**Vietnam General Confederation of Labor**

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**WEB APPLICATION DEVELOPMENT USING NODEJS**

**Point Of Sale**

*Instructor*: **Mr. MAI VAN MANH**

*Student*: **Do Minh Quan – 521H0290**

**Ho Huu An – 521H0489**

**Tran Nhut Anh – 521H0491**

*Class* **: 21H50301**

*Year* **: 25**

**HO CHI MINH CITY, 2023**

Vietnam General Confederation of Labor

**TON DUC THANG UNIVERSITY**

**FACULTY OF INFORMATION TECHNOLOGY**



**FINAL REPORT**

**WEB APPLICATION DEVELOPMENT USING NODEJS**

**Point Of Sale**

*Instructor*: **Mr. MAI VAN MANH**

*Student*: **Do Minh Quan – 521H0290**

**Ho Huu An – 521H0489**

**Tran Nhut Anh – 521H0491**

*Class* **: 21H50301**

*Year* **: 25**

**HO CHI MINH CITY, 2023**

ACKNOWLEDGEMENT

I would like to express my deep appreciation to Mr. Mai Van Manh from Ton Duc Thang University for his invaluable assistance and guidance throughout the development of this web application. His expertise in the field of nodeJS has played a crucial role in shaping the direction of this project.

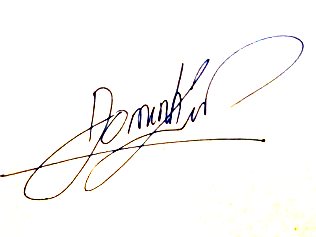
Mr. Mai Van Manh's unwavering commitment to excellence and his openness to share his knowledge have greatly contributed to the success of this endeavor. His mentorship has not only improved my comprehension of utilizing Node.js in web application development but has also motivated me to explore new avenues in logic and web deployment.

I sincerely thank Mr. Mai Van Manh for his continuous encouragement and support, which have been pivotal in the completion of this project. His dedication to creating a conducive learning environment has been a source of inspiration, and I am grateful for the opportunity to work under his guidance

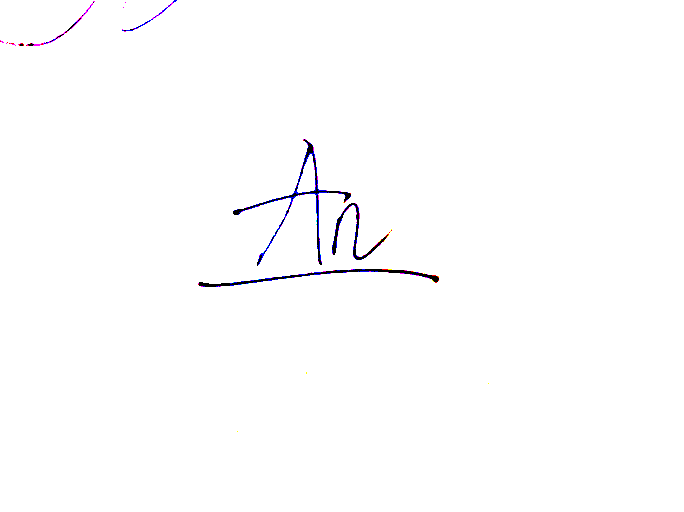
*Ho Chi Minh city, 10th December, 2023*

*Author*

*(Sign and write full name)*

**

*Do Minh Quan*

**

*Ho Huu An*



*Tran Nhut Anh*

**THIS PROJECT WAS COMPLETED AT**

**TON DUC THANG UNIVERSIY**

I fully declare that this is my own project and is guided by Mr. Mai Van Manh; The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

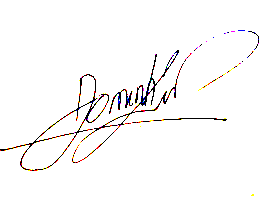
Besides that, the project also uses a number of comments, assessments as well as data from other authors, other agencies and organizations, with citations and source annotations.

