today we have learned these javascript topics.(asynhcronous, promises, recursion among others). we were given a project. an ecommerse like. i will use json server locally to act my database. theres is an admin and a normal shoper. Admin can login with hardcorded credentials, (email and password), add shopping products, update, delete products. a normal user has to sign up with, can enter full names, email address and password or the can use their google account to signup.. on login in the can view the products, add to cart, update or remove from cart. calculate the total items and on checkout they can show the message. if for instance the chose an item and logout without checking out, on loggin again they can still find the item still added to cart. validate all the forms and provide a nice UI..

This is an excellent project to consolidate your knowledge of asynchronous JavaScript, promises, and recursion. Here's how you can structure it step by step:

**1. Setup JSON Server**

* Create a db.json file to act as your database, with collections for products, users, and carts.
* Use json-server to serve the file locally:

bash

CopyEdit

npx json-server --watch db.json --port 3000

**2. Plan Your Data Structure**

* **Users**:

json

CopyEdit

{

"id": 1,

"name": "John Doe",

"email": "john@example.com",

"password": "hashedPassword",

"role": "admin" // or "shopper"

}

* **Products**:

json

CopyEdit

{

"id": 1,

"name": "Product Name",

"description": "Product description",

"price": 100,

"image": "imageURL",

"stock": 10

}

* **Cart**:

json

CopyEdit

{

"userId": 1,

"items": [

{ "productId": 1, "quantity": 2 }

]

}

**3. Implement Features**

**Admin Features**

* **Login**: Verify hardcoded credentials.
* **CRUD for Products**: Use fetch() with methods (POST, PUT, DELETE).

**User Features**

* **Signup/Login**: Save user data to the database. Use Google OAuth for external signup (you can simulate it locally).
* **Cart Operations**: Fetch cart data by user ID, manage items, and persist changes.
* **Session Management**: Use localStorage to remember cart items between logins.

**Checkout Flow**

* Calculate total price and show a success message on checkout.

**4. Validation and UI**

* Use regular expressions for form validation (e.g., email format, password strength).
* Create a responsive UI with frameworks like Bootstrap or Materialize.

**5. Enhance with Asynchronous Operations**

* Use **async/await** for API calls to make your code readable.
* Use **recursion** for complex tasks like traversing product categories if applicable.

Would you like help with any specific part of the implementation?

**You said:**

generate all the fcodes and files required

**ChatGPT said:**

ChatGPT

Ecommerce Platform

Answer in chat instead

I’ve generated the core files and code snippets required for your e-commerce project, including db.json, server.js, index.html, styles.css, and app.js. These cover the setup of the JSON server, fetching and displaying products, and adding items to the cart.

Let me know if you'd like to refine or expand any specific section!