DIVISION OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

Academic Year 2024-2025
ODD SEMESTER

Skill-based Evaluation Report

(III Internal Assessment)

for

20CS2016

Database Management Systems

in partial fulfillment for the award of the degree of
BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
(ARTIFICIAL INTELLIGENCE &
MACHINE LEARNING)

Submitted by
Anson Saju George (URK22CS7064)
Jeril Joseph (URK22CS7086)
Sai Midhun Jayan (URK22CS7090)

Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)
MoE, UGC & AICTE Approved
NAAC A++ Accredited

October 2024

Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES

(Declared as Deemed to be University under Sec.3 of the UGC Act, 1956)

MoE, UGC & AICTE Approved

NAAC A++ Accredited

DIVISION OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that the skill-based evaluation report for the course **20CS2016 Database Management Systems** is a bonafide work done during the odd semester of the academic year 2024-2025 by

Anson Saju George (URK22CS7064)
Jeril Joseph (URK22CS7086)
Sai Midhun Jayan (URK22CS7090)

in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence) of Karunya Institute of Technology and Sciences.

Submitted for the assessment held on 22.10.2024

Dr. D.Narmadha
Assistant Professor
Signature of the Class Teacher

DECLARATION

We hereby declare that the skill-based evaluation report for the course **20CS2016 Database Management Systems** submitted by us to Karunya Institute of Technology and Sciences is an authentic work carried out by us under the guidance of **Dr. D. Narmadha**. All the information presented in this report is original and has not been submitted for any other academic purpose.

We acknowledge that the sources of information used in this report have been appropriately cited and referenced. Any contributions from other individuals or organizations have been duly acknowledged.

We understand that any act of plagiarism or academic dishonesty is strictly prohibited and may result in disciplinary action as per the university's regulations.

We take full responsibility for the content and integrity of this report, and we are willing to provide further clarification or information if required.

Signature of student(s)

Anson Saju George (URK22CS7064) **Jeril Joseph (URK22CS7086)** **Sai Midhun Jayan (URK22CS7090)**

DIVISION OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

TYPE OF SKILLS AND THEIR REQUIREMENT

(The student team may select multiple skills and proceed accordingly with their work)

Select your choice	Type of Generic Skills	Requirement
❑	Programming Skills	Assessing proficiency in programming languages commonly used in engineering, such as Python, C, C++, Java, etc.,
❑	Laboratory Proficiency	Assessing the ability to conduct experiments accurately and efficiently in laboratory settings.
❑	Prototyping and Fabrication	Evaluating skills in building prototypes and fabricating engineering components using various tools and materials.
❑	Computer-Aided Design (CAD)	Measuring proficiency in using CAD software to design and model engineering components and systems.
❑	Software KitrTool Operation	Assessing competency in operating specialized engineering software effectively.
❑	Data Analysis	Measuring proficiency in collecting, analyzing, and interpreting data using statistical methods and software.
❑	Troubleshooting	Evaluating the ability to diagnose and solve technical issues and failures in engineering systems.
❑	Fieldwork and Industry visits	Assessing skills in conducting fieldwork, surveys, and site visits to gather data and assess real-world engineering challenges.
❑	Integration of Theory and Practice	Evaluating the ability to apply theoretical knowledge to practical engineering problems and projects effectively.
❑	ConferencerJournalrPatent publication	Writing high-quality articles and submitting them for publication in Scopus-indexed conferences or journals involves careful preparation and rigorous review processes.

Signature of the Student(s)

Anson Saju George (URK22CS7064) **Jeril Joseph (URK22CS7086)** **Sai Midhun Jayan (URK22CS7090)**

DIVISION OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTER SCIENCE AND TECHNOLOGY

EVALUATION REPORT

(The class teacher is responsible for assessing the student team during the designated evaluation period based on the following criteria.)

S.No	Evaluation Parameter	Max. Marks	Marks Secured
1	Technical Proficiency	10	
2	Problem-solving Ability	5	
3	Creativity and Innovation	5	
4	Communication Skills	5	
5	Collaboration and Teamwork	5	
6	Quality of Implementation	5	
7	Demonstration and Presentation	5	

Signature of the Class Teacher

Dr. D. Narmadha
Assistant Professor
Division of Artificial Intelligence and Machine Learning

TABLE OF CONTENTS

S.No	Content Title	Page No
1	Introduction	1
2	Planning and preparation	6
3	Key components of the identified skill sets	10
4	Proofs of outcome and feedback	12
5	Powerpoint Presentation	17
6	Conclusion	22

SPICE E-COMMERCE MANAGEMENT SYSTEM

Introduction to Spice E-commerce Management System:

This project focuses on the development of a robust and dynamic database-driven website for a fictional spice-selling business, catering to the growing demand for online shopping in today's digital age. As e-commerce continues to revolutionize traditional business models, having a well-structured and user-friendly platform has become essential for any business aiming to reach a broader audience and streamline operations. In response to this need, our project aims to design and implement an intuitive online platform where customers can browse, select, and purchase a variety of spices while ensuring that the business can efficiently manage inventory, customer accounts, and orders through a centralized database.

To achieve these objectives, we employed MySQL as the core relational database management system (RDBMS). MySQL allows for efficient storage, retrieval, and management of crucial data such as product details, customer information, and order histories. We also chose Python for back-end development, which acts as a bridge between the front-end user interface and the MySQL database. This combination allows for seamless interaction between the customer and the database, ensuring a smooth and responsive user experience throughout the shopping and checkout processes. The use of Python also enables advanced features like dynamic content rendering, personalized recommendations, and real-time updates to stock and order statuses.

One of the core features of the database design is the establishment of multiple table relations, ensuring a clear and organized separation of different data categories. These include customer information, product catalog, transaction history, and user authentication details. This structure not only maintains the integrity of the data but also enables efficient queries, reporting, and data analysis.

This report will explore the project's key objectives, the underlying system architecture, the database design principles, and the technology stack used to bring this website to life. We will also discuss the development challenges faced, such as ensuring data security, optimizing performance, and managing scalability, along with proposed solutions and future enhancements that could further improve the system's robustness and functionality.

Frontend Technologies: HTML, CSS, and JavaScript :

The front end of the spice-selling website was designed using a combination of HTML, CSS, and JavaScript. These core web technologies work in tandem to create a responsive, user-friendly, and aesthetically pleasing interface for customers browsing and purchasing spices online.