**Should any frauds were found, I will take full responsibility for the content of my report.** Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).

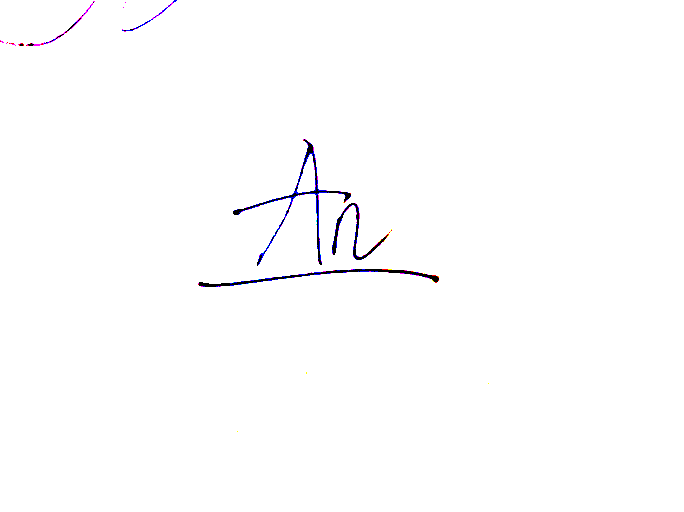
*Ho Chi Minh city, 10th December, 2023*

*Author*

*(Sign and write full name)*

**

*Do Minh Quan*

**

*Ho Huu An*



*Tran Nhut Anh*

CONFIRMATION AND ASSESSMENT SECTION

**Instructor confirmation section**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh December, 2023*

*(Sign and write full name)*

**Evaluation section for grading instructor**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Ho Chi Minh December 2023*

*(Sign and write full name)*

SUMMARY

The Point of Sale (POS) application requires features such as account management, product catalog management, customer management, transaction processing, and reporting/analytics. The account management feature allows administrators to create and manage accounts for salespeople, while salespeople can update their profiles and change passwords. The product catalog management feature enables administrators to add, update, and delete products, while salespeople can view the product list without editing capabilities. The customer management feature automatically creates customer accounts during checkout and provides access to customer information and purchase history. The transaction processing feature facilitates the entry of purchased products, calculates totals, and allows for customer information input. Finally, the reporting and analytics feature allows users to view sales results by different time periods, including total amounts received, order numbers, product numbers, and detailed order information. Administrators have access to additional information such as total profit.

INDEX

[ACKNOWLEDGEMENT i](#_Toc153718499)

[CONFIRMATION AND ASSESSMENT SECTION iii](#_Toc153718500)

[SUMMARY iv](#_Toc153718501)

[LIST OF ABBREVIATIONS 3](#_Toc153718502)

[LIST OF TABLES, DRAWINGS, GRAPHICS 4](#_Toc153718503)

[CHAPTER 1 – OVERVIEW REQUIREMENT 5](#_Toc153718504)

[1.1 Introduction 5](#_Toc153718505)

[1.2 Practical application 5](#_Toc153718506)

[1.3 Use Cases 5](#_Toc153718507)

[CHAPTER 2 – PRESENTS THE APPLICATION, LIBRARIES, SERVICES, TECHNOLOGIES 7](#_Toc153718508)

[2.1 The technologies 7](#_Toc153718509)

[2.1.1 Node JS 7](#_Toc153718510)

[2.1.2 Express JS 8](#_Toc153718511)

[2.2 The librares 8](#_Toc153718512)

[2.2.1 bcrypt (5.1.1): 8](#_Toc153718513)

[2.2.2 body-parser (1.20.2): 8](#_Toc153718514)

[2.2.3 cookie-parser (1.4.6): 9](#_Toc153718515)

[2.2.4 dotenv (16.3.1): 9](#_Toc153718516)

[2.2.5 express (4.18.2): 9](#_Toc153718517)

[2.2.6 express-flash (0.0.2): 9](#_Toc153718518)

