**E-commerce Application on IBM cloud foundry**

|  |  |
| --- | --- |
| **Date** | **26-10-2023** |
| **Team ID** | **4218** |
| **Project Name** | **E-Commerce Application on IBM Cloud Foundry** |

**1. Introduction**

The objective is to create an artisanal E-commerce platform by using IBM Cloud Foundry. The primary goal is to combine the talented artisans by highlighting their handmade goods and offering functionalities such as secure shopping carts, payment gateways, and a user-friendly checkout process. This entails crafting the E-commerce platform, integrating essential features, and guaranteeing a smooth user journey. We have used Node.js to connect a API with the backend and to store the data in the database we have used Mysql.

**2. Problem Statement**

Build an artisanal E-commerce platform using IBM Cloud Foundry. Showcase handmade products, from exquisite jewellery to artistic home decor. Implement secure shopping carts, smooth payment gateways, and an intuitive checkout process. Nurture creativity and support small businesses through an artisan's dream marketplace with high security and performance.

**Backend Development**

**Step 1: Set Up Your Development Environment**

To set up the development environment for an E-commerce app in Node.js (JavaScript), install the necessary dependencies

Additionally, you will need a MySQL database for data storage. You can set up a local MySQL database or use a cloud-based solution like Amazon RDS or Google Cloud SQL.

**Step 2: Design Your Database**

Identify the entities you need to store, such as users, products, orders, and payments. Create tables in your MySQL database to represent these entities and define their relationships.

**Step 3: Create the Node.js Application**

In your project directory, create a javascript application. Define your app and set up the necessary configurations.

**Step 4: Set Up Database Connection**

Configure your Flask app to connect to the MySQL database. You can use a library like SQL Alchemy or Flask-MySQL to simplify database interactions. Set the database credentials in your configuration:

**Step 5: Define Models**

Create models to represent your database tables. These models will map to your database schema. For example, you might have models for User, Product, Order, and Payment. Define the fields and relationships for each model.

**Step 6: Create API Endpoints**

Define routes and endpoints for your E-commerce application. For example, you might create routes to retrieve product information, process orders, and handle user authentication.

**Step 7: Implement User Authentication**

Secure your application by implementing user authentication by using node.js for user sessions and token-based authentication.

**Step 8: Integrate Payment Gateway**

It involves selecting a secure provider, offering various payment methods, streamlining checkout, ensuring compliance, and monitoring transactions for optimal customer experience.