1. **HTML (Hypertext Markup Language)**
   HTML forms the structural backbone of the website. It defines the various elements and layout of the site, including product listings, navigation menus, contact forms, and shopping cart functionality. By organizing content into a clear and logical structure, HTML ensures that customers can easily navigate the site and interact with different features. Specific sections, such as product descriptions, categories, and checkout forms, are structured using semantic HTML5 elements to enhance accessibility and SEO optimization.
2. **CSS (Cascading Style Sheets)**
   CSS is used to style and visually enhance the HTML structure. By applying various styles such as fonts, colors, margins, padding, and layout techniques (e.g., flexbox or grid), the website achieves a modern and attractive design. In this project, CSS plays a crucial role in ensuring the website is responsive, meaning it adjusts seamlessly across different screen sizes and devices, from desktop monitors to smartphones. Media queries and adaptive layouts ensure the content remains user-friendly across various platforms. Additionally, CSS animations are employed to create subtle transitions,

hover effects, and visual feedback that improve the overall user experience.

3. **JavaScript**
   JavaScript is essential for adding interactivity and dynamic behavior to the website. In this project, JavaScript is utilized to handle key functionalities such as:
   - **Dynamic shopping cart updates:** Users can add or remove spices from their shopping cart in real time without reloading the page. This dynamic behavior enhances user engagement and improves the shopping experience.
   - **Form validation:** JavaScript ensures that users input valid data when signing up, logging in, or placing orders. By checking for errors before submission, it prevents issues like incomplete forms or invalid information.
   - **Search and filtering:** Customers can use search bars or filters to find specific spices quickly. JavaScript handles the real-time filtering of products based on user input, streamlining the process of locating desired items.
   - **Asynchronous operations:** Using JavaScript's AJAX functionality, the website can communicate with the server without refreshing the page. This is particularly useful for updating stock levels, order statuses, or loading new products dynamically as users scroll or interact with the site.

     By combining HTML, CSS, and JavaScript, the front end provides a cohesive and intuitive experience for customers, ensuring easy navigation, attractive visuals, and efficient interactivity. Together, these technologies play a vital role in bringing the spice-selling platform to life while maintaining a smooth and engaging user experience.

Backend Development with NodeJS :

For the backend development of the spice-selling website, **Node.js** was chosen as the primary runtime environment. Node.js is known for its fast, scalable, and efficient performance, making it an ideal choice for building web applications with real-time, data-intensive capabilities. In this project, the backend is responsible for handling server-side logic, managing communication with the database, and ensuring smooth interaction between the client and the server.

1. **Why Node.js?**
   Node.js operates on a single-threaded, non-blocking event-driven architecture, which makes it particularly efficient for handling multiple concurrent requests. Since the spice-selling website requires dynamic interactions between customers and the database, such as real-time updates to inventory, orders, and user accounts, Node.js is perfectly suited for ensuring quick and reliable responses without performance degradation. Additionally, using JavaScript on both the client and server sides provides consistency in the codebase, making development faster and more efficient.
2. **Express.js for Routing and Middleware Express.js**, a minimalist web application framework for Node.js, is used to manage the routing and middleware functions of the website. Express simplifies the process of setting up a robust API that handles various HTTP requests (GET, POST, PUT, DELETE) from the frontend. Key functionalities of Express in this project include:
   - **Routing:** Express is responsible for defining routes that map different URLs to specific server-side functions. For instance, it handles requests for viewing product pages, adding items to the cart, processing orders, and managing user authentication. Each route is linked to a corresponding controller, which executes the appropriate business logic based on the request type.
   - **Middleware:** Express makes extensive use of middleware functions to process incoming requests. Middleware is used for tasks such as parsing JSON data from form submissions, validating user input, logging errors, handling cookies, and managing sessions.
3. **Database Connectivity**
   Node.js interacts with the MySQL database using libraries such as **MySQL2** or **Sequelize** (an ORM tool). These libraries enable seamless communication between the backend and the database, allowing for CRUD (Create, Read, Update, Delete) operations to be executed efficiently. Key responsibilities of the backend include:

- **Fetching and updating product information:** When a customer browses the spice catalog or adds items to the cart, the backend queries the MySQL database to fetch the relevant product details, such as price, description, and stock availability.
- **Order processing:** When a customer places an order, the backend validates the input, checks product availability, calculates the total cost, and stores the transaction details in the database. It also triggers updates to the inventory to reflect the reduced stock.
- **User account management:** The backend handles customer registration, login, and profile updates by securely interacting with the database to store and retrieve user credentials and account data.

4. **Security and Authentication**

   Security is a crucial consideration in any e-commerce platform, and Node.js plays a significant role in implementing various security measures.

- **Password encryption:** The backend uses libraries like **bcrypt** to hash and securely store user passwords in the database. This ensures that even if the database is compromised, user credentials remain protected.
- **JWT (JSON Web Tokens):** For user authentication, **JWT** is used to generate and validate secure tokens when users log in. This token-based authentication allows users to remain logged in while interacting with the site without repeatedly submitting login credentials.
- **Input validation and sanitization:** The backend ensures that all user inputs, such as sign-up forms and payment details, are validated and sanitized to prevent injection attacks or malicious data entry.

5. **Real-Time Features**

   Node.js, with its non-blocking nature, allows the implementation of real-time features, which are important for enhancing the user experience. In this project:

   - **Real-time inventory updates:** As multiple customers interact with the website, Node.js ensures that inventory levels are updated in real-time across all sessions. If a product goes out of stock while a customer is browsing, the change is reflected dynamically without requiring the user to refresh the page.
   - **Notifications and order status updates:** Customers are notified of the status of their orders (e.g., "Processing," "Shipped") in real-time. Additionally, the backend can handle email notifications to update customers on order confirmations or shipping statuses using Node.js libraries like **Nodemailer**.

6. **APIs and Third-Party Integration**

   The backend also manages integration with external services and APIs that enhance the website's functionality:

- **Payment gateways:** Secure payment processing is a critical aspect of e-commerce. The backend integrates with payment gateways (such as Stripe or PayPal) to facilitate secure transactions and order processing.
- **Shipping services:** The backend can communicate with shipping APIs to calculate delivery costs based on the customer's location and provide real-time tracking information.

7. **Error Handling and Logging**

   Node.js ensures the website operates smoothly by implementing proper error handling mechanisms. When a customer encounters an issue, such as a failed payment or an invalid form submission, the backend gracefully handles the error, providing meaningful feedback to the user while logging the issue for further investigation. Express.js makes it easy to define error-handling middleware that captures and processes errors across all routes.

Database Management with MySQL :

In the spice-selling website project, **MySQL** serves as the relational database management system (RDBMS) responsible for handling the storage, organization, and retrieval of crucial business data. MySQL's structured approach to data, combined with its ability to efficiently manage relationships between various entities, makes it an ideal choice for e-commerce platforms. It ensures that the website can seamlessly handle customer information, product details, orders, and transaction histories.