[2.2.7 express-handlebars (7.1.2): 9](#_Toc153718519)

[2.2.8 express-session (1.17.3): 9](#_Toc153718520)

[2.2.9 express-validator (7.0.1): 10](#_Toc153718521)

[2.2.10 handlebars (4.7.8): 10](#_Toc153718522)

[2.2.11 jsonwebtoken (9.0.2): 10](#_Toc153718523)

[2.2.12 moment (2.29.4): 10](#_Toc153718524)

[2.2.13 moment-timezone (0.5.43): 10](#_Toc153718525)

[2.2.14 morgan (1.10.0): 10](#_Toc153718526)

[2.2.15 multer (1.4.5-lts.1): 10](#_Toc153718527)

[2.2.16 mysql2 (3.6.3): 11](#_Toc153718528)

[2.2.17 nodemailer (6.9.7): 11](#_Toc153718529)

[2.2.18 nodemon (3.0.1): 11](#_Toc153718530)

[2.2.19 sequelize (6.34.0): 11](#_Toc153718531)

[2.3 Service 11](#_Toc153718532)

[2.4 General User Interface 15](#_Toc153718533)

LIST OF ABBREVIATIONS

EC2: Elastic Compute Cloud

S3: Simple Storage Service

RDS: Relational Database Service

IAM: Identity and Access Management

JWT: JSON Web Token

HTTP: Hypertext Transfer Protocol

API: Application Programming Interface

HTML ORM: Object-Relational Mapping in HTML

AWS: Amazon Web Services

LIST OF TABLES, DRAWINGS, GRAPHICS

**List of figures**

[Figure 1: General Use Cases 6](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718712)

[Figure 2: Logo of Node.js 7](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718713)

[Figure 3: Logo of ExpressJS 8](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718714)

[Figure 4: Add Security Rule for SSH Connection 12](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718715)

[Figure 5: Installation for node application 12](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718716)

[Figure 6: MySQL configuration 13](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718717)

[Figure 7: Environment Variables for Node Application 13](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718718)

[Figure 8: Configure nginx for forwarding to node application 14](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718719)

[Figure 9: Using pm2 to start up application 14](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718720)

[Figure 10: Login Interface 15](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718721)

[Figure 11: DashBoard Interface 15](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718722)

[Figure 12: View List of Products Interface 16](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718723)

[Figure 13: View List of Users Interface 16](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718724)

[Figure 14: View List of Customers Interface 17](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718725)

[Figure 15: Sales Interface 17](file:///C:\Users\An\Desktop\521H0290_DoMinhQuan_521H0489_HoHuuAn_521H0491_TranNhutAnh.docx#_Toc153718726)

**List of tables**

[Table 1: User Interface Description 17](#_Toc153718727)

CHAPTER 1 – OVERVIEW REQUIREMENT

* 1. Introduction

The project introduces the development of a comprehensive web application tailored for a retail phone store in Vietnam. Built using Node.js, Express, and Sequelize, the application prioritizes Point of Sale (POS) functionality, catering to the specific needs of salespeople and administrators. This summary highlights key features, including secure account management, intuitive product catalog management, streamlined customer interactions, efficient transaction processing, and robust reporting and analytics. The choice of modern technologies ensures scalability, maintainability, and a user-friendly experience, making it a well-rounded solution for retail store management.

* 1. Practical application

The practical application of a Point of Sale (POS) system is to facilitate and streamline transactions in retail environments. It serves as a centralized platform for processing sales, managing inventory, and tracking various aspects of business operations. A POS system is used in diverse industries, such as retail stores, restaurants, and hospitality, providing a convenient and efficient way to handle transactions, generate receipts, and maintain accurate records.

