ENROLLMENT MANAGEMENT USING PostgreSQL

A NAAN MUDHALVAN PROJECT REPORT

SUBMITTED BY

ULAGAPPAN.P 912421106018

VEERAMANI PL 912421106019

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SHANMUGANATHAN ENGINEERING COLLEGE

ARASAMPATTI, PUDUKKOTTAI – 622 507

YEAR & SEMESTER : IV & VII

SUBJECT CODE : NM1050

COURSE NAME : SaaS

ANNA UNIVERSITY::CHENNAI 600 025

NOV/DEC 2024

ANNA UNIVERSITY::CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this Naan Mudhalvan project report **"ENROLLMENT MANAGEMENT USING PostgreSQL"** is the bonafide work of **"UALAGAPPAN P (912421106018), VEERAMANI PL (912421106019) "** who carried out the project work under my guidance.

SIGNATURE

Mrs. D. LATHA, M.E.,

NAAN MUDHALAVAN COORDINATOR

Assistant Professor,

Dept. of Electronics and Comm. Engg,

Shanmuganathan Engineering College,

Arasampatti-622 507

SIGNATURE

Dr. A.MUTHU MANICKAM , M.E., Ph.D.,

HEAD OF THE DEPARTMENT

ASSISTANT PROFESSOR,

Department of Electronics & Communication Engineering,

Shanmuganathan Engineering College,

Arasampatti – 622 507

Submitted for the Internship viva-voice on _____

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGMENT

At this pleasing moment having successfully completed our internship report, we wish to convey our sincere thanks to our beloved chairperson **Mrs. PICHAPPA VALLIAMMAL,** correspondent **Dr. P. MANIKANDAN B.E**, director (Academic) Shri **M.SHANMUGANATHAN**, director(Administration) Shri **M. PICHAPPA** and honourable secretary **Mr. M. VISWANATHAN** for their extensive support.

I thankful to our principal **Dr. KL. MUTHURAMU M.E(W.R)., M.E(S.E)., Ph.D., FIE., M.I.S.T.E.,** Shanmuganathan engineering college, for providing the opportunity to conduct our project.

I extend our gratitude to **Dr. A.MUTHU MANICKAM M.E., Ph.D.,** the head of the department and our internship co-ordinator of Electronics and communication engineering for providing a valuable suggestion and supports given through the study.

Gratitude never fails. We are grateful to our dynamic and effective internal guide **Asst. Prof. Mrs. D. LATHA, M.E, AP/ECE**. for his valuable innovative suggestion, constructive interactions, constant encouragement and valuable helps that have been provided us throughout the project.

I also express my heartfelt thanks to all other staff members of Electronics communication engineering department for their support. Above all, we thank our parents, for affording us the valuable education till now.

ABSTRACT

The "Enrollment Management System Using HTML, CSS, JavaScript, and PostgreSQL" is a web-based application designed to efficiently manage student enrollments within educational institutions. This project integrates a PostgreSQL database for robust and reliable storage of enrollment data, while the frontend is developed using HTML, CSS, and JavaScript to provide a user-friendly and responsive interface. The system allows administrators to handle key functions such as student admissions, course registrations, schedule management, and faculty assignments. With role-based access control, users such as administrators, faculty, and students can access specific features according to their permissions. The PostgreSQL database ensures secure data storage and enables efficient querying for real-time data access. JavaScript enhances the interactivity of the system, allowing for dynamic search, form validation, and data visualization on the frontend. Additionally, this project features real-time reporting, offering insights into enrollment trends and academic progress. By combining a powerful database with an intuitive interface, this system aims to improve the efficiency and accessibility of enrollment processes for educational institutions, providing a scalable and user-centric solution.

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 ENROLLMENT MANAGEMENT SYSTEM:

The "Enrollment Management System Using HTML, CSS, JavaScript, and PostgreSQL" is a comprehensive solution designed to streamline the enrollment processes within educational institutions. In an era where data management and accessibility are crucial, this project focuses on digitizing and centralizing the tasks associated with student enrollments, making it easier for institutions to handle admissions, course registrations, schedules, and academic records.

Educational institutions often face challenges in managing large volumes of student data, especially when handling sensitive information such as grades, schedules, and personal details. Manual processes or outdated systems can lead to data redundancy, errors, and inefficiencies. This project leverages PostgreSQL, a robust relational database management system, to store and manage enrollment data securely and efficiently. The use of HTML, CSS, and JavaScript on the frontend offers a modern and responsive interface, allowing users to access the system seamlessly across various devices.

The Enrollment Management System supports multiple user roles, including administrators, faculty, and students, each with specific permissions to view or edit data. For instance, administrators can manage admissions, update course offerings, and track overall enrollment statistics, while faculty members can access class rosters and student performance data. Students, on the other hand, can view their academic schedules, register for courses, and monitor their academic progress. By combining PostgreSQL with frontend technologies, this project ensures data security, real-time access, and efficient data retrieval. Features such as role-based access control, form validation, and interactive elements make the system user-friendly and reliable. With built-in reporting capabilities, the system provides valuable insights into enrollment trends, helping administrators make informed decisions.

CHAPTER 2
SYSTEM ARCHITECTURE

The architecture of the Enrollment Management System is designed to ensure a seamless flow of data between the frontend, backend, and the PostgreSQL database. It leverages a multi-tiered structure comprising the presentation layer (frontend), application layer (backend), and data layer (database). Each layer has a distinct role and communicates with others in a systematic manner to facilitate efficient enrollment management.

