

# CS472-Final-Project-May-2025 Product Review & Rating Platform

This is a single person project connecting all parts of the content in the course.

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:
  - id, name, description, category, price, dateAdded, averageRating (computed from reviews).
  - 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:
  - id, productId, author, rating (1–5), comment, and date.
  - You may add/update fields
- 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) Requirements

- 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) Requirements**

- 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

### **Submission:**

- Inside your project, only direct two folders: client, server. I'll only run "npm install" on server side, if it doesn't work, need to debug, lose points. You may provide the readme file.
- Upload your source code (zip) along with GitHub link to the Sakai.
- You will create 15–20-minute video explaining verbally the tasks that you have implemented. Don't forget to submit your video link to Sakai and open permission to access it directly.
- Your project grade is based on the project you present. If you miss some features or some features don't work during presentation, you'll lose points, I won't do another presentation with you.
- Deadline: Thursday May 22 at 11:57 PM.

### **Important Notes:**

- You are not allowed to share codes with others. If detected, you will get NC.
- Remember to respect the code honor submission policy. All written code must be original. Presenting something as one's own work when it came from another source is plagiarism and is forbidden.
- Plagiarism is a very serious thing in all American academic institutions and is guarded against vigilantly by every professor.

### **Evaluation Criteria**

- Does the application meet all specified requirements?
- Are all features implemented and working correctly?
- Is the code well-organized and easy to understand?
- Are the best practices followed for both React and Express?
- Is the user interface intuitive and visually appealing?
- The proper use of version control ? (including clear and descriptive commit messages that follow best practices)

**Bonus (maximum 3 extra points):**

- Use any database (MySQL, MSSQL, or MongoDB etc. ) [1 point].
- Integrate the backend with ChatGPT APIs or any other AI APIs [1 point].
- Use the swagger for documentation [1 point].
- Deploy the frontend on a service like Netlify or Vercel [1 point].
- Deploy the backend on a service like Heroku, Render, or AWS etc. [1 point].

Good luck, and happy coding!