* 1. Use Cases

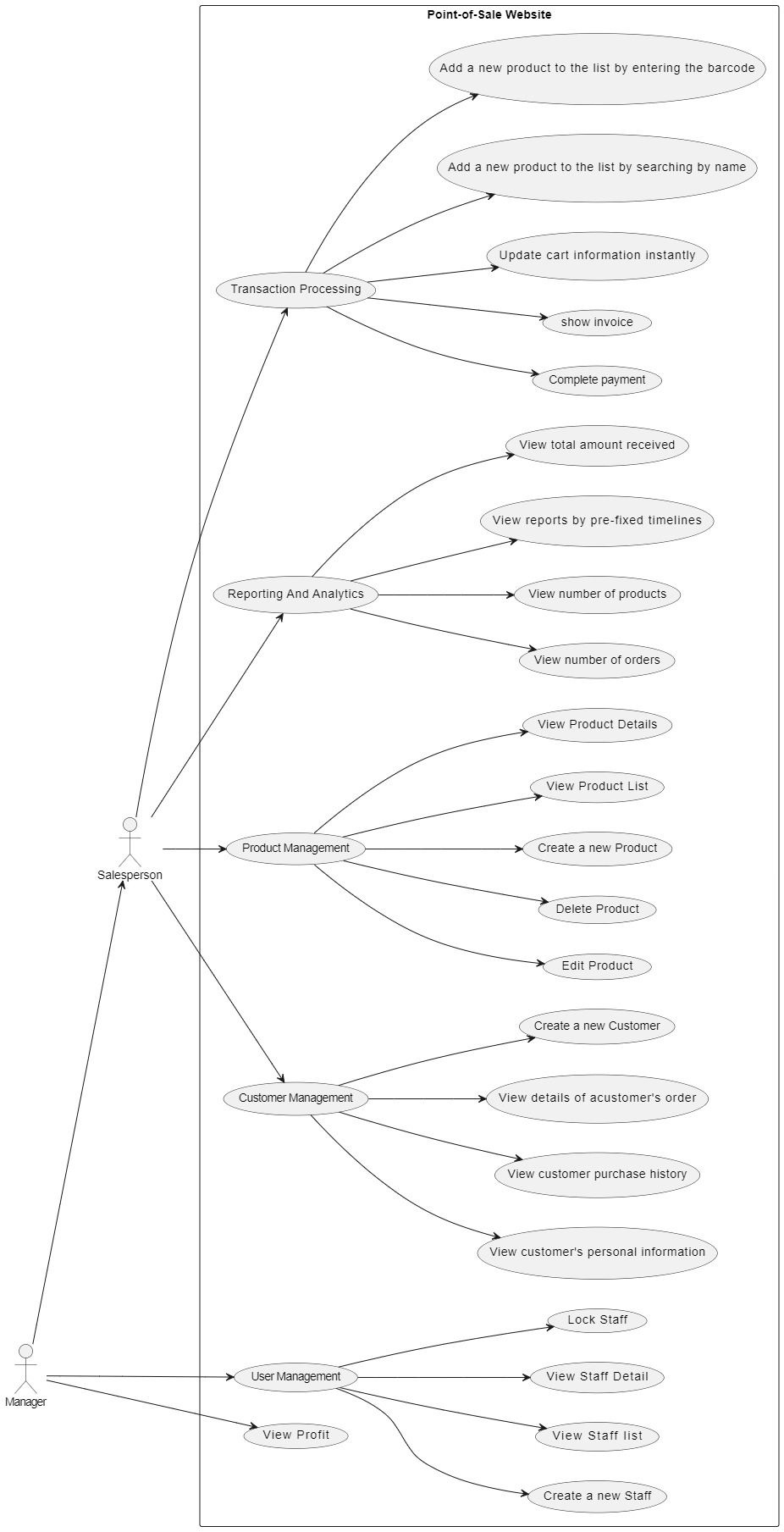


Figure : General Use Cases

CHAPTER 2 – PRESENTS THE APPLICATION, LIBRARIES, SERVICES, TECHNOLOGIES

1. The technologies

2.1.1 Node JS

Node.js is a server-side JavaScript runtime environment built on the V8 JavaScript engine. It allows developers to use JavaScript for both client-side and server-side scripting, unifying web application development. Node.js has since gained widespread adoption due to its efficiency and scalability.