The frontend serves as the user interface, allowing different types of users (administrators, faculty, and students) to interact with the system. HTML provides the structure of the web pages, CSS styles them, and JavaScript ensures interactivity, allowing for real-time data validation, dynamic page updates, and interactive elements.

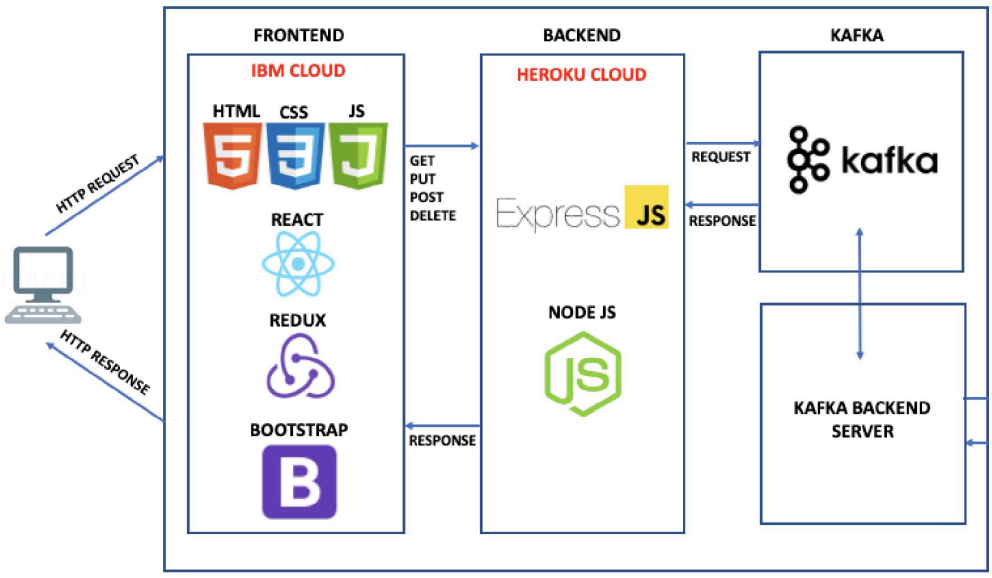

Fig 2.1 **System Architecture**

The backend is responsible for processing requests from the frontend, performing business logic, and handling interactions with the PostgreSQL database. It serves as an intermediary between the database and the user interface, managing queries, updates, and user permissions.

CHAPTER-3

FRONTEND LAYER

The frontend is the client-facing component of the architecture, responsible for interacting with users and displaying inventory data. Built using HTML, CSS, and JavaScript, this layer is designed for responsiveness, usability, and efficient data handling.

3.1 HTML & CSS

In the "Build a RESTful API for an Inventory Management System" project, HTML and CSS work in tandem to create a structured, user-friendly, and visually appealing interface that enhances the user experience while managing inventory data. HTML (Hyper Text Markup Language) forms the foundation of the user interface by defining the structure and layout of each web page. HTML elements like forms, tables, buttons, and input fields provide essential building blocks that allow users to interact with the system

Fig 3.1.1 **HTML**

CSS (Cascading Style Sheets), on the other hand, enhances the presentation and layout of these HTML elements, making the application aesthetically pleasing and intuitive to use. CSS provides styling and formatting rules that bring consistency and visual hierarchy to the interface. With CSS, elements like tables, buttons, and forms can be customized with colors, font styles, and spacing, ensuring they are visually distinct and easy to identify.

