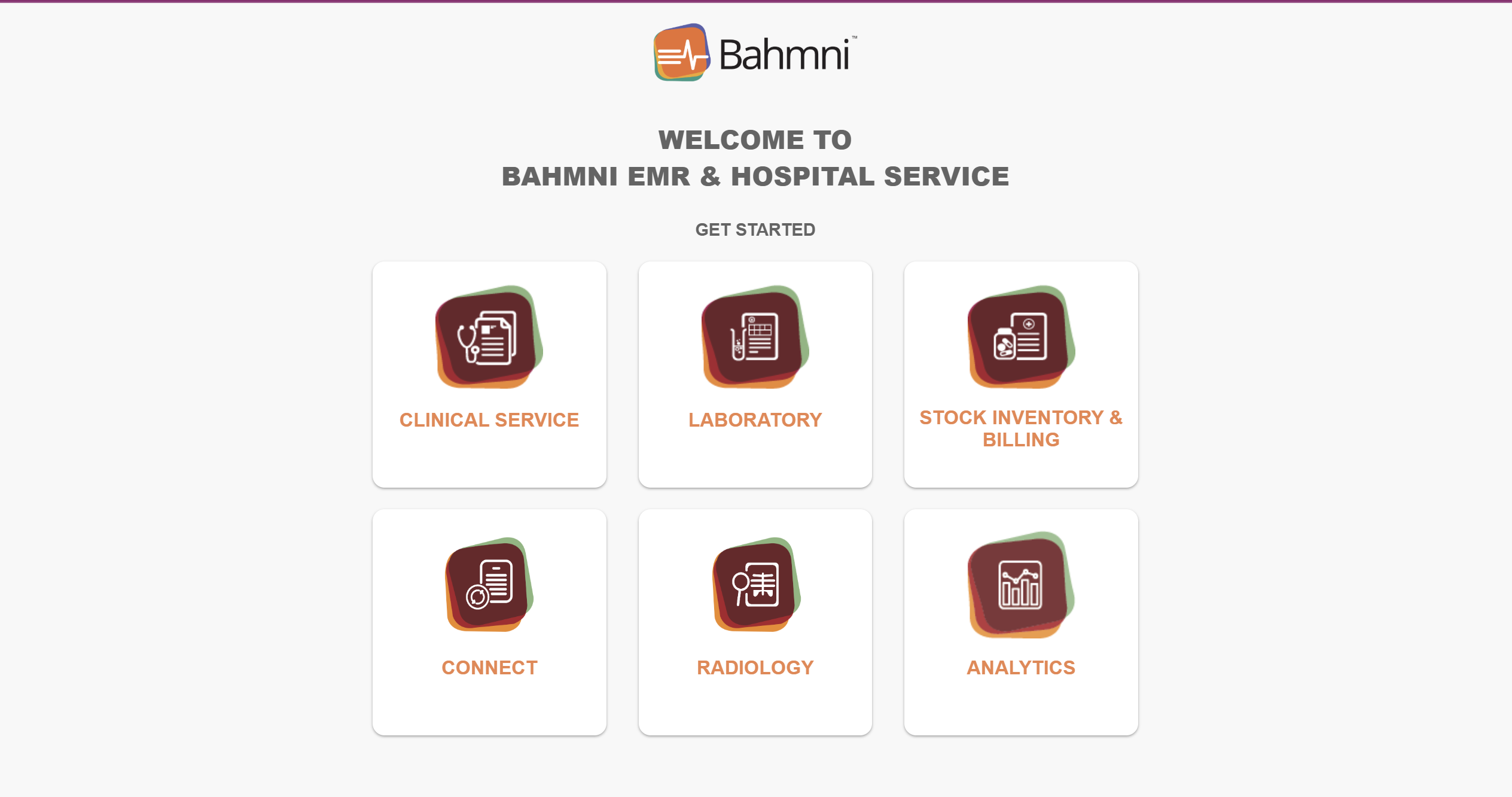


Figure 2.3.1 Bahmni Online Demo

# Chapter 3: Design and Implementation

This chapter outlines the step-by-step process and methodologies followed throughout the development of the KU Health Care project.

Following are the steps we implemented to complete this project:

**Step 1. Project Planning:**

* The project work was planned by dividing tasks among team members according to their expertise in frontend and backend development.
* Clear milestones and deadlines were set to ensure timely progress and continuous collaboration.

**Step 2. System Design and Architecture:**

* The system architecture was designed after gathering and finalizing the project requirements. It includes defining the database schema, user interface components, and the overall modular structure of the application.
* Data flow and interactions between components, such as patient management, appointment scheduling, and medical records handling, were mapped out through flowcharts and system diagrams.

**Step 3. Development and Coding:**

* The frontend was developed using React, leveraging JavaScript (90%) and TypeScript(10%) to create a responsive and user-friendly interface.
* For the backend and database, PostgreSQL was used to implement a robust relational database system to manage patient data securely and efficiently.
* API endpoints and business logic were built to support core functionalities such as user authentication, appointment management, and data retrieval.