Key features include its non-blocking, event-driven architecture, enabling asynchronous handling of requests. This makes Node.js well-suited for building real-time applications, such as chat applications and online gaming platforms. Node.js also boasts a vast ecosystem of open-source packages available through npm (Node Package Manager).

With a focus on speed and performance, Node.js is known for its ability to handle a large number of simultaneous connections. It's commonly used for building scalable network applications and APIs. As a cross-platform runtime, Node.js supports various operating systems, making it versatile for developers across different environments.

Node.js is extensively used by major companies, including Netflix, LinkedIn, and PayPal, showcasing its reliability for handling high traffic and demanding applications. Its flexibility, speed, and robust community support make Node.js a powerful choice for modern web development.

Figure : Logo of Node.js

2.1.2 Express JS

Express.js is a popular web application framework for Node.js. It is a minimal and flexible framework that provides a set of robust features and utilities for building web applications and APIs.

Express.js simplifies the process of handling HTTP requests, routing, middleware management, and rendering dynamic content. It follows the middleware pattern, allowing developers to add modular functions (middleware) to process requests and responses. This makes it easier to implement various functionalities like authentication, error handling, and logging.

 With Express.js, you can create server-side applications, build RESTful APIs, handle routing, serve static files, and more. It provides a straightforward and intuitive API for handling HTTP methods (GET, POST, PUT, DELETE, etc.), URL routing, request parameters, and response handling

Figure : Logo of ExpressJS

1. The librares

2.2.1 bcrypt (5.1.1):

Purpose: A library for hashing passwords securely.

Usage: Commonly used in authentication systems to securely store and verify user passwords.

2.2.2 body-parser (1.20.2):

Purpose: Middleware for parsing incoming request bodies in Express applications.

Usage: Extracts data from form submissions or API requests, making it accessible in the request object.

2.2.3 cookie-parser (1.4.6):

Purpose: Parses cookies attached to the client's request object.

Usage: Extracts and processes cookies, commonly used for session management and user authentication.

2.2.4 dotenv (16.3.1):

Purpose: Loads environment variables from a .env file into Node.js applications.

Usage: Safely stores sensitive information, such as API keys or database credentials, outside of code repositories.

2.2.5 express (4.18.2):

Purpose: A fast, unopinionated, minimalist web framework for Node.js.

Usage: Facilitates the development of web applications and APIs with a robust set of features and a vibrant ecosystem.

2.2.6 express-flash (0.0.2):

Purpose: Middleware for displaying flash messages in Express applications.

Usage: Provides a way to show temporary messages to users, typically after a form submission or other user actions.

2.2.7 express-handlebars (7.1.2):

Purpose: A view engine for Express that integrates the Handlebars templating engine.

Usage: Simplifies rendering dynamic content in web pages, making it easy to inject data into HTML templates.

2.2.8 express-session (1.17.3):

Purpose: Middleware for handling sessions in Express applications.

Usage: Enables the storage and retrieval of user session data, often used for user authentication and tracking.

2.2.9 express-validator (7.0.1):

Purpose: A set of Express.js middlewares for data validation.

Usage: Validates and sanitizes incoming data from requests, ensuring it meets specified criteria.

2.2.10 handlebars (4.7.8):

Purpose: A popular templating engine for creating dynamic HTML.

Usage: Allows the creation of reusable templates with dynamic content, often used in conjunction with Express.js.

2.2.11 jsonwebtoken (9.0.2):

Purpose: A library for generating and verifying JSON Web Tokens (JWT).

Usage: Commonly used for user authentication and session management in web applications.

2.2.12 moment (2.29.4):

Purpose: A JavaScript library for parsing, validating, manipulating, and formatting dates.

Usage: Simplifies working with dates and times in JavaScript applications.

2.2.13 moment-timezone (0.5.43):

Purpose: An extension for Moment.js that provides support for time zone handling.

Usage: Allows developers to work with dates and times in specific time zones, useful for global applications.