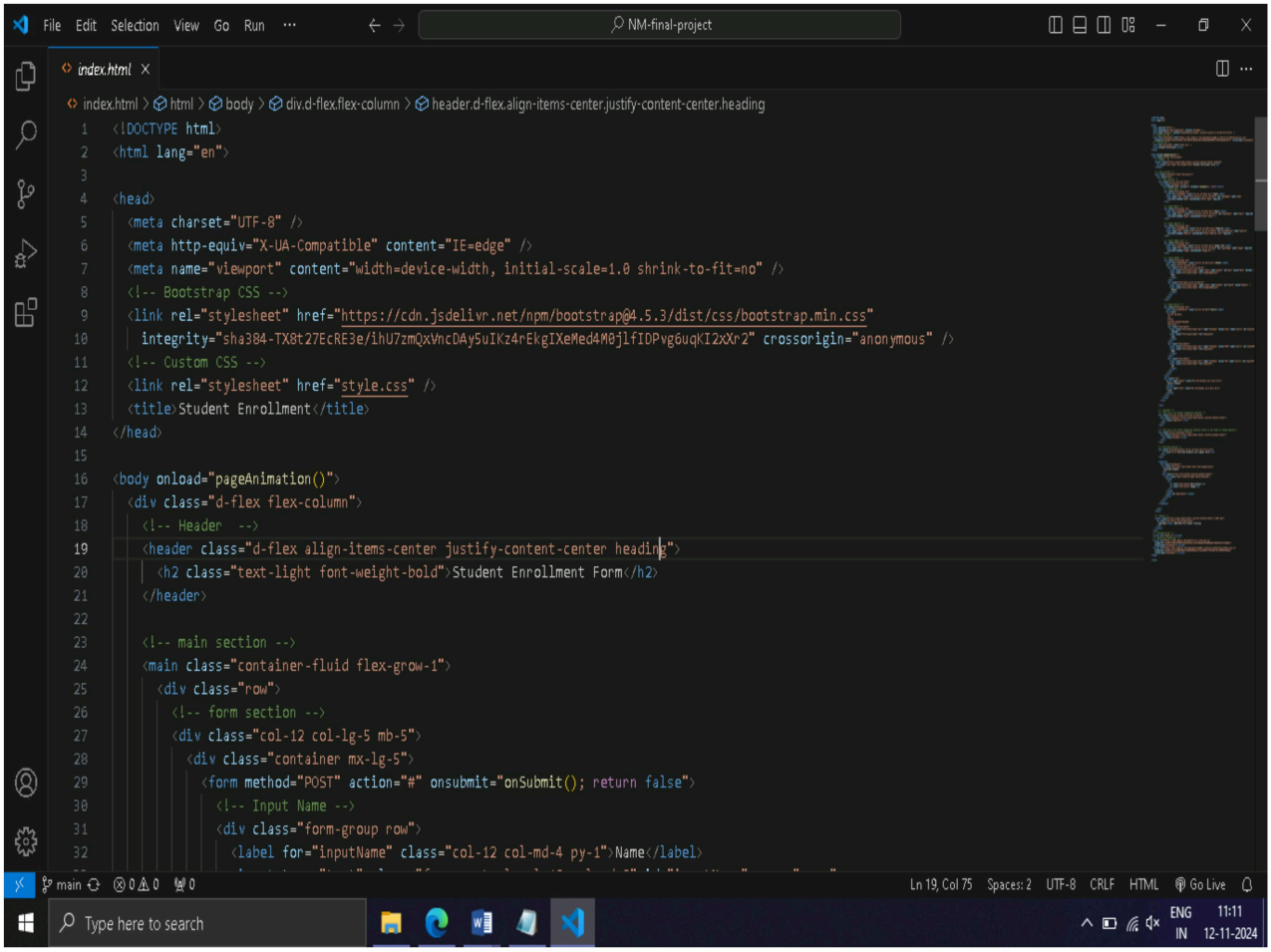

Fig 3.1.2 **HTML Code Execution.**

Fig 3.1.3 **CSS**

CSS enables responsive design, allowing the layout to adapt to different screen sizes, making the application accessible on desktops, tablets, and smartphones. Additionally, CSS frameworks like Bootstrap can be incorporated to quickly implement standardized styling, making the design process faster and ensuring a professional look. Together, HTML and CSS create a cohesive and interactive experience for users.