**Step 4. Testing and Debugging:**

* Comprehensive testing was performed, including unit testing of individual components, integration testing to verify interactions between modules, and user acceptance testing to ensure the system met user expectations.
* Bugs and issues were tracked continuously, with fixes applied iteratively to maintain system stability and performance.

**Step 5. Documentation:**

* Detailed documentation was prepared, including user manuals to guide end-users and technical documentation to support future development and maintenance.

Throughout the project, continuous communication and collaboration among the development team were maintained to align with project goals and deliver a reliable healthcare platform tailored to the Kathmandu University community.

## Chapter 3.1 System Requirement Specifications

### 3.1.1 Software Specifications

1. Code Editor: Visual Studio Code (VS Code) was used as the primary code editor for writing, organizing, and managing the entire project. Its lightweight nature and extensive extension support made development efficient.

2. Front-end Library: React was used for building the user interface of the system. It allows the development of dynamic and interactive web applications through a component-based structure.

3. Backend Framework: Express.js, a lightweight and flexible Node.js framework, was used to create RESTful APIs for handling server-side logic and communication with the database.

4. System for Database Management (DBMS): PostgreSQL was used for managing and storing healthcare-related data. It is a robust relational database system that supports complex queries and ensures data integrity.

5. Tools for Documentation: Microsoft Word was used for documentation and record-keeping purposes, including technical reports and user manuals.

### 3.1.2 Hardware Specifications

The system is designed to be accessible from various client devices,

including desktop computers, laptops and smartphones. Ensure that the user

interface is responsive and compatible with different screen sizes and

resolutions.

Minimum Specification:

* Processor: Intel Core i5 or equivalent
* RAM: 8GB
* Storage: 256 GB SSD
* Graphics Card: Integrated Intel UHD Graphics
* Operating System: Windows 8, macOS or Linux Based OS

# Chapter 4: Working Mechanism

The working mechanism of the KU Health Care system is explained below, incorporating both frontend and backend components that ensure a seamless and functional healthcare service platform.

1. Frontend Interface

* User Interface: The frontend of KU Health Care is developed using React, providing a dynamic and interactive interface for users such as patients, doctors, and administrative staff. Key modules include appointment booking, health record viewing, user registration, and profile management.
* Responsive Design: The application is designed to be fully responsive, ensuring accessibility across devices including desktops, tablets, and smartphones.

2. Backend Architecture

Express.js and PostgreSQL Components:

* Models (Database Schema):

Defined using SQL and implemented in PostgreSQL to represent essential entities in the healthcare system:

* + User Model: Stores user credentials and roles (Patient, doctor, admin).
  + Appointment Model: Stores appointment details (date, time, doctor, patient).
  + Medical Record Model: Stores patient health records including symptoms, diagnoses, and prescriptions.
* API Routes (Controllers):

Built using Express.js to manage HTTP requests and business logic:

* Auth Routes: Handle user registration, login, and secure authentication.
* Appointment Routes**:** Allow users to schedule, update, or cancel appointments.
* Record Routes: Enable doctors to create and retrieve patient records.
* Feedback Routes: Allow users to submit feedback on their experience.

3. User Interaction Flow

Registration and Authentication:

* Registration: Users sign up with name, email, password, and role selection.
* Authentication: Secure login and logout mechanisms allow access to personalized dashboards.

Appointment Management Flow:

* Appointment Booking: Patients browse available slots and book appointments with doctors.
* Appointment Confirmation**:** Users receive confirmation details after successful booking.
* Appointment History**:** Patients and doctors can view upcoming and past appointments.

Health Record Access:

* Record Viewing**:** Patients can access their own medical history securely.
* Record Entry**:** Doctors can input and update records after consultations.

Feedback System:

* Feedback Submission: Users can provide feedback on services for quality improvement.
* Feedback Storage: Feedback entries are stored and can be reviewed by administrators.

**4. Integration with Email Notification System**

* **Notification Triggers:** Automatic email notifications are sent for:
  + Appointment confirmations and reminders.
  + Password resets and registration confirmation.
* **SMTP Integration:** Uses a secure email service (e.g., Nodemailer with Gmail or another SMTP provider).

**5. Security Measures**

* **User Authentication:** Implemented using JWT (JSON Web Tokens) and secure password hashing to protect user credentials.
* **Data Protection:** Sensitive medical data is securely stored with access controls based on user roles.
* **Input Validation:** All user inputs are validated on both client and server to prevent injection and other common vulnerabilities.

**6. Scalability and Performance**

* **Efficient Querying:** PostgreSQL indexes and optimized queries are used to ensure fast data retrieval.
* **Modular Design:** The system follows a modular architecture, allowing future scaling of features such as teleconsultation or prescription uploads.
* **Stateless APIs:** RESTful API design allows horizontal scalability and better performance under load.