🛒 Full-Stack E-Commerce Website Development Plan

Objective:
Develop a full-featured e-commerce website inspired by Enza Home Pakistan, encompassing product listings, category navigation, shopping cart, and user authentication.

Product Categories:

- Bed
- Chairs
- Tables
- Sofas
- Dining
- Carpets
- Built-in Kitchen
- Wooden Doors
- Main Gate
- Paintings
- Accessories
- Others

📚 Step 1: Learn the Necessary Technologies

Frontend:

- **HTML & CSS:** Structure and style your web pages.
- **JavaScript:** Add interactivity to your website.
- **React.js:** Build dynamic user interfaces.
- **Tailwind CSS:** Utilize utility-first CSS framework for rapid UI development.

Backend:

- **Node.js:** JavaScript runtime for server-side development.
- **Express.js:** Web framework for Node.js to build APIs.
- **MongoDB:** NoSQL database to store product and user data.
- **Mongoose:** ODM (Object Data Modeling) library for MongoDB and Node.js.

Additional Tools:

- **Git & GitHub:** Version control and code hosting.
- **Postman:** API testing tool.
- **VS Code:** Code editor.

Estimated Learning Time: 4–6 weeks with consistent daily practice.

🧱 Step 2: Set Up the Development Environment

1. **Install Node.js and npm:** Download and install from Node.js official website.
2. **Install MongoDB:** Follow instructions from MongoDB official website.
3. **Set Up Git:** Install Git from Git official website and create a GitHub account.

4. **Install VS Code:** Download from VS Code official website.

🖥️ Step 3: Design the Database Schema

Design your MongoDB collections to reflect the product categories and user information.

Collections:

- **Users:**
    - username
    - email
    - passwordHash
    - role (e.g., admin, customer)
- **Products:**
    - name
    - description
    - price
    - category (e.g., Bed, Chairs)
    - images (array of image URLs)
    - stockQuantity
- **Categories:**
    - name
    - description

- **Orders:**
    - userId
    - products (array of product references and quantities)
    - totalPrice
    - orderDate
    - status (e.g., pending, shipped)

🎨 Step 4: Develop the Frontend

Initialize React App:

bash
CopyEdit
npx create-react-app enza-home-clone
cd enza-home-clone

1.
2. **Install Tailwind CSS:**
   Follow the Tailwind CSS installation guide.

3. **Create Components:**
    - **Header & Navigation:** Include links to all categories.
    - **Home Page:** Showcase featured products and promotions.
    - **Product Listing Page:** Display products by category.
    - **Product Detail Page:** Show detailed information, images, and add-to-cart button.
    - **Shopping Cart:** List selected products with quantities and total price.

- **User Authentication Pages:** Login and registration forms.
- **Admin Dashboard:** Interface for managing products and orders.

4. **Implement Routing:**
   Use react-router-dom to handle navigation between pages.

🧮 Step 5: Develop the Backend

Initialize Node.js Project:

bash
CopyEdit
mkdir enza-home-backend
cd enza-home-backend
npm init -y

1.

Install Dependencies:

bash
CopyEdit
npm install express mongoose bcryptjs jsonwebtoken cors

2.
3. **Set Up Express Server:**
   Create server.js and define routes for:
   - **User Authentication:** Register, login, and logout.
   - **Product Management:** CRUD operations for products.
   - **Order Processing:** Create and manage orders.
4. **Connect to MongoDB:**
   Use Mongoose to define schemas and connect to your MongoDB database.

🔄 Step 6: Integrate Frontend and Backend

1. **Enable CORS:** Allow your frontend to communicate with the backend.
2. **API Integration:** Use fetch or axios in React to call backend APIs.
3. **State Management:** Utilize React's useState and useEffect hooks to manage application state.

🧪 Step 7: Testing and Debugging

1. **Frontend Testing:**
    - Ensure all components render correctly.
    - Test navigation and user interactions.
2. **Backend Testing:**
    - Use Postman to test API endpoints.
    - Validate data being stored and retrieved from MongoDB.
3. **End-to-End Testing:**
    - Simulate user workflows, such as browsing products, adding to cart, and placing orders.

🚀 Step 8: Deployment

1. **Frontend Deployment:**
    - Use Netlify or Vercel to deploy your React app.

2. **Backend Deployment:**
    - Deploy your Express server using Heroku or Render.
3. **Database Hosting:**
    - Use MongoDB Atlas for a cloud-hosted MongoDB instance.

🗓️ Suggested Timeline

Week	Goals
1-2	Learn HTML, CSS, JavaScript basics
3	Learn React and Tailwind CSS
4	Learn Node.js, Express, and MongoDB
5	Design database schema and set up backend
6	Develop frontend components
7	Integrate frontend with backend
8	Testing and debugging
9	Deployment and final touches

📚 Additional Resources

- **React Documentation:** https://reactjs.org/docs/getting-started.html
- **Express Documentation:** https://expressjs.com/
- **MongoDB Documentation:** https://docs.mongodb.com/
- **Tailwind CSS Documentation:** https://tailwindcss.com/docs