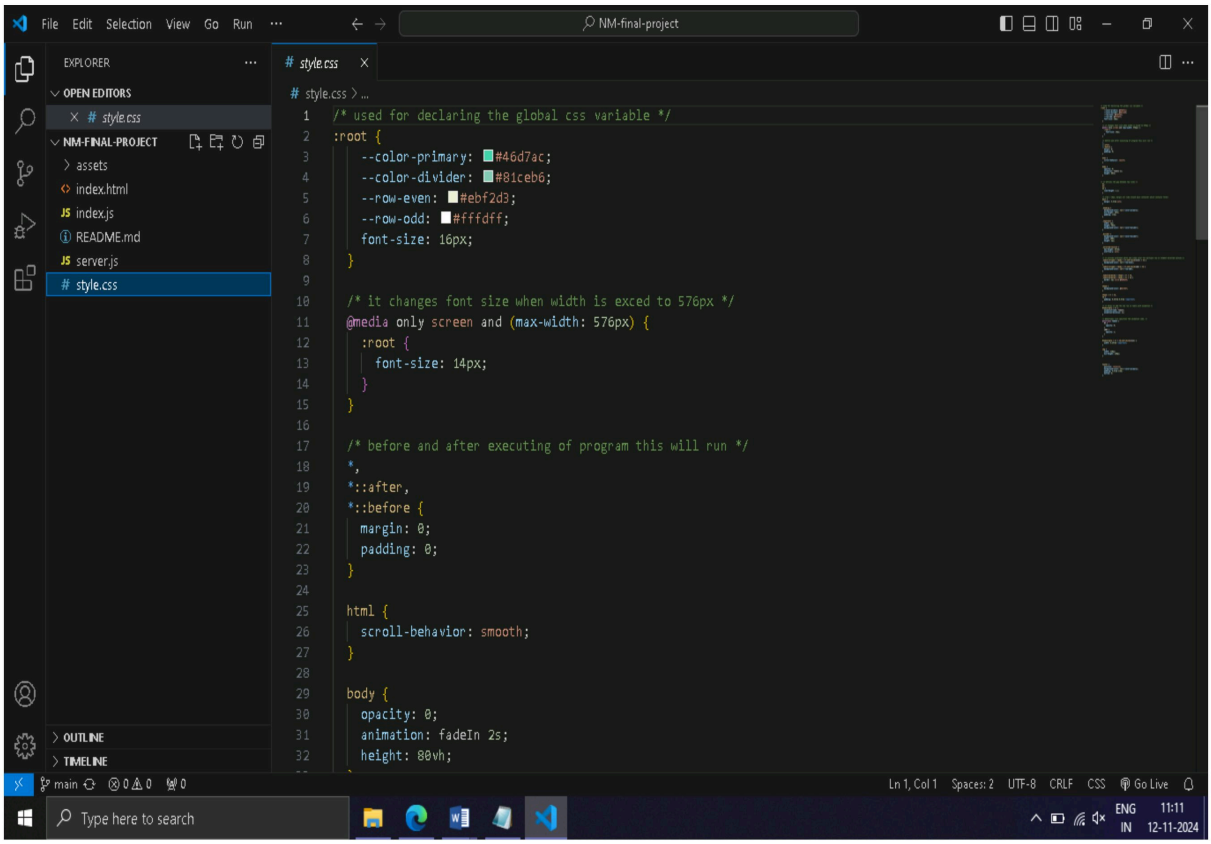

Fig 3.1.4 **CSS Code Execution**

3.2 JAVASCRIPT

Positioned as the primary scripting language on the frontend, JavaScript is responsible for managing the communication between the user interface (built with HTML and styled by CSS) and the backend RESTful API, allowing users to interact with the system seamlessly. JavaScript's primary function in this project is to handle AJAX (Asynchronous JavaScript and XML) requests, enabling asynchronous communication with the backend API.

This means that when a user performs an action, such as adding a new item, updating inventory details, or deleting an item, JavaScript can send an HTTP request to the backend without requiring a full page reload, keeping the interface responsive and user-friendly. This asynchronous communication is vital for providing a real-time, interactive experience. In addition to handling data fetching and display, JavaScript enables interactive elements that enhance usability. It powers actions like hover effects, button clicks, and form submissions, making the interface more responsive and intuitive. JavaScript frameworks and libraries like jQuery can be employed to simplify these tasks, particularly when dealing with complex animations or manipulating multiple elements at once.

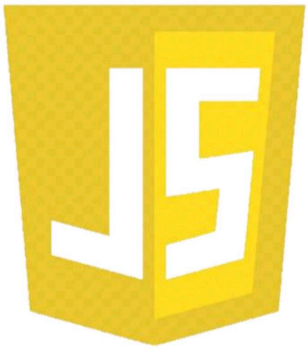

Fig 3.2.1 **JavaScript**

JavaScript's flexibility also supports error handling in API calls, allowing the application to catch errors and provide feedback to the user if something goes wrong, such as network issues or invalid requests. This is particularly useful for displaying meaningful messages, like "Item added successfully" or "Error: Could not delete item." Additionally, JavaScript's interaction with CSS allows for dynamic styling adjustments, like highlighting table rows when selected or changing button colors based on user actions, making the interface feel responsive and alive. By integrating with HTML and CSS, JavaScript transforms a static web page into a fully interactive, responsive, and user-centric application, enhancing the efficiency and usability of the inventory management system.

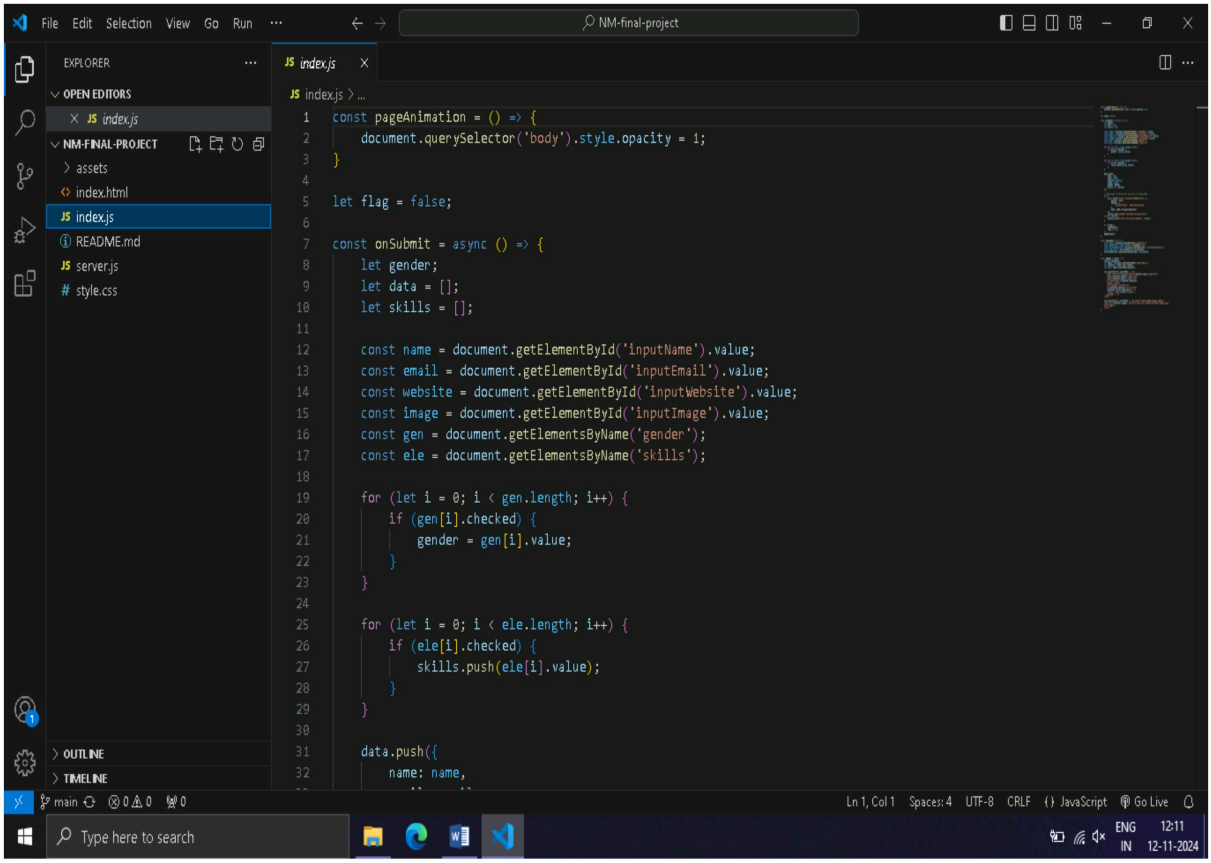

Fig 3.2.2 **JS Code Execution**

CHAPTER 4
DATABASE AND BACKEND

4.1., PostgreSQL:

PostgreSQL is an advanced open-source relational database management system (RDBMS) that provides powerful features for handling complex data relationships, high performance, and data integrity. It is well-suited for applications that require reliable data storage, robust querying capabilities, and scalability, making it an ideal choice for an Enrollment Management System. Here's a detailed overview of how PostgreSQL can be used for the database layer in this system.

ACID Compliance PostgreSQL is fully ACID-compliant, ensuring that all transactions are handled safely and reliably. This is crucial for an enrollment system where data integrity is paramount, especially when managing student records, enrollments, and course capacities. Relational and Object-Relational Capabilities PostgreSQL is a relational database with support for object-relational features, allowing it to store complex data types and manage relationships between tables, which is essential for representing student-course-faculty relationships in an educational setting.

Fig 4.1.1 **PostgreSQL**

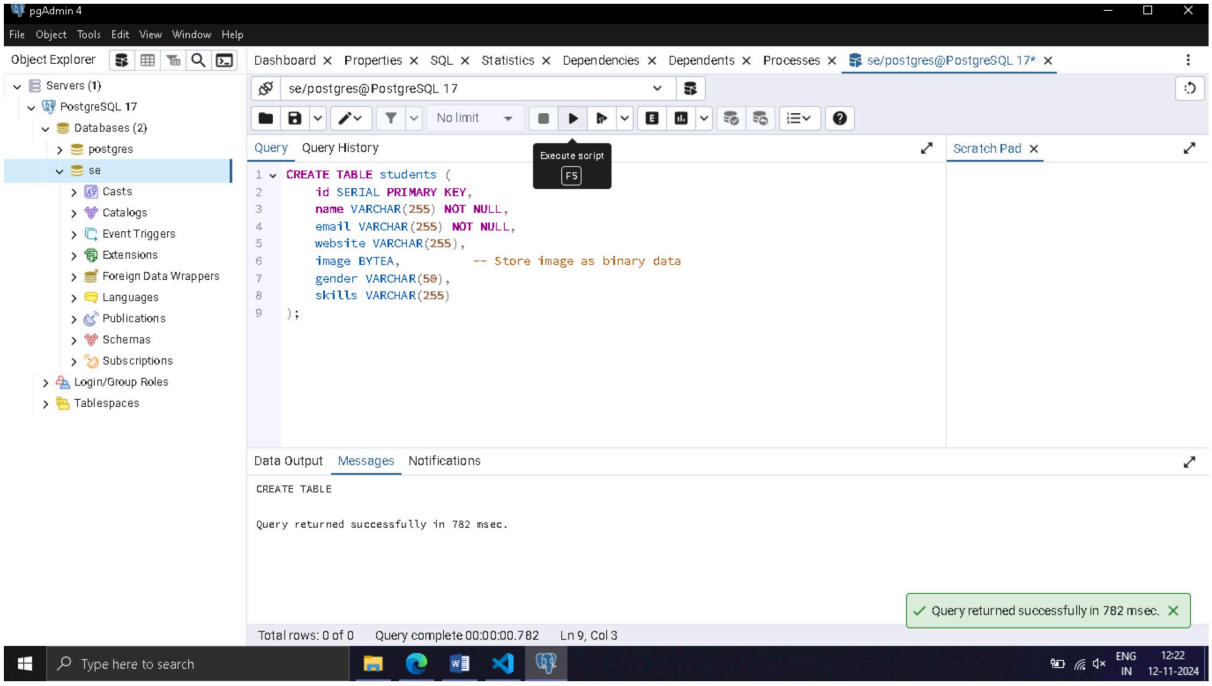

Fig 4.1.2 **JS Code Execution**

4.2 NODE.JS

Node.js plays a critical role in the API layer by serving as the backend runtime environment that powers the server-side logic. Node.js, an open-source, JavaScript-based runtime, is built on Chrome's V8 JavaScript engine and is designed for high-performance, non-blocking I/O operations, making it ideal for handling the numerous simultaneous client requests an inventory management system may receive. This efficiency is particularly valuable for our project, as Node.js enables the API to manage real-time data transactions, ensuring that users can add, retrieve, update, or delete inventory items with minimal delay. Its event-driven architecture allows the API layer to handle multiple requests concurrently, which is essential for scalability and responsiveness, as the system needs to handle growing volumes of inventory data and user requests without sacrificing performance.