1. **Why MySQL?**
   MySQL was selected for its reliability, scalability, and extensive support for relational data. Its SQL-based query language is powerful for managing complex data structures while maintaining data integrity through relationships between tables. MySQL also supports ACID (Atomicity, Consistency, Isolation, Durability) properties, which are essential for handling sensitive data, such as customer information and financial transactions.
2. **Database Design and Structure**
   The MySQL database is designed with multiple tables to handle different aspects of the business. Each table is dedicated to storing specific types of information, such as products, customers, orders, and transactions. By designing the database with normalized tables, we avoid redundancy, maintain consistency, and ensure data integrity.

   Key tables in the database include:

Products Table:

This table stores information about the spices available for sale, including the product ID, name, description, price, category, and stock quantity. The primary key is the `product_id`, which uniquely identifies each product. Other attributes include:

`name`: The name of the spice.

`description`: A brief description of the spice's flavor profile or use.

`price`: The price of the spice.

`stock_quantity`: The current inventory level of the product.

Customers Table:

The customers table holds personal and account information for registered users. The primary key is the `customer_id`, which uniquely identifies each customer. Fields include:

`name`: The full name of the customer.

`email`: The email address, used for login and communication.

`password_hash`: The encrypted password for secure authentication.

`address`: Shipping and billing address information.

Orders Table:

This table tracks customer orders and links to both the products and customers tables via foreign keys. Each entry corresponds to a specific order placed by a customer, and the primary key is the `order_id`. Attributes include:

`order_date`: The date and time the order was placed.

`customer_id`: A foreign key that links to the `customer_id` in the customers table, identifying who placed the order.

`total_amount`: The total amount of the order, calculated based on the products ordered.

Order Details Table:

This table captures the details of each order, including which products were purchased and in what quantity. It forms a many-to-many relationship between the orders and products tables, with the following key fields:

`order_id`: A foreign key referencing the orders table.

`product_id`: A foreign key referencing the products table.

`quantity`: The number of units of a specific product purchased in the order.

Transactions Table:

The transactions table logs payments made by customers, recording details like payment method, transaction date, and status. Attributes include:

`transaction_id`: The primary key to uniquely identify each transaction.

`order_id`: A foreign key linking to the corresponding order.

`payment_method`: Information about how the customer paid (e.g., credit card, PayPal).
`transaction_status`: Whether the payment was successful or failed.

3. **Data Relationships and Normalization**
   MySQL's relational database model enables the establishment of relationships between tables, which helps prevent data duplication and ensures that related data can be retrieved efficiently. This design adheres to the principles of database normalization, ensuring minimal redundancy and avoiding anomalies during data insertion, updates, and deletion.

- **One-to-Many Relationships:**
  The relationship between the customers and orders tables is a one-to-many relationship. Each customer can place multiple orders, but each order is linked to only one customer. Similarly, the orders table hasa one-to-many relationship with the order details table, where a single order can include multiple products.
- **Many-to-Many Relationships:**
  The relationship between products and orders is many-to-many, as one product can be included in multiple orders, and each order can contain multiple products. This relationship is captured by the **order details table**, which links products and orders through foreign keys.

4. **Queries and Transactions**
   MySQL allows for complex SQL queries to retrieve and manipulate data from multiple tables. Some of the common queries used in this project include:

- **Product Search and Filtering:** Customers can search for products by name, category, or price range. SQL `SELECT` queries with conditions (such as `WHERE`, `LIKE`, or `BETWEEN`) are used to filter and display products based on the user's preferences.
- **Customer Orders and History:** When a user logs in, they can view their past orders. SQL joins between the customers, orders, and order details tables retrieve the complete order history and corresponding details.
- **Inventory Management:**
  SQL `UPDATE` queries ensure that inventory levels are adjusted in real time as products are purchased or restocked. This prevents customers from purchasing out-of-stock items.

  MySQL also supports **transactions**, which allow multiple SQL operations to be executed as a single unit. In the context of order processing, transactions are essential to ensure data consistency. For instance, when a customer places an order, the following steps must all succeed, or none of them should be applied:

- Deduct the ordered quantities from the product stock.
- Create a new entry in the orders table.
- Insert the corresponding records into the order details table. If any part of this process fails (e.g., a product goes out of stock), the transaction is rolled back to its initial state, ensuring data consistency.

5. **Data Integrity and Constraints** MySQL supports a range of integrity constraints that ensure the accuracy and reliability of the data. Common constraints used in this project include:

- **Primary Key Constraints:** Ensure that each record in a table is uniquely identifiable, such as the `product_id` in the products table and the `order_id` in the orders table.
- **Foreign Key Constraints:** Enforce relationships between tables, ensuring that data remains consistent across related tables. For instance, an `order_id` in the order details table must reference a valid entry in the orders table.
- **Check Constraints:** These constraints validate data according to specific rules. For example, the price of a product must always be a positive value, and the stock quantity must never be negative.

6. **Data Security and Backup** Since the website handles sensitive information, such as customer details and transaction data, ensuring data security is a top priority. MySQL's security features include:

- **User Roles and Permissions:** Specific roles (e.g., admin, regular user) are created, and permissions are granted to control who can access or modify certain data. Only administrators can access and modify the product catalog, while customers can view their orders but not those of others.

- **Data Encryption:** MySQL supports data encryption techniques, ensuring that sensitive information like passwords and payment details is stored securely. For example, passwords are stored as hashed values using secure algorithms.
- **Regular Backups:** To prevent data loss, regular database backups are scheduled. MySQL provides mechanisms for full and incremental backups, ensuring that the database can be restored in the event of hardware failure or data corruption.

Primary objective:-

The primary objectives of the spice-selling website project are as follows:

- **Develop a responsive and user-friendly interface:**
  Design and implement a responsive front-end that allows customers to seamlessly browse through the spice catalog, view detailed product descriptions, and add items to the shopping cart. The interface should adapt to different screen sizes and devices (e.g., mobile phones, tablets, desktops) to ensure a consistent user experience.
- **Implement a robust login and sign-up system:**
  Build a secure user authentication system that allows customers to register and log in with confidence. The system must ensure the protection of sensitive user data by employing password encryption and prevent unauthorized access through secure login mechanisms, such as encrypted passwords and session management.
- **Design and develop a back-end database with MySQL:**
  Create a well-structured and normalized database using MySQL to efficiently store, manage, and retrieve information related to users, products, orders, and transaction histories. The database must be optimized for performance and designed to maintain data integrity, ensuring smooth and error-free operations.
- **Build a smooth interaction layer using Node.js (or Python):**
  Establish a reliable communication layer between the front-end and back-end using **Node.js** (or **Python**), enabling real-time interaction between the user interface and the database. This includes handling user requests, updating product information dynamically, managing shopping cart operations, and processing orders securely.
- **Ensure seamless and secure operation of all key functionalities:**
  Guarantee that the core operations, such as user registration, product browsing, adding items to the cart, order placement, and payment processing, are smooth, error-free, and secure. This includes implementing proper error handling, form validation, and real-time updates, ensuring a flawless user experience.
- **Address data security concerns:**
  Ensure that user data, including personal information and transaction details, is securely stored and transmitted. The project will utilize industry-standard security practices such as data encryption, secure communication (e.g., HTTPS), and secure database access control.
- **Optimize performance with proper database indexing:**
  Implement efficient indexing and database optimization techniques to improve query performance and reduce the time required to retrieve data, especially as the number of customers, products, and orders increases over time. Proper indexing will also help in managing and scaling the system as the website grows.

