ASD project product Review & Rating Platform

Build a product review and rating platform where users can browse products, post reviews, and rate products

Core Functionality

Products

- Store product data in products.json with fields:
  - o id, name, description, category, price, dateAdded, averageRating (computed from reviews).
    - o You may add/update fields
- Use Node's fs module to store products and reviews in JSON files.

Reviews

- Add, edit, and delete reviews for each product.
- Each review has:
  - o id, productId, author, rating (1–5), comment, and date.
  - o You may add/update field
- Store in reviews.json.

Rating & Filtering

- Compute and display average product ratings from all reviews.
- Filter products by category.
- Search products by name.

API Endpoints

Products

- GET /products: Fetch all products in pages of 10, sorted by dateAdded (desc). Accepts optional page, category query strings.

- GET /products/search?q=: Return products whose names match query.

Reviews

- GET /products/:id/reviews: Fetch all reviews for a product.
- POST /products/:id/reviews: Add a review.
- PUT /products/:productId/reviews/:id: Update a review.
- DELETE /products/:productId/reviews/:id: Delete a review.

Frontend (React + TypeScript)

- Create a well-structured component hierarchy.
- Fetch data from the backend using native fetch or Axios.
- Handle loading and error states appropriately.
- Design a user-friendly interface using Tailwind CSS, Bootstrap or custom CSS.
- State Management: use React Context API, Redux Toolkit, Jotai or Zustand
- React Router v6+
- Type Safety: All components, props, and API responses must be typed
- Performance Optimization: Avoid unnecessary re-renders
- Memoization with React.memo, useMemo, useCallback

Backend (Express + TypeScript)

- Use Express middleware for requests and handling errors.
- The API should return data in only JSON format with appropriate HTTP status codes.

- The API should have good error handling and return appropriate HTTP status codes.
- The API should have good client and server-side validation.
- Node filesystem