**Step 9: Test Your Application**

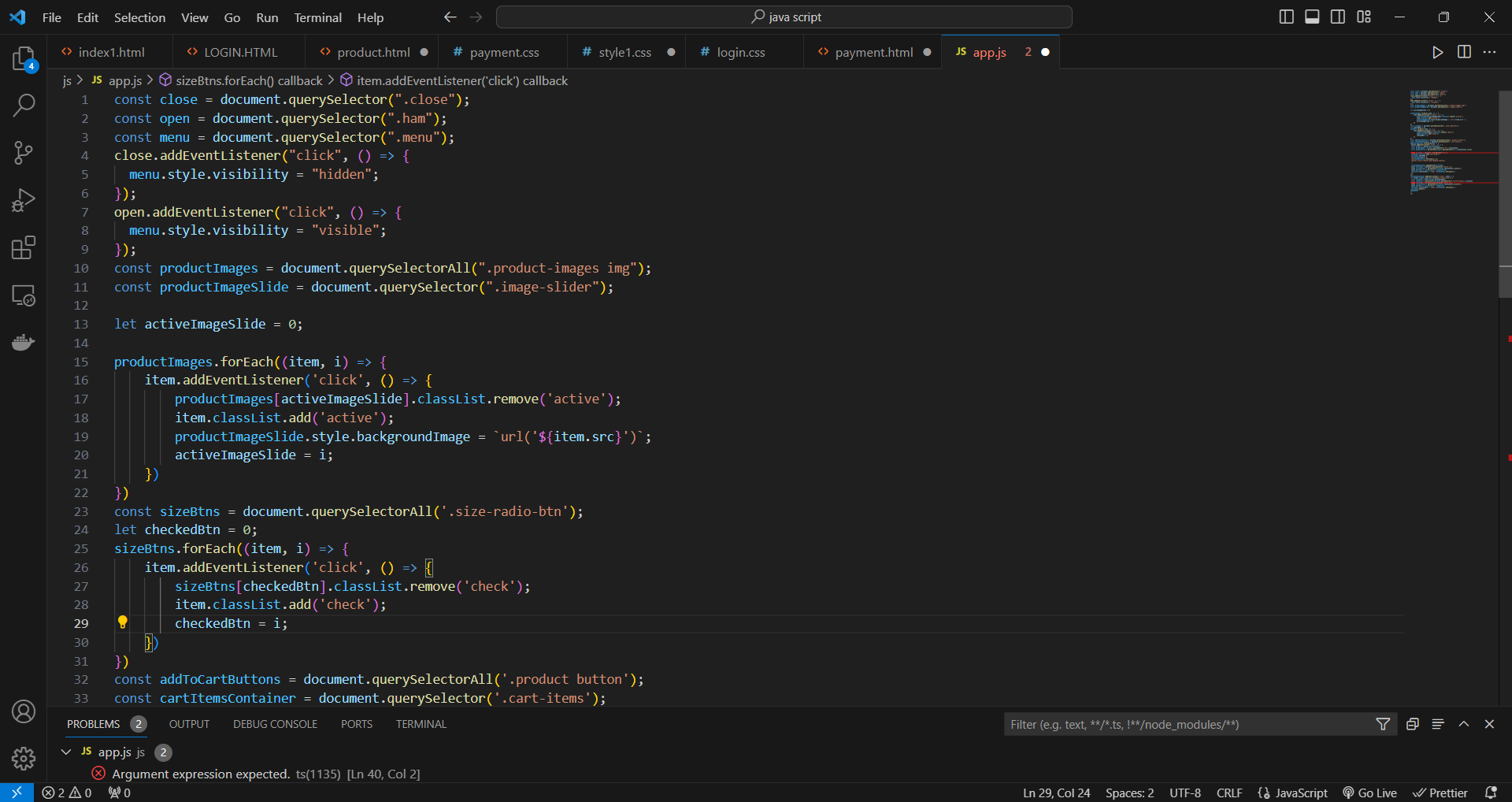
Thoroughly test your E-commerce backend by simulating user interactions, including product searches, adding items to the cart, and completing orders. Make sure all components work seamlessly.