  In addition to these core objectives, the project aims to address key challenges such as user input validation to prevent data entry errors, ensuring data consistency and avoiding redundant or conflicting information. Through these objectives, the spice-selling website aims to provide an engaging, secure, and efficient platform for both customers and administrators.

System Architecture

The system architecture of the spice-selling website is based on the **three-tier architecture model**, a widely-used approach in web applications. This model separates the system into three distinct layers:

the **front-end (presentation layer)**, the **middle-tier (application layer)**, and the **back-end (data layer)**. Each layer is responsible for specific functions, ensuring modularity, scalability, and maintainability.

1. Front-End Layer (Presentation Layer)

The front-end of the website is designed to offer a seamless and engaging user experience. It is developed using:

- **HTML (Hypertext Markup Language):** Provides the structure and layout of the web pages. It defines elements such as headers, product listings, forms, and buttons, creating a clear and navigable interface.
- **CSS (Cascading Style Sheets):** Adds styling to the HTML structure, ensuring that the website is visually appealing and responsive. CSS is used to apply fonts, colors, spacing, and layout adjustments, making the site look modern and professional across different devices.
- **JavaScript:** Enhances the interactivity and dynamic behavior of the website. JavaScript handles tasks such as form validation, dynamic cart updates, and real-time product filtering. It also improves user experience by enabling the website to respond to user actions without requiring full-page reloads.

The front-end layer is responsible for displaying data fetched from the back-end and interacting with users. This includes collecting user inputs, such as login credentials, product selections, and payment information, which are then passed to the middle-tier for processing.

2. Application Layer (Middle-Tier)

The middle-tier, also known as the **application layer**, serves as the intermediary between the front-end and the back-end. This layer is developed using **Node.js**, a JavaScript runtime environment that excels in building scalable, high-performance applications. The key functions of this layer include:

- **Handling User Requests:** Node.js processes HTTP requests from the front-end (e.g., fetching product details, managing the shopping cart, or handling user authentication) and returns appropriate responses.
- **Business Logic Execution:** The application layer contains the business logic necessary to handle operations such as order processing, user registration, and inventory management. It validates user input, calculates total prices, applies discounts, and manages session handling.
- **API and Server-Side Logic:** Node.js works with **Express.js**, a web application framework, to manage routing and middleware. It defines routes that map user requests to specific back-end functions and serves as the API endpoint for handling communication between the front-end and the database.
- **Security and Authentication:** This layer also manages user authentication, including password hashing, session management, and token-based authentication (JWT), ensuring secure access to user data and preventing unauthorized access.

3. Back-End Layer (Data Layer)

The back-end, or **data layer**, is powered by **MySQL**, a relational database management system that stores and manages all the essential data for the website. This layer handles:

- **Data Storage:** All critical information—such as product details, customer accounts, orders, and transaction histories—is stored in the database. The data is stored in a normalized format across multiple tables, ensuring that it is well-organized and easy to manage.
- **Data Relationships and Integrity:** Tables are linked using foreign keys to maintain referential integrity between related data (e.g., linking customers to their orders). This ensures that the database remains consistent, with no orphaned records or redundant data.
- **CRUD Operations:** The back-end layer performs CRUD (Create, Read, Update, Delete) operations. These operations are crucial for tasks such as adding new products, retrieving order history, updating customer profiles, and deleting outdated records.
- **Data Security:** MySQL incorporates several security measures to protect sensitive information, such as encrypted passwords, access control using roles and privileges, and regular database backups. The system ensures that only authorized users can access and modify specific data.

The back-end layer ensures data consistency, reliability, and security, which are critical to the overall functioning of the website. This layer communicates with the application layer to retrieve, update, or

store data as needed.

Front-EndDevelopment:

The front-end design is a critical component of any e- commerce website. For this project, we aimed to create a simple yet efficient interface that is easy to navigate. Key pages include the homepage, product listing page, product detail page, user login/sign-up pages, and the shopping cart/checkout pages.

- **Homepage**: The homepage features a clean design with sections highlighting popular or featured spices. The use of images and brief descriptions helps capture user attention. Customers can navigate to different product categories or use the search bar to find specific spices.
- **Product Page**: Each product listed on the site includes detailed information such as price, description, available stock, and customer reviews (to be implemented in future versions). Users can add items to their cart directly from this page, which dynamically updates the total cost.
- **Login/Sign-Up Page**: These pages are designed with user security in mind. The forms validate inputs to ensure that only correct information is submitted. Passwords are encrypted before being sent to the back-end for storage in the database.
  Shopping Cart & Checkout Page: Users can view their selected items in the cart, make adjustments, and proceed to checkout. The checkout page collects necessary shipping details and provides an overview of the order before final submission.

Database Design:

The core of the project lies in the database design. The database is implemented in MySQL, a relational database management system. The design focuses on the effective management of data, ensuring that it can handle large volumes of transactions efficiently.

- **Customers Table**: This table stores user information such as the customer ID (primary key), name, email, and password. The password is stored in an encrypted format to maintain security. The customer ID is used to link this table with other tables, such as the orders table, to track the customer's activity on the website.
- **Products Table**: This table contains detailed information about the spices available for sale. Fields include product ID (primary key), name, description, price, and stock level. The product ID is referenced in the orders table to identify which items have been purchased.
- **Orders Table**: The orders table is used to track each transaction made on the website. It includes fields such as order ID (primary key), customer ID (foreign key from the customers table), product ID (foreign key from the products table), and quantity. This table ensures that all orders are correctly recorded and linked to the respective customer and product.
- **Login and Sign-Up Table**: This table manages user authentication by storing login credentials such as email and password. The password is stored in a hashed format

Backend Development:

The back-end development is primarily focused on the management of data and the seamless interaction between the website's user interface and the MySQL database. Python serves as the intermediary language that connects these two layers. For this project, we used the MySQL connector library in Python to manage the database interactions.
Some key back-end functions include:

- **User Registration and Login**: The system allows users to register by entering their personal details, which are then securely stored in the MySQL database. During login, the entered credentials are checked against the stored records. Passwords are hashed using secure algorithms, ensuring that sensitive data remains protected.
- **Product Display and Search**: Python is used to query the database and retrieve product details. These

details are then displayed on the product pages, allowing users to browse and search for the desired spices.

