CUSTOMER RELATIONSHIP MANAGEMENT SYSTEM

Narenkumar.C, Student, Department of BCA DevOps, Rathinam College of Arts and Science, Coimbatore

Mr. N. Thamizharasan, Assistant Professor, Department of BCA DevOps, Rathinam College of Arts and Science, Coimbatore

Abstract

Customer Relationship Management (CRM) systems are essential for managing customer interactions, sales processes, and customer satisfaction. This project proposes a centralized CRM platform that automates customer data management, lead tracking, task scheduling, and reporting. The system reduces manual work, improves communication, enhances productivity, and supports data-driven decision-making. Developed using HTML, CSS, JavaScript, Node.js/PHP, and MySQL/MongoDB, the system provides a scalable and user-friendly solution for modern businesses.

Keywords

CRM, Customer Management, Lead Tracking, Task Scheduling, Reporting, Web Application

1. INTRODUCTION

Customer Relationship Management is an important business strategy that helps organizations centralize customer information, improve communication, and increase operational efficiency. Traditional systems based on spreadsheets or disconnected tools are inefficient and error-prone. The proposed CRM system offers automation, centralized storage, and real-time updates.

2. PROBLEM STATEMENT

Businesses face challenges such as scattered customer records, poor lead management, delayed follow-ups, communication gaps, and lack of analytical insights. Existing systems increase workload and reduce productivity. The proposed CRM system solves these issues by integrating customer management, lead tracking, task automation, and reporting into one web-based platform.

3. SYSTEM ANALYSIS

The CRM system analyzes customer data, tracks leads through sales pipelines, schedules tasks, and generates reports. It ensures accurate data management, workflow automation, and improved coordination between departments.

4. SYSTEM DESIGN

The system follows a three-tier architecture:

• Presentation Layer – HTML, CSS, JavaScript

• Application Layer – Node.js / PHP

• Data Layer – MySQL / MongoDB

It also follows MVC architecture for separation of frontend, backend, and database operations.

SYSTEM ARCHITECTURE OVERVIEW

Figure 1: System Architecture Diagram

Frontend → Backend → Database

Figure 2: Use Case Diagram

User Login → Customer Management → Lead Management → Task Scheduling → Reports

Figure 3: Level 1 DFD – Data Flow

User Input → Authentication → Data Processing → Database → Dashboard Output

Diagram Note

This structure ensures centralized management, automation, security, and efficient business process flow.

5. RESULTS

The CRM system successfully manages customer records, tracks sales leads, automates reminders, and generates analytical reports. Testing confirms improved data accuracy, reduced manual work, faster response time, and better customer service. The user-friendly interface ensures smooth navigation and efficient operation.

Key Achievements

• Centralized customer database

• Improved lead conversion

• Automated workflow

• Better communication

• Real-time reporting

6. CONCLUSION

The Customer Relationship Management system provides an efficient, scalable, and secure solution for managing customer interactions and business processes. It improves productivity, reduces operational errors, and supports organizational growth.

FUTURE ENHANCEMENTS

• AI-based customer prediction

• Mobile application support

• Third-party integrations

• Advanced analytics dashboard

REFERENCES

[1] CRM System Development Concepts

[2] Web Application Development using Node.js/PHP

[3] MySQL/MongoDB Documentation

[4] Business Process Automation Techniques

[5] Customer Data Analytics Methods