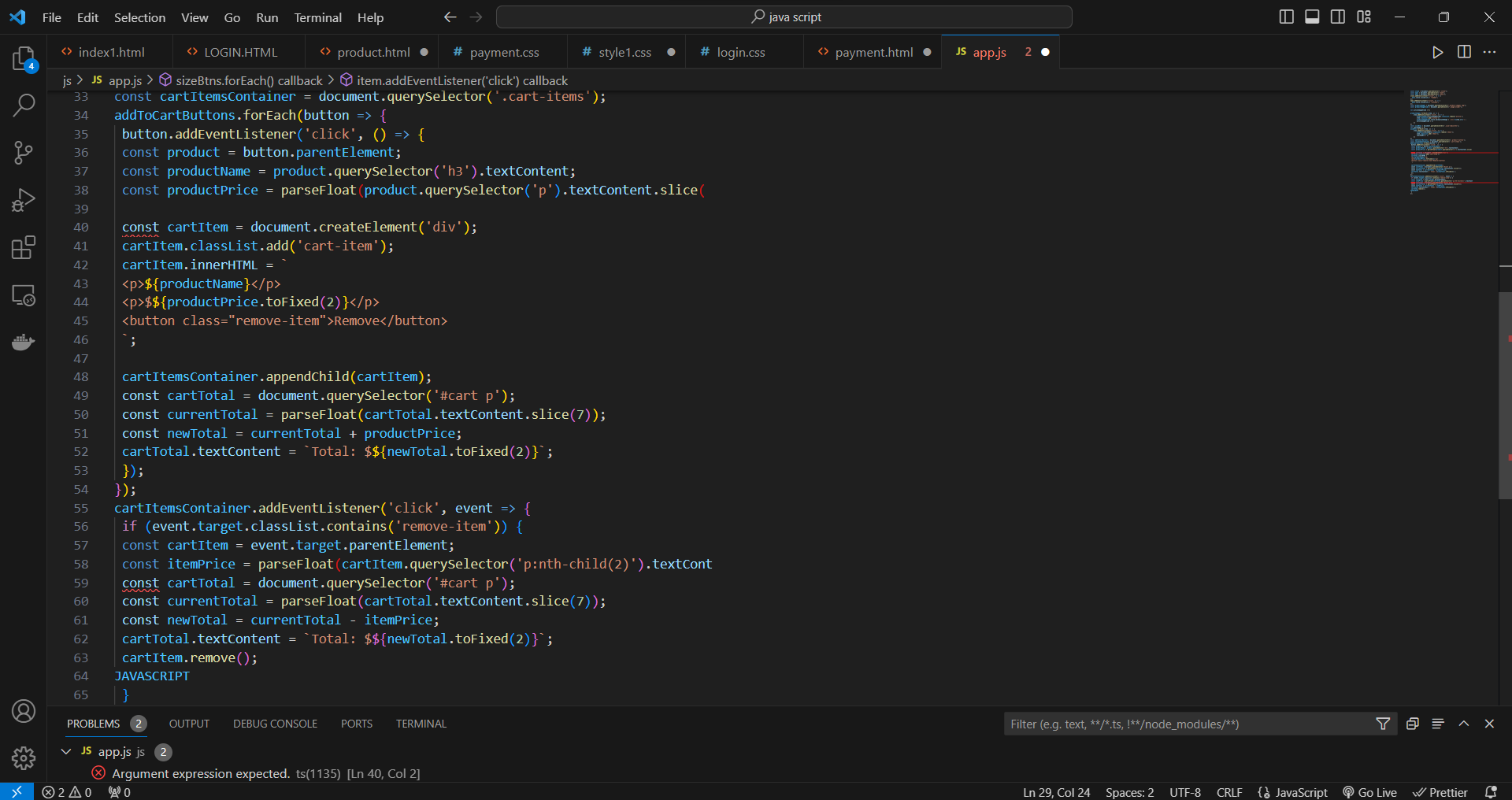
**Step 10: Deploy Your Backend**

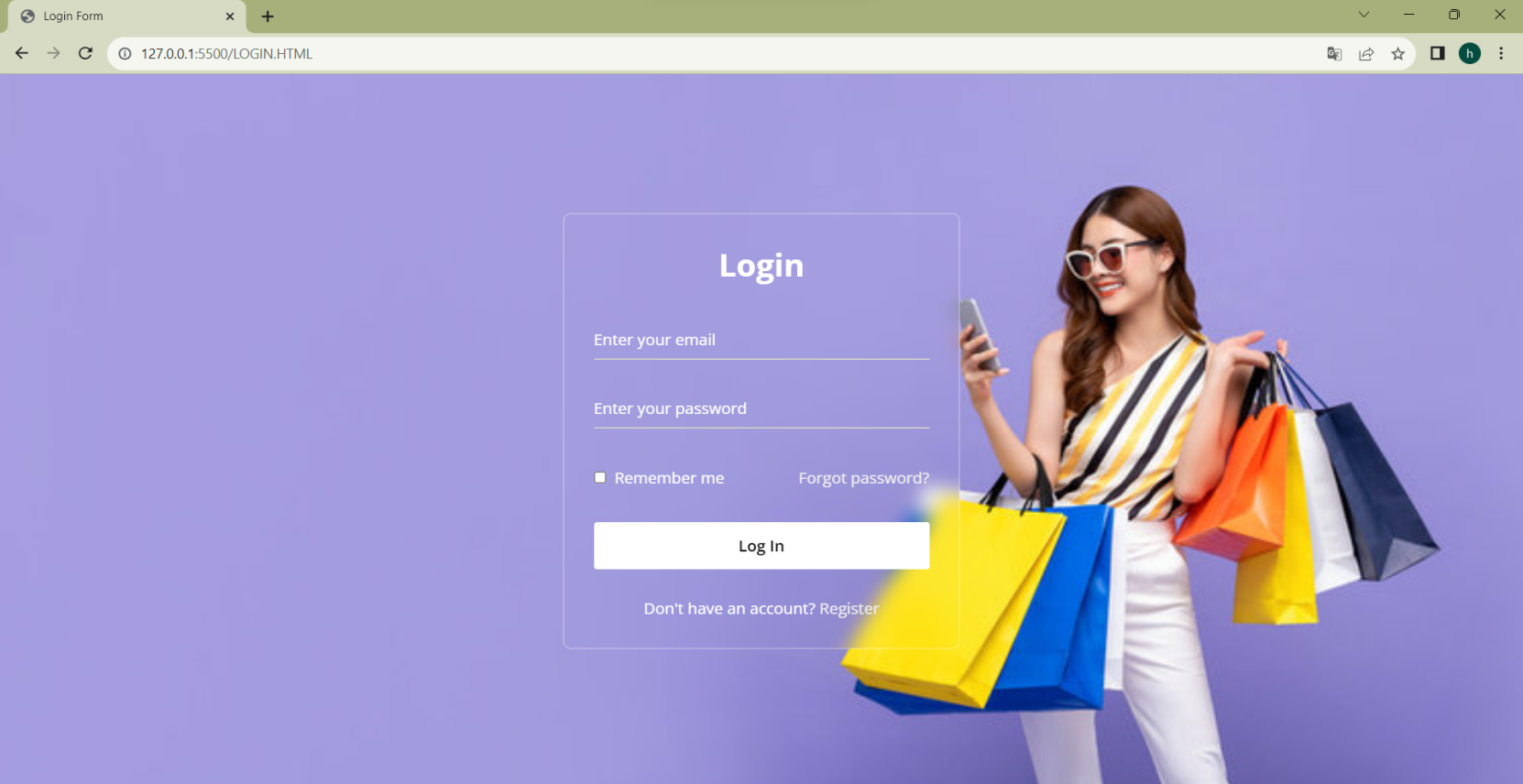
Once your backend is fully tested and ready, deploy it to a production server. Popular hosting options include AWS, or a dedicated web hosting service.