- **Order Placement**: When a customer completes a purchase, the order details are recorded in the orders table. Python handles the communication between the website and the database to ensure that the correct customer ID and product IDs are associated with the order.

Data Flow and Connectivity:

The data flow in the project starts with user interaction on the front-end and proceeds to the back-end where Python scripts process the user requests. Once a request is initiated (e.g., a login attempt or product search), Python retrieves the relevant information from the MySQL database and returns the result to the front-end.

For instance, when a user enters their login details, the system checks the entered data against the records in the login table. If the credentials are correct, the user is granted access to their account.

Similarly, when users browse products, Python queries the products table in MySQL to fetch the necessary data.

Data flow follows a structured process:

- **User Input**: The front-end collects inputs from the user (such as login details or product selections).
- **Data Validation**: The inputs are first validated on the front-end to ensure they meet the required criteria (e.g., email format, password length).
- **Backend Processing**: Validated inputs are passed to the Python back-end, where SQL queries are executed to retrieve or update data in the MySQL database.
- **Response to Front-End**: The results of the database operations are returned to the front-end for display (e.g., confirmation of a successful login or product listing).

This real-time data flow ensures the website's responsiveness and allows users to interact with the system without delays.

Security Features:

Security is a vital aspect of any online system, especially those that handle sensitive user information such as passwords and transaction details. In this project, we implemented several security measures to protect user data.

- **Password Encryption**: All user passwords are encrypted using a hashing algorithm before being stored in the database. This ensures that even if the database is compromised, the actual passwords cannot be easily retrieved or misused.
- **SQL Injection Prevention**: To prevent SQL injection attacks, which are common in web applications, we have employed parameterized queries. This ensures that any user input is treated as data rather than part of the SQL query, thereby avoiding potential security breaches.

Input Validation: Both client-side and server-side input validation are implemented to ensure that all user inputs are legitimate and free of malicious code. This helps protect the website from various attacks, including cross- site scripting (XSS) and injection attacks

Python-MySQL Connection:

Python acts as the middleware that connects the front-end of the website with the MySQL database. The MySQL connector library in Python allows the execution of SQL queries from within Python scripts. This connection is established using database credentials (host, username, password, and database name) stored securely within the application.

For example, the following Python code snippet illustrates how the connection is established and a query is executed to retrieve product details:

```python
import mysql.connector

# Establishing connection to MySQL database
db_connection = mysql.connector.connect(
    host="localhost",
    user="root",
    password="password",
    database="spices_db"
)

# Creating a cursor object to interact with the database
cursor = db_connection.cursor()

# Query to fetch all products
query = "SELECT * FROM products"
cursor.execute(query)

# Fetching and printing the results
products = cursor.fetchall()
for product in products:
    print(product)
```

Testing and Debugging:

Testing is an integral part of any software development project, as it ensures that the system functions as expected under various scenarios. We conducted multiple levels of testing to verify the functionality, security, and performance of the website.

- **Functionality Testing**: Each feature of the website, such as user login, product browsing, and order placement, was thoroughly tested to ensure correct operation. Edge cases were considered, such as attempting to log in with incorrect credentials or trying to purchase an out-of-stock product.
- **Security Testing**: The security of user data was tested by simulating SQL injection attacks and verifying that the input validation mechanisms were effective. Password encryption was also tested to ensure that sensitive information is stored securely.
- **Performance Testing**: The performance of the database queries was analyzed to ensure quick retrieval of data, even when handling large amounts of records. Proper indexing of tables helped improve query execution times, and Python's efficient handling of SQL queries ensured smooth data flow between the front-end and the database.

Challenges Faced:

Throughout the development of the project, we encountered several challenges, including:

• Managing Multiple Table Relationships: Designing the database with multiple interrelated tables was initially complex. Ensuring that the foreign keys and relationships between tables were correctly set up to maintain data integrity required careful planning and execution.

• Real-Time Data Updates: Since users are constantly interacting with the website, ensuring real-time data updates without performance lags was a challenge. We optimized SQL queries and Python scripts to handle multiple requests simultaneously, ensuring a smooth user experience.

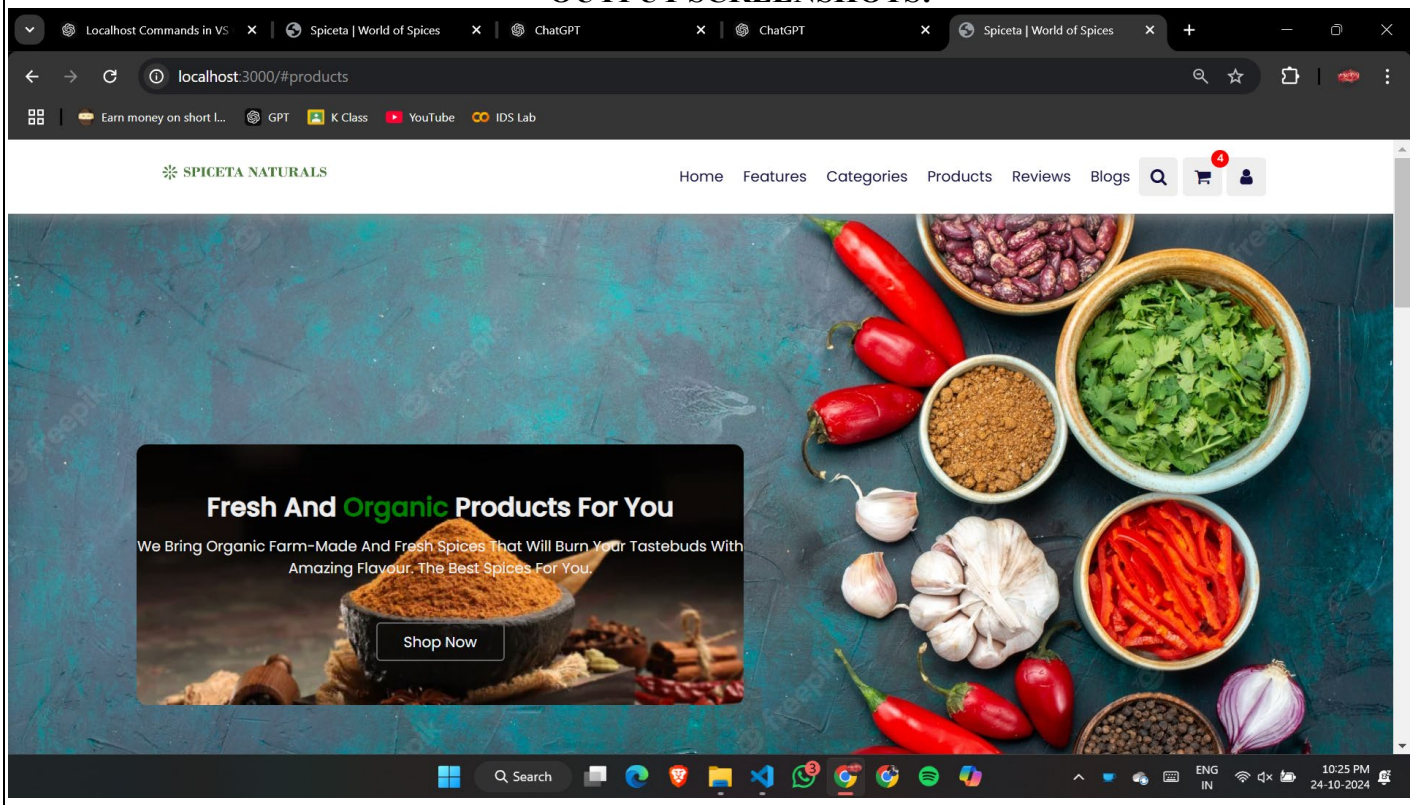
• User Input Validation: Ensuring that all user inputs were valid and secure was another challenge. We implemented both client-side and server-side validation to ensure that no invalid or malicious data could enter the system.