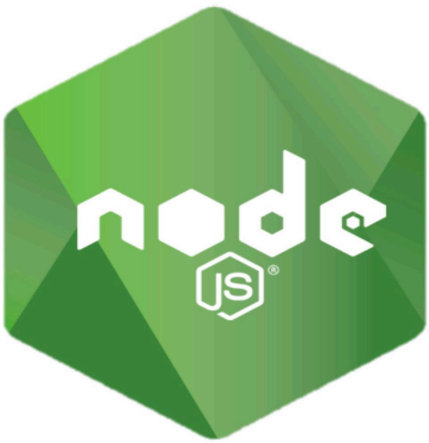

Fig 4.2.1 **Node.js**

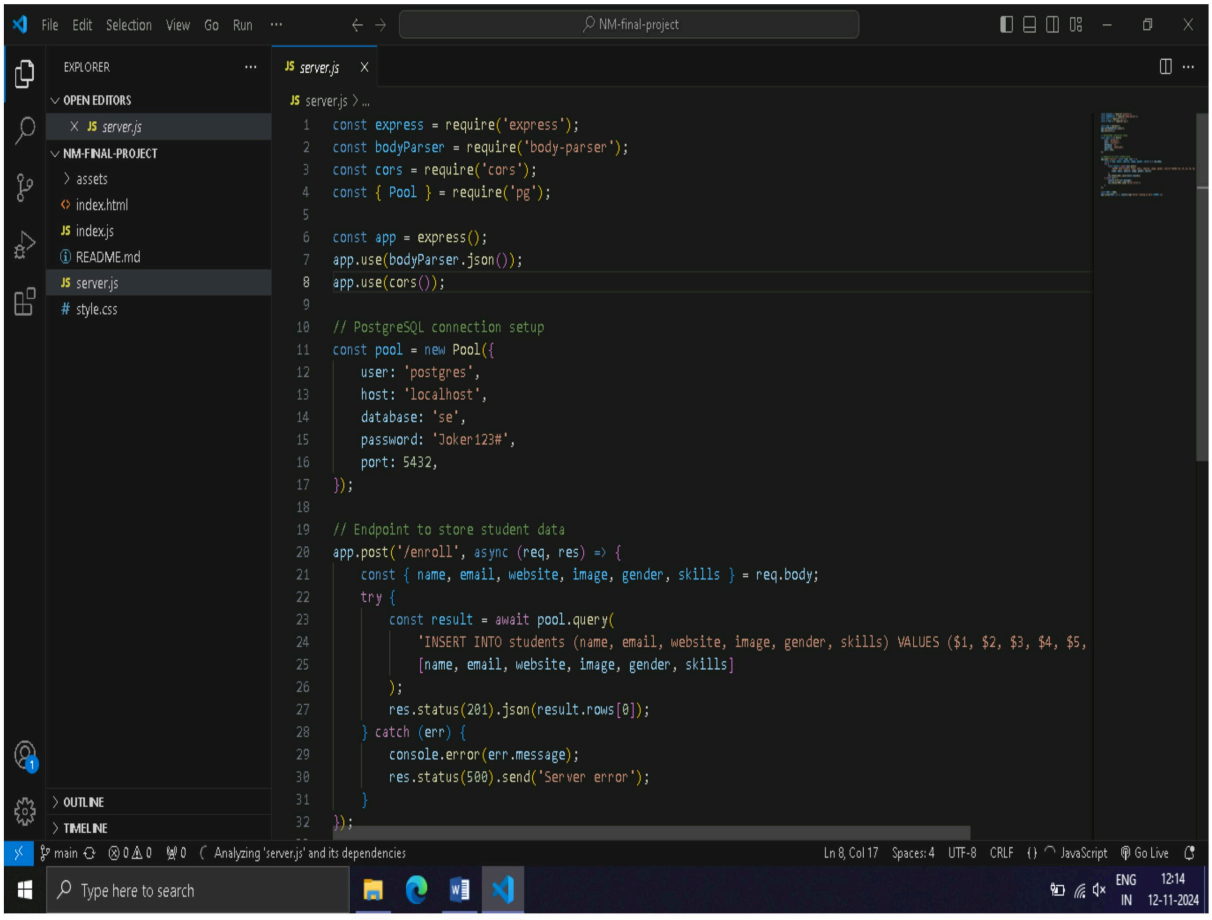

Fig 4.2.2 **node JS Code Execution**

Node.js is an open-source, cross-platform JavaScript runtime environment that allows developers to build server-side applications using JavaScript. Traditionally, JavaScript was confined to the browser, but Node.js enables developers to use JavaScript on the server, making it possible to build the entire stack (both frontend and backend) with a single programming language. Node.js is popular for its performance, scalability, and its vast ecosystem, making it an excellent choice for building modern web applications, including REST APIs, real-time applications, and microservices.

1. **Environment Configuration**:

   Sensitive data, such as database credentials and API keys, should be managed using environment variables, often stored in .env files and loaded using packages like dotenv.

2. **Data Validation and Sanitization**:

   Libraries like Joi and validator help validate and sanitize input data, ensuring security against attacks like SQL injection and XSS (Cross-Site Scripting).

3. **Authentication and Authorization**:

   Node.js supports authentication using methods like JSON Web Tokens (JWT), OAuth, and sessions, ensuring only authorized users can access protected routes.

