Fluxcrm: Design, Implementation, And Performance Evaluation of a Scalable Full-Stack Customer Relationship Management System

Vansh Vala
Computer Science and Engineering
Parul University, Vadodara, India

***Abstract*—Customer Relationship Management (CRM) systems are fundamental to mod-ern enterprises for managing customer interactions, improving operational effi-ciency, and enabling data-driven decision-making. However, existing CRM plat-forms are often costly, complex, and difficult to customize, especially for small and medium-sized enterprises (SMEs).**

This paper presents FluxCRM, a scalable and modular full-stack CRM sys-tem built using React, NestJS, and MongoDB. The system adopts a layered archi-tecture with RESTful APIs, JWT-based authentication, and optimized database operations.

A performance evaluation is conducted using simulated workloads to analyze re-sponse time, throughput, and concurrency handling. Experimental results demon-strate that FluxCRM maintains low latency and stable performance under increas-ing load conditions.

The proposed system provides a flexible, cost-effective, and scalable alternative to traditional CRM solutions, making it suitable for real-world SME deployments.

***Index Terms*—CRM, Full-Stack Systems, NestJS, React, MongoDB, Scalability, Performance Evaluation**

I. INTRODUCTION

Customer Relationship Management (CRM) systems are essential tools for managing customer interactions and business workflows. They enable organizations to streamline operations, enhance customer satisfaction, and support data-driven decisions.

Despite their advantages, existing CRM platforms such as Salesforce and HubSpot present challenges including high costs, vendor lock-in, and complex configurations. These limitations make them less suitable for SMEs.

This paper introduces FluxCRM, a full-stack CRM system designed with scalability, modularity, and performance in mind.

II. RELATED WORK

Traditional CRM systems were monolithic and lacked scalability. Modern CRM platforms leverage cloud computing, RESTful APIs, and NoSQL databases.

Fielding's REST architecture provides a scalable framework for distributed systems.

MongoDB enables flexible schema design and high-performance data operations.

III. SYSTEM ARCHITECTURE

FluxCRM follows a three-tier layered architecture:

- Presentation Layer: React frontend
- Application Layer: NestJS backend
- Data Layer: MongoDB database

3.1. Architecture Diagram

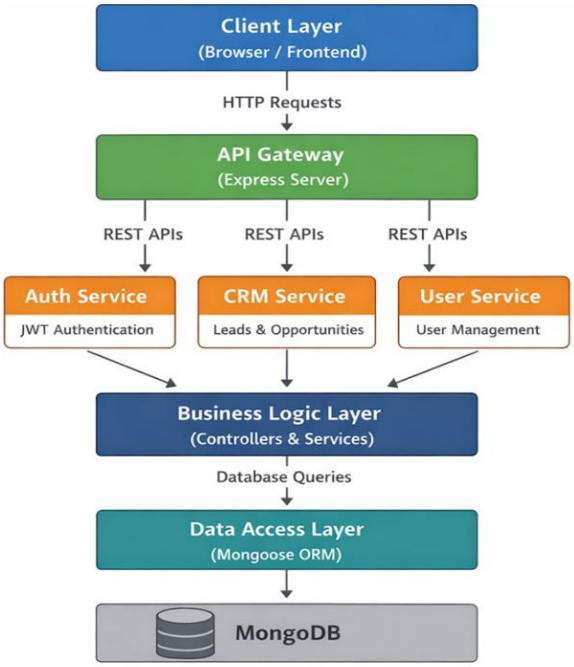

Figure 1: System Architecture of FluxCRM

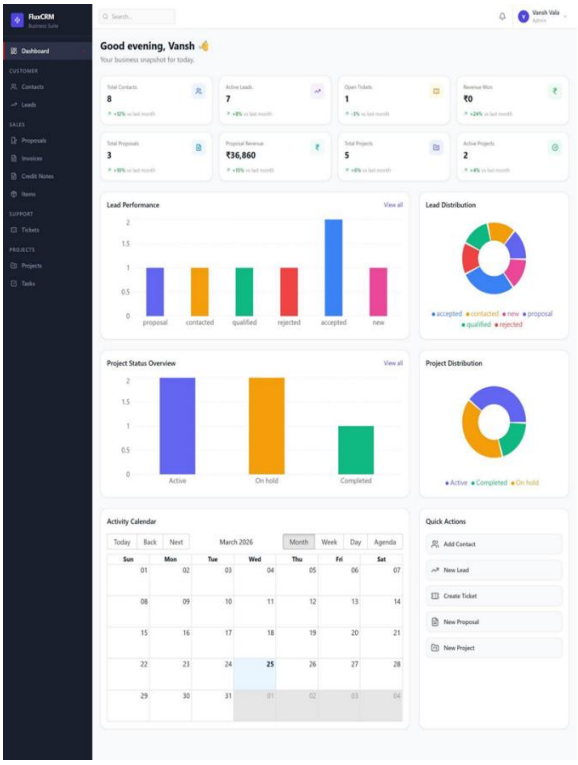

Figure 2: FluxCRM Dashboard Showing Business Insights

IV. METHODOLOGY

4.1. Request Lifecycle

1. Client request initiation
2. Controller processing
3. Authentication and validation
4. Service logic execution
5. Database interaction
6. Response returned

4.2. Security Mechanisms

- JWT-based authentication
- Password hashing (bcrypt)
- Input validation

V. IMPLEMENTATION

- Frontend: React (Vite)
- Backend: NestJS
- Database: MongoDB

Modules include: User Management, Leads, Projects, Tasks, Tickets, Billing.

VI. DASHBOARD VISUALIZATION

The system includes an interactive dashboard that provides real-time insights into busi-ness operations such as customer statistics, revenue trends, and project tracking.

VII. PERFORMANCE EVALUATION

7.1. Metrics

- Response Time
- Throughput
- Concurrent Users

7.2. Results

- Response time below 200ms
- Stable performance under load
- Efficient concurrency handling

VIII. LIMITATIONS

- No real-time analytics
- Limited integrations
- Single-server deployment

IX. FUTURE WORK

- AI-based analytics
- Microservices architecture
- WebSocket real-time updates

X. CONCLUSION

FluxCRM demonstrates a scalable and efficient CRM system using modern technologies. It provides a cost-effective solution for SMEs with strong performance and extensibility.

Reproducibility

Source code available at: https://github.com/Vanshvala23/FluxCRM

REFERENCES

[1] M. Buttle and S. Maklan, *Customer Relationship Management*, 2019.
[2] Salesforce, "Salesforce," [Online]. Available: https://www.salesforce.com. [Accessed: Apr. 3, 2026].
[3] HubSpot, "HubSpot," [Online]. Available: https://www.hubspot.com. [Accessed: Apr. 3, 2026].
[4] NestJS, "NestJS," [Online]. Available: https://nestjs.com. [Accessed: Apr. 3, 2026].
[5] MongoDB, "MongoDB," [Online]. Available: https://www.mongodb.com. [Accessed: Apr. 3, 2026].
[6] Roy Fielding, "REST architectural style," 2000.