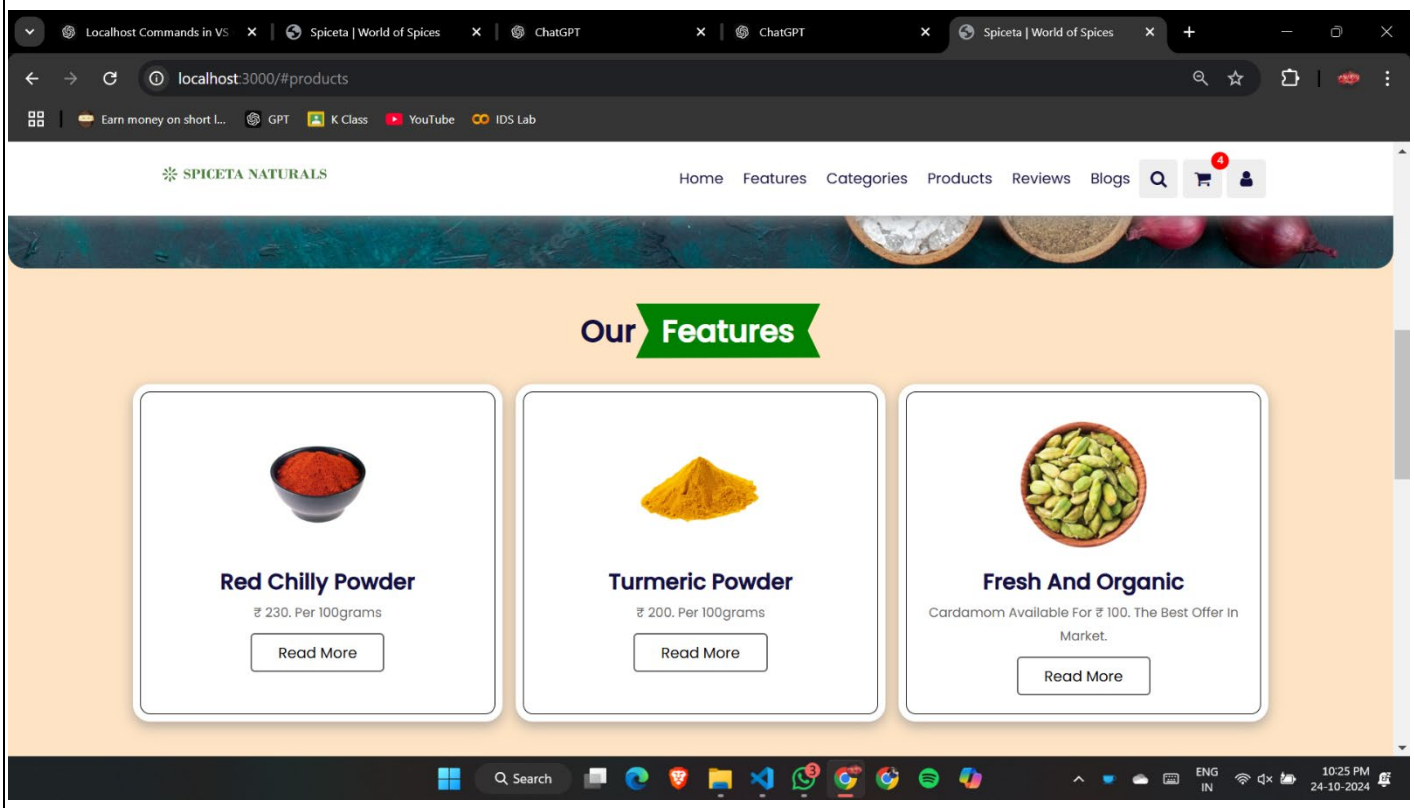
Future Work:

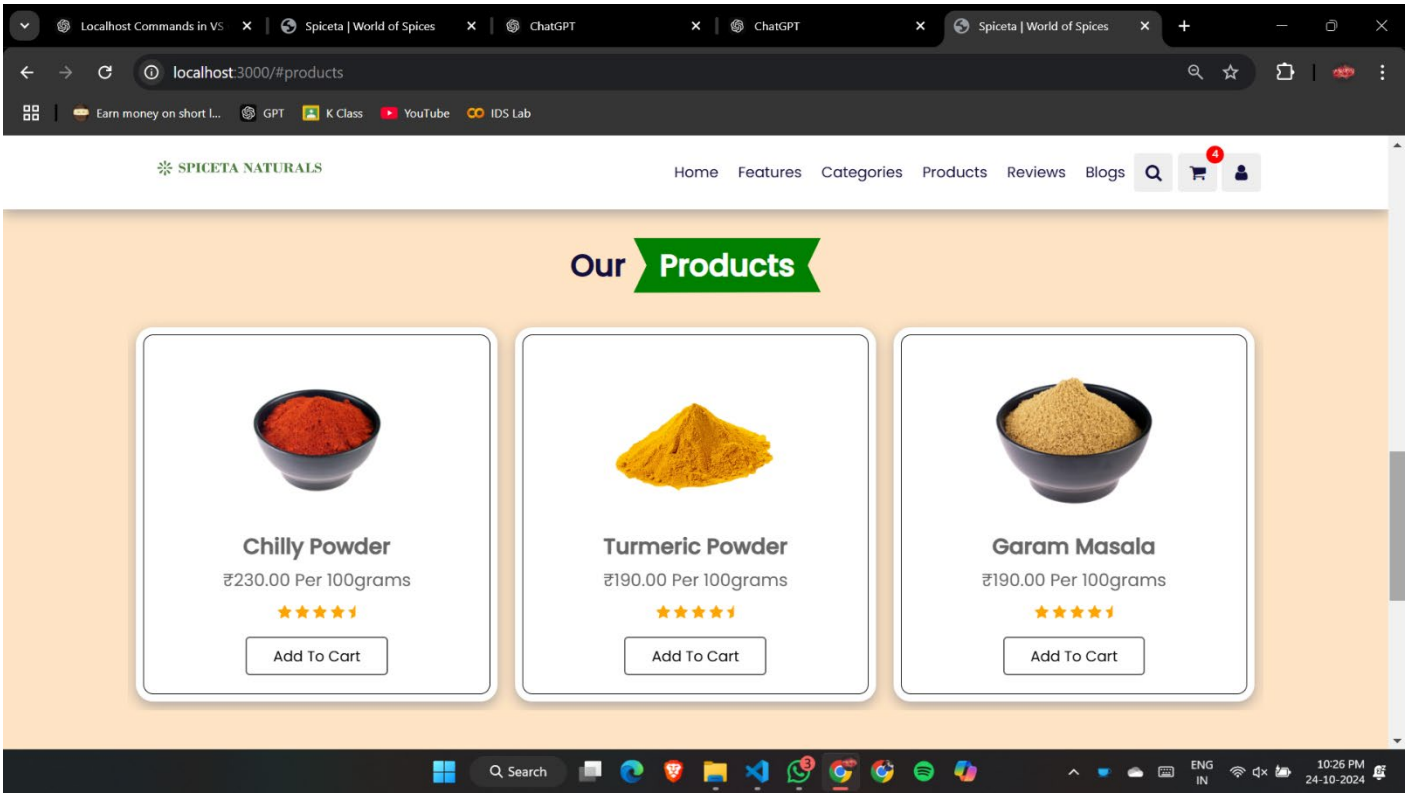
While the current version of the project meets the basic requirements of a spice-selling e-commerce website, there are several enhancements that could be made in future iterations:

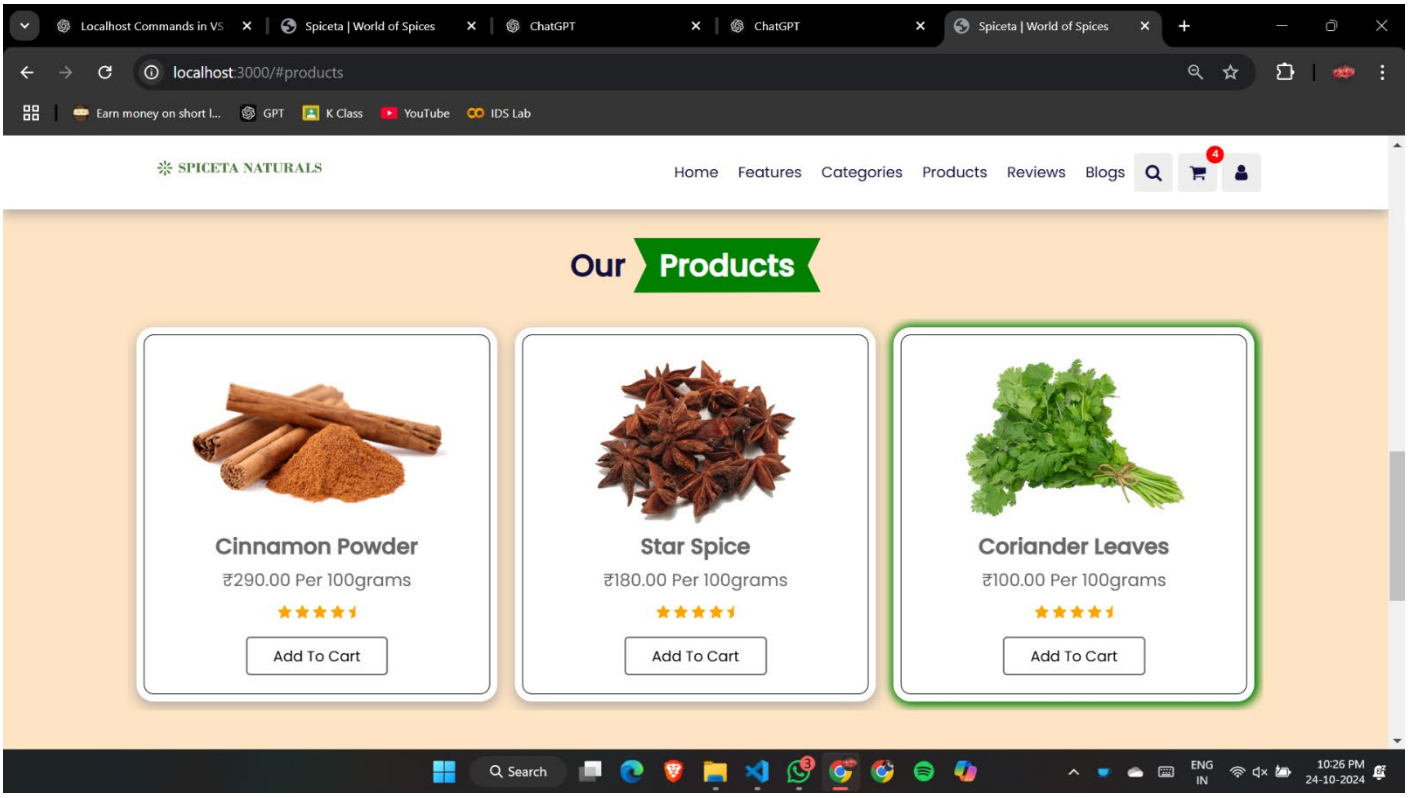
- **Product Recommendations**: Implementing a recommendation engine that suggests spices to users based on their browsing or purchasing history.
- **Payment Gateway Integration**: Adding secure payment processing functionality to allow users to pay directly through the website.
- **Mobile Optimization**: Enhancing the front-end design to ensure full compatibility and usability on mobile devices, as more users are now accessing e-commerce platforms through their smartphones.
- **User Reviews and Ratings**: Allowing users to leave reviews and ratings for products, which could help other customers make informed purchasing decisions.
- **Real-Time Inventory Management**: Improving the inventory system to reflect real-time stock updates and notify customers when products are out of stock or back in stock.