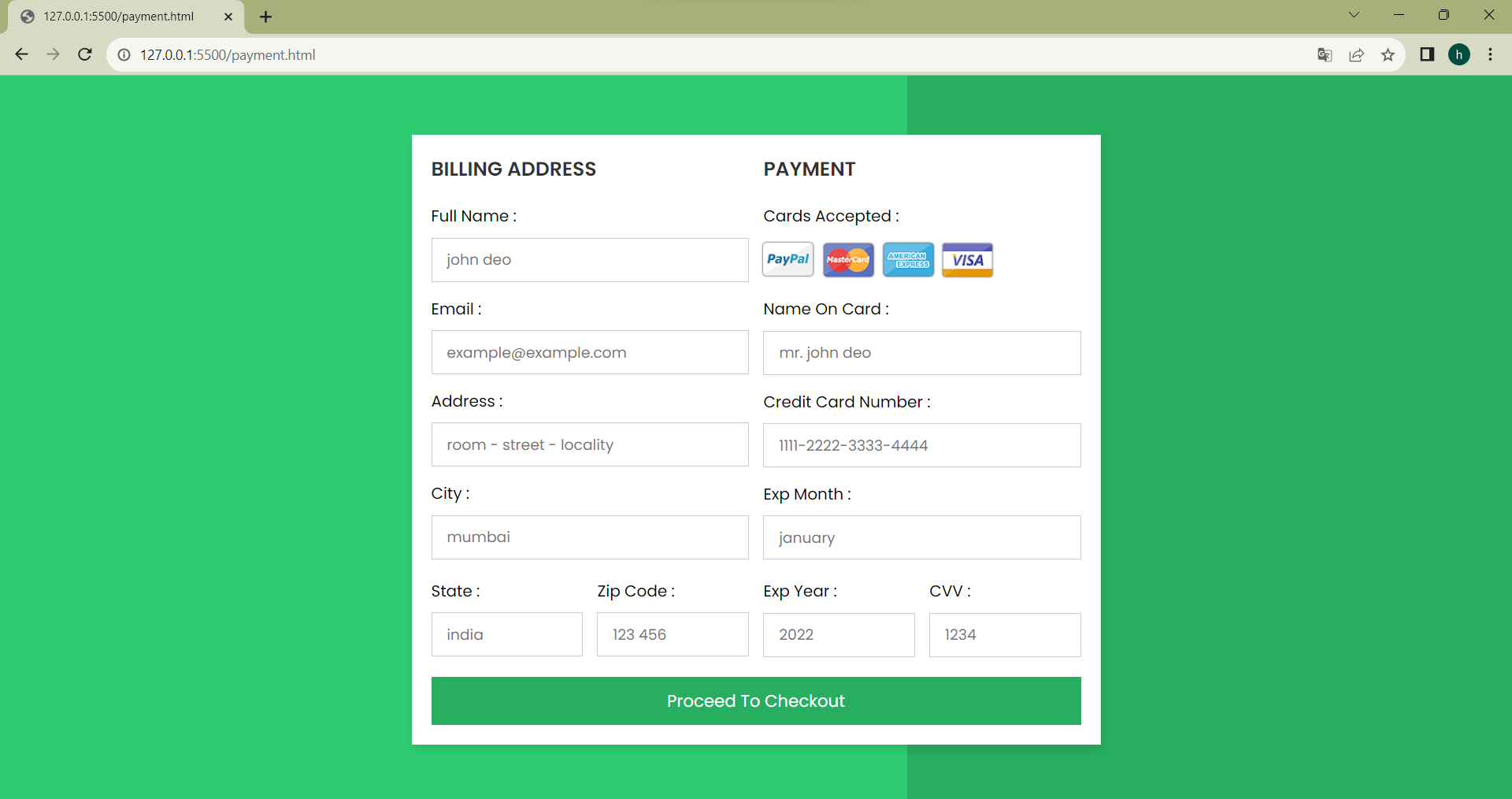
**Step 11: Monitor and Maintain**

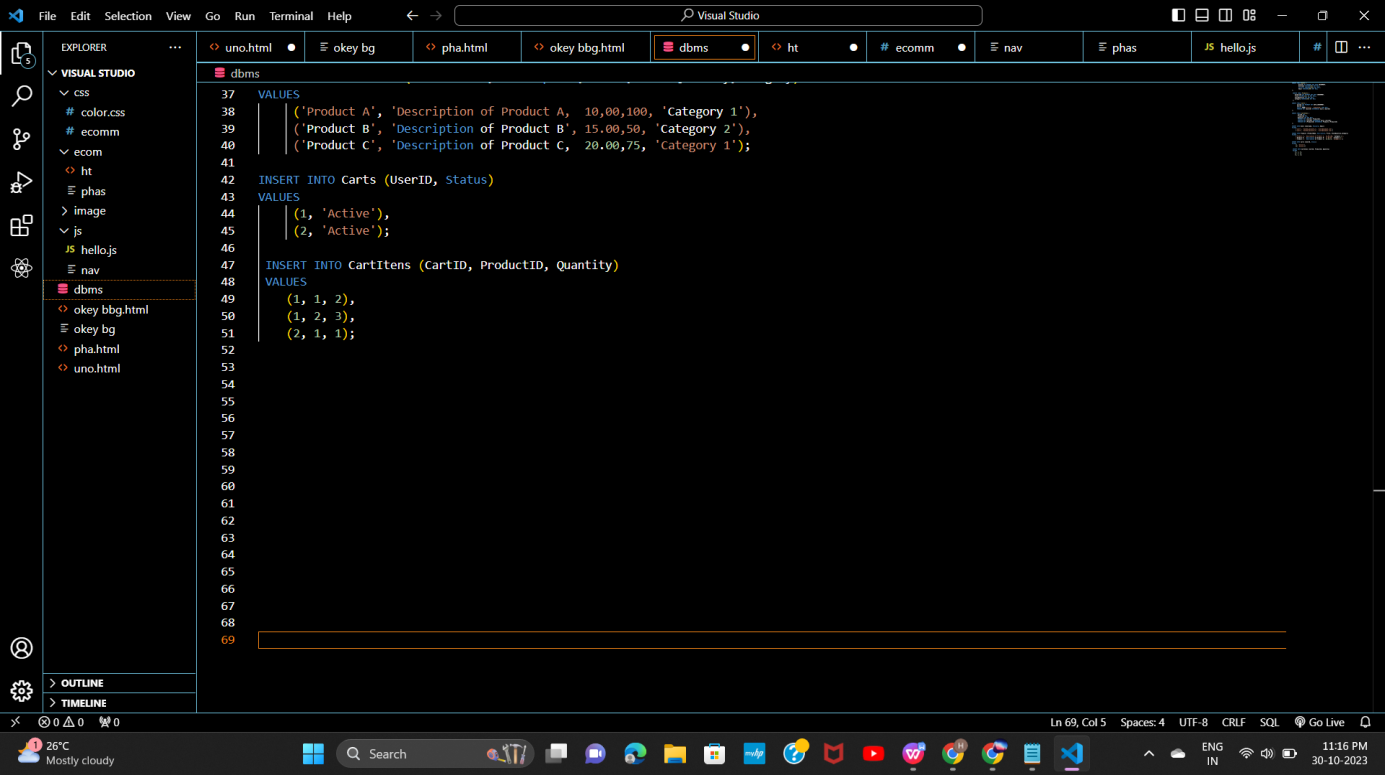
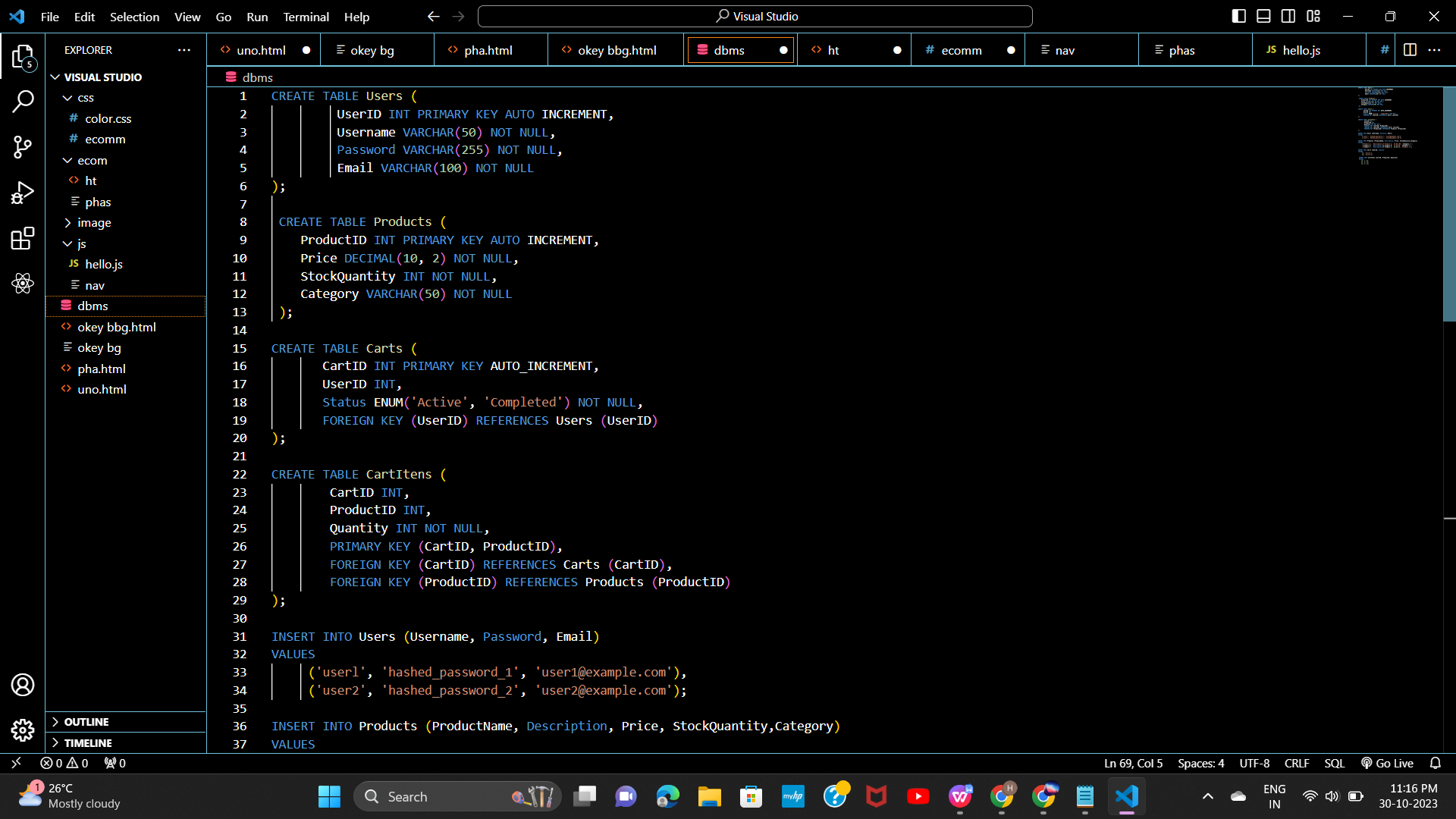
Regularly monitor your E-commerce backend for performance, security, and scalability. Implement updates and improvements as needed to ensure a smooth user experience.











**Conclusion**

In conclusion, building an E-commerce backend with node.js and MySQL is a dynamic process. This combination provides a strong foundation for secure and scalable platform and we have created the development environment, design the database, create API endpoints, implement user authentication, integrate a payment gateway, checkout process and deploy your application. Keep in mind that ongoing maintenance and adaptation to industry trends are key to sustained success in the competitive E-commerce landscape.