ACKNOWLEDGEMENT

I would like to express my sincere gratitude to **Amnil Technologies Pvt. Ltd**. for providing me with the opportunity to intern as a Node.js developer. I am especially grateful to my mentor, **Subin Joshi**, for his invaluable guidance and support throughout my internship. His expertise and feedback have played a crucial role in my development and understanding of backend technologies.

I also convey my sincere gratitude to my academic supervisor **Pesal Rai** of **Asian College of Higher Studies**. Without his kind direction and proper guidance this study would have been a little success. In every phase of the project his supervision and guidance shaped this report to complete perfection.

I would also like to express my heartfelt gratitude to all the people who have directly or indirectly assisted me in the completion of this internship report. Their contributions are sincerely appreciated and acknowledged.

Thank you all for your invaluable contributions.

Puran Dahal
TU Reg: 6-2-1177-27-2020

ABSTRACT

The Sales Cart & Reporting System is a web application developed during my internship at Amnil Technology Pvt. Ltd., designed to manage sales transactions and generate insightful reports using Node.js, MongoDB, Express, HTML, CSS, and JavaScript. This system provides an efficient solution for businesses to track product sales, monitor revenue, and analyze customer purchasing trends.

The application includes a dynamic sales cart where users can browse products, add items to the cart, and complete purchases. It also features a sales reporting module that offers detailed insights into sales performance through day-wise, monthly, and yearly reports, utilizing data visualization techniques such as bar charts and pie charts.

The backend is built using Node.js and Express.js, ensuring a fast and scalable architecture. MongoDB is used for database management, leveraging Mongoose ODM for seamless data handling. The system also integrates Redis caching for improved performance and Swagger API documentation for easy API management.

With a user-friendly frontend built using HTML, CSS, and JavaScript, the Sales Cart & Reporting System offers an intuitive interface for both customers and administrators. The admin panel allows for CRUD operations on products, detailed sales analysis, and exporting reports in multiple formats (CSV, Excel, PDF).

This project demonstrates expertise in full-stack web development, API integration, and modern web-based reporting solutions, contributing to enhanced sales tracking and business intelligence.

Keywords:

Sales Cart, Sales Report, AMNIL Technology Pvt. Ltd., Node.js, Express.js, MongoDB, Mongoose, Redis, Swagger, HTML, CSS, JavaScript, Data Visualization, API Integration, Bar Charts, Pie Charts, CSV Export, Excel Export, PDF Reports, Business Intelligence, Full-Stack Web Development

Contents

CHAPTER 1: INTRODUCTION

1.1. INTRODUCTION:

During my internship at AMNIL Technology Pvt. Ltd., I had the opportunity to develop a Sales Cart & Reporting System using Node.js, Express.js, MongoDB, HTML, CSS, and JavaScript. This project focuses on implementing essential sales management functionalities, including a shopping cart system, sales tracking, and dynamic reporting. The system provides businesses with a streamlined solution to manage product sales, track customer purchases, and generate insightful sales reports for better decision-making.

The application features a user-friendly frontend where customers can browse products, add items to their cart, and complete purchases seamlessly. Additionally, an admin panel enables efficient CRUD operations on products, monitoring of sales trends, and generation of detailed reports such as daily, monthly, and yearly sales analytics. The integration of bar charts and pie charts enhances data visualization, making it easier to analyze revenue and top-selling products.

To optimize performance, the backend is built using Node.js and Express.js, while MongoDB is utilized as the primary database with Mongoose ODM for effective data management. The application also incorporates Redis caching for faster data retrieval and Swagger API documentation for seamless API management and testing. These modern technologies ensure a robust, scalable, and high-performance system for businesses of all sizes.

This project aims to simplify sales tracking, improve business intelligence, and enhance user experience through an interactive and data-driven approach. It serves as a valuable tool for businesses seeking to automate their sales processes, analyze customer trends, and make data-driven decisions effectively. By leveraging full-stack web development technologies, this system contributes to modernizing e-commerce and sales reporting solutions.

1.2. Problem statement:

Managing sales data efficiently is crucial for businesses in various industries, including retail, e-commerce, and logistics. However, traditional methods of tracking sales, managing carts, and generating reports manually can be time-consuming, error-prone, and inefficient, especially when handling large datasets and real-time transactions. Businesses require a streamlined, automated system that ensures accurate sales tracking, real-time cart management, and insightful reporting. There is a need for a user-friendly, scalable, and

data-driven solution that optimizes sales operations and provides meaningful analytics to support better decision-making.

1.3. Objective:

- To develop a comprehensive sales management system using Node.js, MongoDB, Express, HTML, and CSS.
- To implement features for efficient sales tracking, cart management, and sales reporting.
- To allow users to add, update, and manage sales transactions dynamically.
- To generate insightful sales reports with meaningful analytics for better decision-making.

1.4. Scope and Limitation:

Scope:

- The project focuses on developing a sales cart and reporting system using Node.js, MongoDB, Express, HTML, and CSS.
- It provides an interactive user interface for managing product sales, tracking transactions, and generating detailed sales reports.
- The system allows users to input sales data, track stock availability, and monitor sales performance dynamically.
- The application is designed to handle large datasets efficiently using caching, MongoDB full-text search, and lazy loading techniques.
- The project enhances user experience by providing real-time analytics, sales trends, and visual reports to support informed decision-making.
-

Limitation:

- The system is limited to tracking individual product sales and does not support complex business inventory management.
- The accuracy of sales reports depends on proper data entry by users, as incorrect inputs may affect analytics.
- The system requires an active MongoDB database and Redis caching for optimal performance, which may require technical setup.
- The project is designed primarily for web-based access and does not include a mobile application version.
- The application does not support advanced machine learning-based sales predictions,

focusing instead on fundamental analytics and reporting.

1.5. Report Organization:

The report is organized into five chapters.

Chapter 1: The introduction to the project assigned during the internship and the work done is briefly introduced. The problem statement defines the problem that was being targeted by the project. It also mentions the different objectives of the project/work. The scope and limitations describe the targeted audience and goals for the project but also the goals that the project fails to meet.

Chapter 2: It starts off with the introduction to the organization and carries on to describe the organization's hierarchy. It also describes the different domains that the organization is working in. Then it goes on to the details about the department or unit that the intern worked in. Finally, the section has a literature review/ related study that evaluates findings of different papers that has been relevant for research during this internship project work.

Chapter 3: The section consists of the analysis of the current system the roles and responsibilities assigned during the internship are mentioned. It also displays the weekly log of the technical activities performed during the internship period. It also includes the description of the projects involved during the internship and the different tasks and activities performed during the time span.

Chapter 4: In chapter four, it has the conclusion for the report. It restates the topic, internship statement and summarizes the whole document. Finally, since this is an internship report, it is necessary to write the learning outcomes from this internship period.

Chapter 5: The final chapter is the conclusion section which includes the learning outcomes from the internship.