OUTPUT SCREENSHOTS:






Localhost Commands in VS
Spiceta | World of Spices
ChatGPT
ChatGPT
Spiceta | World of Spices
localhost:3000/#products
Earn money on short l...
GPT
K Class
YouTube
IDS Lab
SPICETA NATURALS
Home
Features
Categories
Products
Reviews
Blogs
Our Products
Chilly Powder
₹230.00 Per 100grams
Add To Cart
Turmeric Powder
₹190.00 Per 100grams
Add To Cart
Garam Masala
₹190.00 Per 100grams
Add To Cart
Search
ENG
IN
10:26 PM
24-10-2024


Localhost Commands in VS
Spiceta | World of Spices
ChatGPT
ChatGPT
Spiceta | World of Spices
localhost:3000/#products
Earn money on short l...
GPT
K Class
YouTube
IDS Lab
SPICETA NATURALS
Home
Features
Categories
Products
Reviews
Blogs
Our Products
Cinnamon Powder
₹290.00 Per 100grams
Add To Cart
Star Spice
₹180.00 Per 100grams
Add To Cart
Coriander Leaves
₹100.00 Per 100grams
Add To Cart
Search
ENG
IN
10:26 PM
24-10-2024

COURSE COMPLETION CERTIFICATE:

Proof of Completion

Congratulations to

JERIL JOSEPH URK22CS7086

For successfully completing

MongoDB Python Developer Path

On 09-30-2024

Sahir Azam
CPO
MongoDB, Inc

MongoDB.

MDB5xzie53f4k

Proof of Completion

Congratulations to

SAI MIDHUN JAYAN URK22CS7090

For successfully completing

MongoDB Python Developer Path

On 09-30-2024

Sahir Azam
CPO
MongoDB, Inc

MongoDB.

bSSDF5iDSD3f4k

POWER POINT PRESENTATION (PPT):

SPICETA NATURALS

E-COMMERCE WEBSITE DATABASE

Presented By:-

Anson Saju George, Jeril Joseph and Sai Midhun Jayan

Role of DBMS

- Ensures data consistency across all operations.
- Provides robust security measures to protect sensitive information.
- Enhances efficiency in data retrieval and storage, improving overall website performance

Issues Faced

- **Challenges Faced by Spiceta Naturals Without a Proper Database:**
  - **Inventory Management:**
    - Difficulty tracking stock levels of diverse spice products, leading to overstocking or stockouts.
- **Handling High Traffic:**
  - Inability to efficiently manage peak traffic periods during festivals or promotional sales.
- **Data Security:**
  - Risks of unauthorized access to user information and payment details.
- **Search and Recommendation Performance**
  - Poor user experience due to slow or inaccurate search results and lack of personalized product recommendations.

Overview of Spiceta Naturals

- An online platform dedicated to offering a wide variety of natural spices and spice blends.
- Serves both individual consumers and bulk buyers like restaurants and retailers.

Importance of Databases in E-commerce:-

- Efficient management of extensive product catalogs, including different spices, packaging sizes, and suppliers.
- Handling user data such as customer profiles, preferences, and order histories.
- Managing transactions securely and reliably, ensuring smooth order processing and payment handling.

Objective

- Design a Scalable and Efficient Database:
  - Cater to the growing number of products and users.
- Seamless Management of Core Entities:
  - Products, users, orders, payments, and reviews.
- Support Fast Queries and Transactions:
  - Ensure quick response times for user actions and administrative tasks.
- Guarantee Security and Privacy:
  - Protect sensitive data through encryption and access controls.

Database Requirements

- **Functional Requirements**:
  - **User Management**:
    - Registration, login, profile updates, and password management.
  - **Product Management**:
    - Adding, updating, and deleting spice products, managing categories and suppliers.
  - **Order Processing**:
    - Tracking orders, modifying order details, and managing order statuses.
  - **Payment Handling**:
    - Secure processing of payments, handling refunds, and managing payment statuses.
- **Non-Functional Requirements**:
  - **High Availability and Fault Tolerance**:
    - Ensure the website remains operational with minimal downtime.
  - **Data Integrity and Security**:
    - Maintain accurate and consistent data, protect against breaches.
  - **Performance Optimization**:
    - Implement indexing and caching for faster data retrieval.
  - **Scalability**:
    - Ability to handle an increasing number of users and transactions seamlessly.

Relational Schema

Tables and Attributes:

- **Users**:
  - user_id (PK), name, email, password, role, address, phone_number
- **Products**:
  - product_id (PK), name, description, price, stock, category_id (FK), supplier_id (FK)
- **Categories**:
  - category_id (PK), category_name, description
- **Suppliers**:
  - supplier_id (PK), supplier_name, contact_info
- **Orders**:
  - order_id (PK), user_id (FK), order_date, status, total_amount, shipping_address
- **OrderDetails**:
  - order_id (FK), product_id (FK), quantity, price
- **Payments**:
  - payment_id (PK), order_id (FK), payment_method, payment_status, transaction_date
- **Reviews**:
  - review_id (PK), user_id (FK), product_id (FK), rating, review_text, review_date

Security Measures

- **Data Encryption:**
  - Encrypt sensitive information such as user passwords and payment details using strong encryption algorithms.
- **Role-Based Access Control (RBAC):**
  - Define roles (e.g., Admin, Supplier, Customer) with specific permissions to restrict access to sensitive data and functionalities.
- **Backup and Recovery:**
  - Implement regular database backups and establish recovery procedures to prevent data loss in case of failures.
- **Input Validation:**
  - Sanitize and validate all user inputs to protect against SQL injection and other malicious attacks.
- **Secure Communication:**
  - Use HTTPS to encrypt data transmitted between the user's browser and the server.

Summary

- A robust and well-designed database is critical for the success of Spiceta Naturals, ensuring efficient management of products, users, and transactions.
- Implementing best practices in database design, security, and optimization will provide a seamless and secure shopping experience for customers.

- **Final Thoughts:**
  - Continuous evaluation and enhancement of the database will support the growing needs of Spiceta Naturals, enabling scalability and adaptability in a competitive e - commerce landscape.

Conclusion:

The spice-selling website developed as part of this project illustrates the effectiveness of integrating a relational database system like **MySQL** with a modern technology stack, such as **Node.js** and **JavaScript**, to create a dynamic and secure e-commerce platform. The adoption of a three-tier architecture, which clearly separates the front-end, application, and database layers, has allowed for the development of a modular system that is not only robust but also easy to maintain and expand.

Through this project, we successfully achieved the key objectives, delivering a user-friendly interface for seamless product browsing and purchasing, while ensuring the secure handling of sensitive data within the back-end database. The implementation of Node.js for application logic, along with MySQL for database management, has demonstrated a powerful and scalable solution suitable for the complexities of an e-commerce environment.

Overall, the project showcases how modern web development techniques and relational database systems can come together to provide a smooth, secure, and efficient experience for both customers and administrators. Future improvements, such as the addition of advanced security features, enhanced scalability, and improved user experience, can further elevate the platform, ensuring its long-term success in the e-commerce market.