2.2.14 morgan (1.10.0):

Purpose: HTTP request logger middleware for Node.js.

Usage: Logs information about incoming requests, helping with debugging and monitoring.

2.2.15 multer (1.4.5-lts.1):

Purpose: Middleware for handling multipart/form-data, primarily used for file uploads.

Usage: Enables the server to handle file uploads from client requests.

2.2.16 mysql2 (3.6.3):

Purpose: A MySQL database driver for Node.js.

Usage: Facilitates interaction with MySQL databases, allowing queries and data manipulation in Node.js applications.

2.2.17 nodemailer (6.9.7):

Purpose: A module for sending emails with Node.js.

Usage: Enables the sending of email notifications from Node.js applications.

2.2.18 nodemon (3.0.1):

Purpose: A utility that monitors for changes in Node.js applications and automatically restarts the server.

Usage: Facilitates a smoother development workflow by eliminating the need to manually restart the server after code changes.

2.2.19 sequelize (6.34.0):

Purpose: A promise-based ORM (Object-Relational Mapping) for Node.js, supporting various SQL databases.

Usage: Simplifies database interactions by providing an abstraction layer over SQL databases, allowing developers to work with models and queries in a JavaScript-centric manner.

1. Service

**AWS**

Deployment of our project is conducted on a VPS (virtual private server) instance provided by AWS. With the instance configuration’s as:

* OS: Ubuntu
* CPU architecture: 64-bit (x86)
* Storage: 8 Gib
* Ram: 1Gib

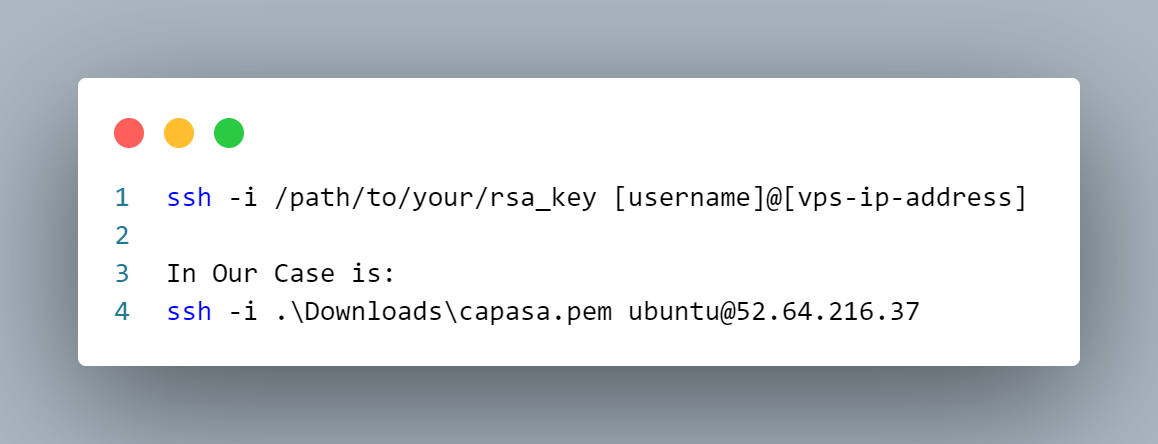
The Instance is connected via SSH with RSA authentication key pair. First the instance doesn’t allow access due to the default security rules. To access the VPS via SSH, we add security rules to the security groups in the console. After that we SSH to the VPS with the following commands:

Figure : Add Security Rule for SSH Connection

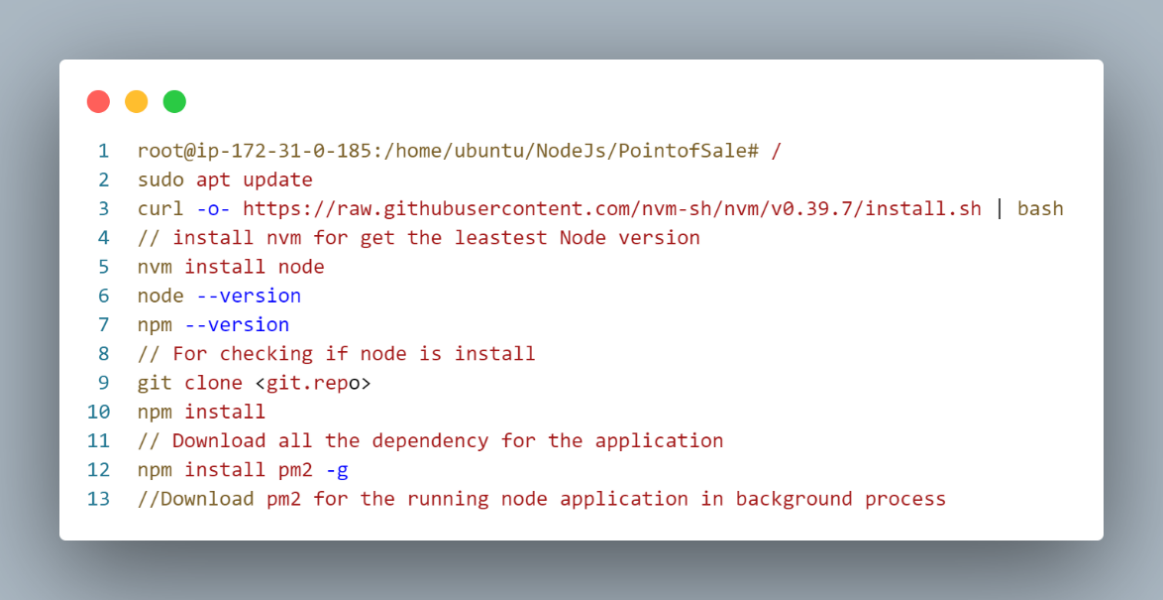
Once connected you will be introduced to the VPS terminal and we start installing the essential Node environment, MySQL Database, … for the Node application to run. The MySQL instance we need to create and grant a new user for connecting to the database.

Figure : Installation for node application

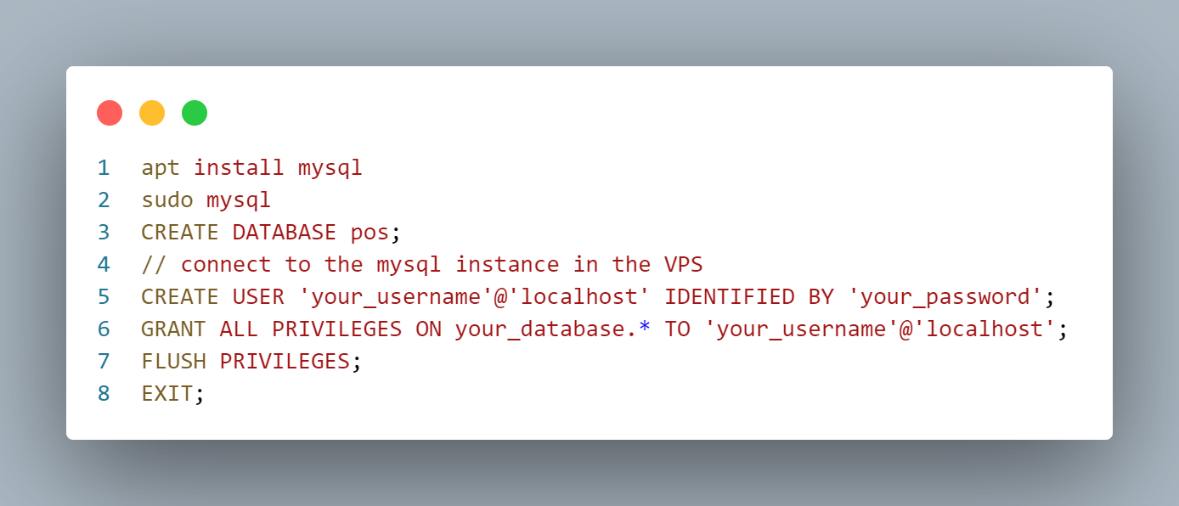


Figure : MySQL configuration

After that we access the configuration .env file for adjusting the environment variables to connect to this MySQL instance. We can use text editor like vim or nano which is self-integrated in the ubuntu.

