Leo Cafe Website

Welcome LEO CAFE

Spices For Various Tastes

Discover The Genuine Charm of LEO CAFE. Indulge In the Delectable Flavors of LEO CAFE.

1. **Database Setup:**
   - Set up a database (e.g., MySQL).
   - Create a database and configure the connection details in the appropriate configuration file (if applicable).
2. **Import Database Schema:** If your project includes a database schema or structure, provide a script or instructions on how to import it into the database.
3. **Environment Variables:**
   - Create a **.env** file in the project root.
   - Define necessary environment variables such as database connection details, API keys, etc.
4. **Run the Application:**

   ```
   npm start
   ```
5. **Access the Website:** Open your web browser and navigate to **http://localhost/hansika**/

Database Setup Details

Provide specific information on setting up the database:

- **Database Type:** Specify the type of database used (e.g., MySQL).
- **Database Name:** The name of the database your application will use.
- **Connection Details:** Include details such as host, port, username, and password.
- **Schema Import Script:** If applicable, provide a script or instructions for importing the database schema.

Additional Notes

Include any additional information or specific steps that might be important for running the project successfully.

Sample. env File

```
DB_HOST=”localhost”

DB_USER=root

 DB_PASSWORD=””

DB_DATABASE=” restaurant website”
```

Features

- Homepage**: A visually appealing homepage showcasing featured items.

- Product Catalog**: Detailed catalog with sorting, filtering, and dynamic updates.

- Shopping Cart**: Sleek cart design with dynamic updates for customization.

-Backend (PHP and MySQL) **: Manages product data, user carts, and order history.

- Checkout Process**: Streamlined multi-step process with address and payment details.

- User Authentication**: Secure login and registration system with user tracking.

- Online Ordering and Delivery**: Integration with external services for delivery functionalities.

- Security Measures**: Password hashing, input sanitization, and secure data

transmission.

-Admin Panel**: Elevate control with an intuitive admin panel that allows monitoring of all activities on the webpage. Admins can effortlessly add new products, ensuring the platform stays up-to-date with the latest culinary offerings

Technologies Used

- HTML

- CSS

- JavaScript

- PHP

- MySQL

Setup and Installation

1. Navigate to the project directory: `cd trendy-restaurant-ecommerce`

2. Set up a local development server (e.g., using XAMPP, WAMP, or a similar tool).

3. Open the project in your preferred code editor.

4. Configure database settings in the backend PHP files.

Usage

1. Start your local development server.

2. Open the project in a web browser.

3. Explore the platform, add items to the cart, and proceed through the checkout process.

Database Setup

1. Create a MySQL database for the project.

2. Import the provided SQL script to set up the necessary tables and initial data.

Security Measures

- Password hashing is implemented for user authentication.

- Inputs are sanitized to prevent SQL injection attacks.

- Prepared statements are used for interacting with the database.

- Sensitive data transmission is secured, especially during the checkout process.