CHAPTER 5

INTERFACE OF THE PROJECT

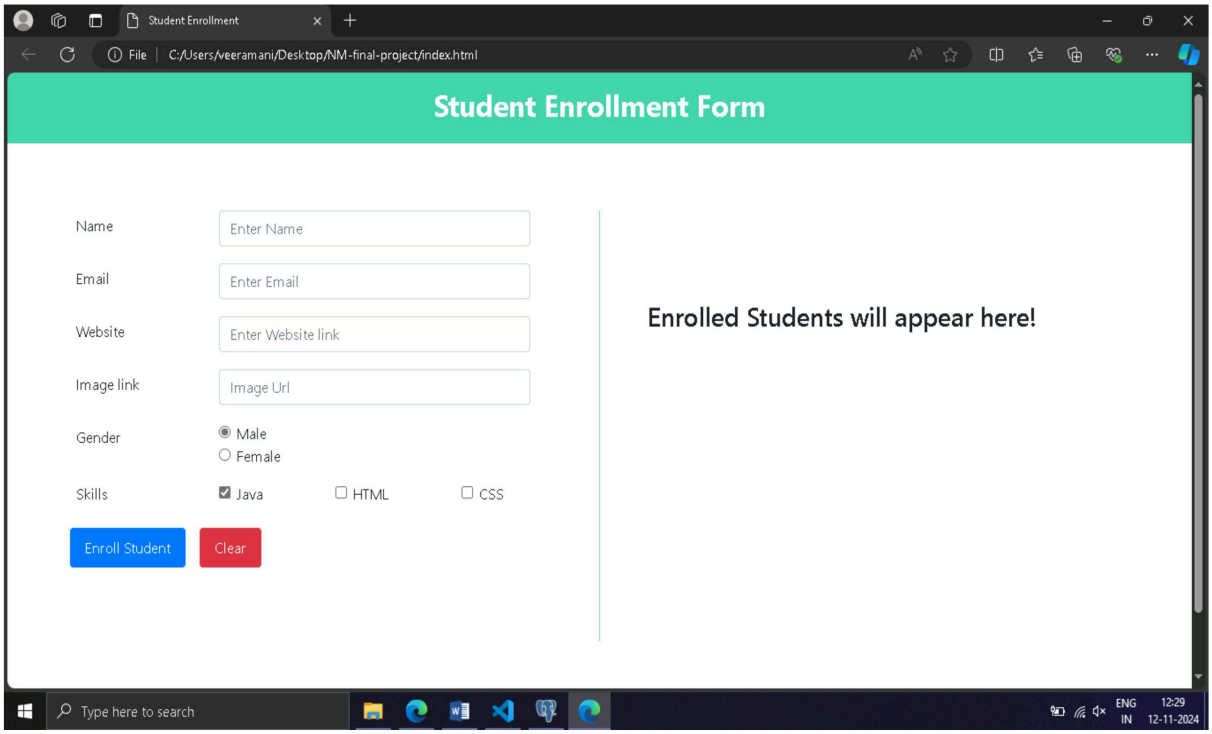

Fig 5.1 **STARTING INTERFACE**

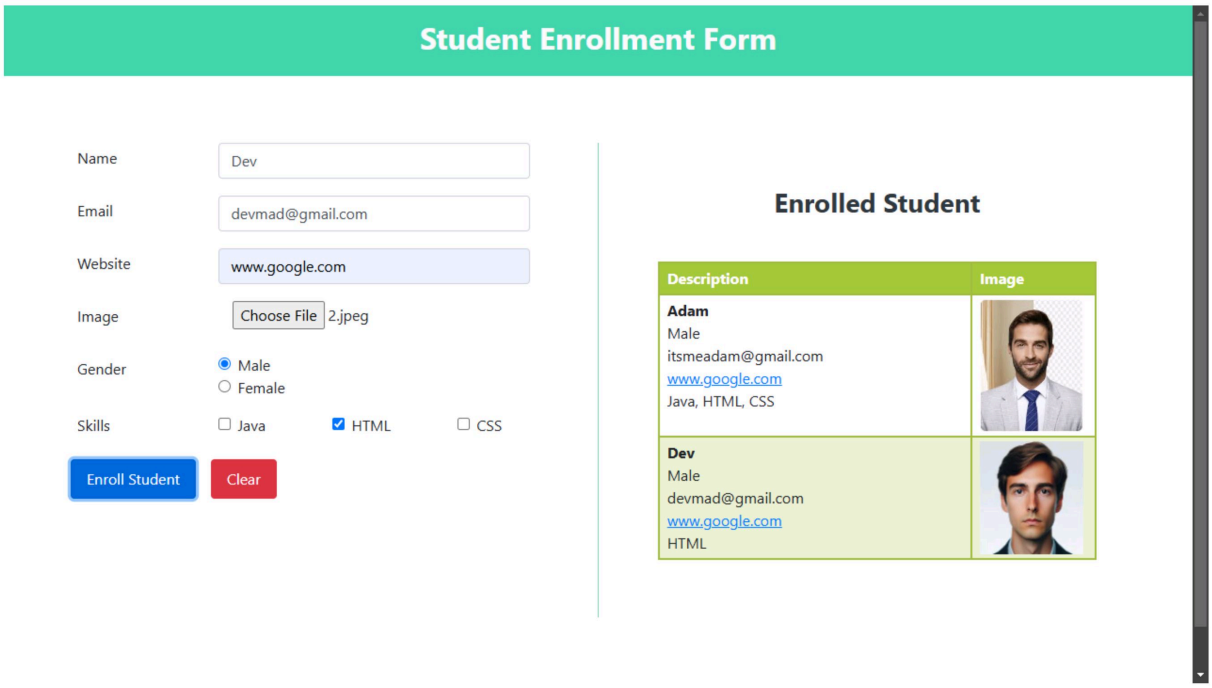

Fig 5.1 **INPUT LISTED INTERFACE**

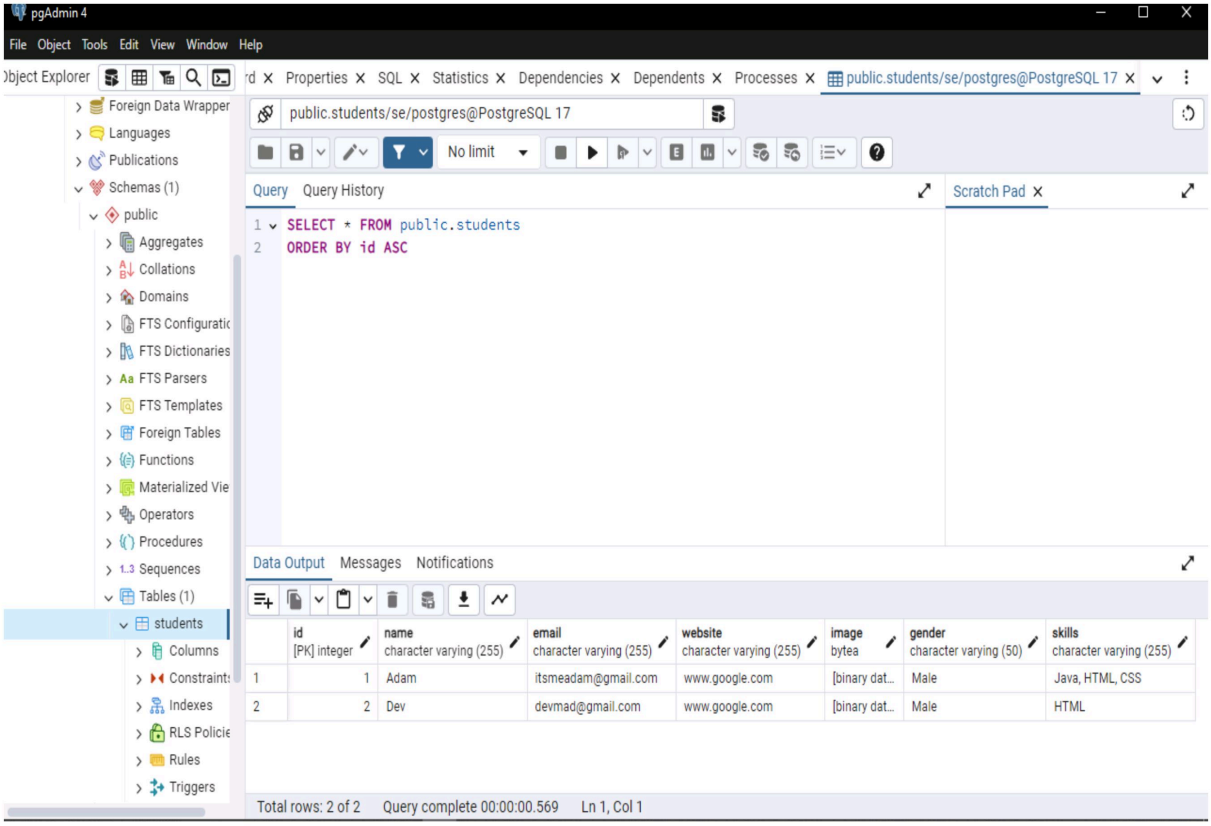

Fig 5.3 **PostgreSQL OUTPUT STORED INTERFACE**

CHAPTER 6

CONCLUSION

In conclusion, Node.js offers a powerful, efficient, and versatile environment for building modern web applications and backend services. Its non-blocking, event-driven architecture, powered by the V8 JavaScript engine, makes it highly suitable for I/O-intensive applications requiring concurrent connections, such as RESTful APIs and real-time applications. Node.js supports seamless integration with SQL and NoSQL databases, allowing it to handle various data management needs. With a vast ecosystem of libraries and tools through NPM, developers have access to pre-built solutions for everything from data validation to real-time communication, accelerating development.

Security and scalability are well-supported in Node.js, with options for environment configuration, role-based access control, data encryption, and clustering to optimize server utilization. The built-in module system promotes modular coding, while popular frameworks like Express simplify web server and API development. Moreover, Node.js has robust testing, debugging, and deployment support, making it reliable for both small and large-scale applications. Its compatibility with microservices, serverless architectures, and cloud platforms like AWS and Azure further enhances its flexibility and scalability. Overall, Node.js enables developers to build highly efficient, scalable, and maintainable applications that meet diverse performance and operational demands.

GUVI

Google for Education Partner

ISO 9001-27001 CERTIFIED

Ulagappan Pandiyan

is here by awarded the certificate of achievement
for the successful completion of

First SaaS Application

M. Arunprakash

Founder and CEO,
GUVI Geek Networks.

Certificate issued on : October 24 2024

Verified certificate ID: 970458OL427jhT11Nt

Verify certificate at: www.guvi.in/certificate?id=970458OL427jhT11Nt

GUVI

Google for Education Partner

ISO 9001-27001 CERTIFIED

Veeramani Palaniappan

is here by awarded the certificate of achievement
for the successful completion of

First SaaS Application

M. Arunprakash

Founder and CEO,
GUVI Geek Networks.

Certificate issued on : October 24 2024

Verified certificate ID: 76xi612Gg93Mf12761

Verify certificate at: www.guvi.in/certificate?id=76xi612Gg93Mf12761