Figure : Environment Variables for Node Application

Since we do want to change the code too much the port of the application still 3000. Instead, we install nginx which is a play as a role of reverse-proxy server for forwarding the http request from port 80 to this node application running at port 3000

A screenshot of a computer

Description automatically generatedWith the node configuration file as store in the /etc/nginx/sites-enabled directory

Figure : Configure nginx for forwarding to node application

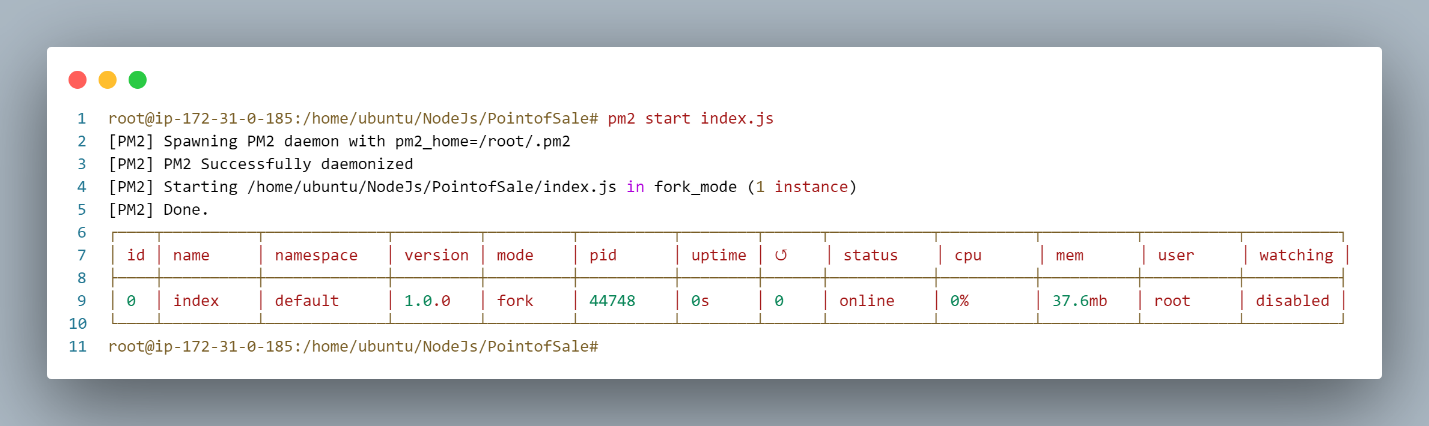
Lastly, we start up node application using pm2 as pm2 will start the node app on as the background process. Therefore, as long as this VPS instance is running the node won’t shut down.

Figure : Using pm2 to start up application

1. General User Interface

In this Chapter, we will demonstrate some main interface in our project

|  |  |
| --- | --- |
| User Interface | Description |
| Figure : Login Interface | Login Interface:  The Users can type their username and password to log into the system |
| Figure : DashBoard Interface | DashBoard Interface:  The Users can see Reporting and Analytics, total amount received, number of orders, number of products,… |
| Figure : View List of Products Interface | View List of Products Interface:  The user can view, edit, or delete a certain product  Besides there has a button for navigation to add a new product |
| Figure : View List of Users Interface | View List of Users Interface:  Ony Admin(Manager) can see this interface used for can view, edit, lock, or resend an email for a certain user |
| Figure : View List of Customers Interface | View List of Customers Interface:  The user can view list of customers, detail information of a certain user |
| Figure : Sales Interface | Sales Interface:  The user can add a product by searching or barcode. In this interface, all actions will be done by Ajax technique, bringing more easy for user |

Table : User Interface Description