

Project B Checklist: FW Webb Product Catalog

Objective

Build a product catalog web application with features for browsing, filtering, and viewing product details.

☒ Frontend (Next.js)

1. ☐ Set Up Pages

- ☐ / : Product listing page.
- ☐ /product/[id] : Product details page.
- ☐ /contact : Contact form.

2. ☐ Install React Hooks

- ☐ Install any necessary React state management libraries.

```
npm install
```

3. ☐ Fetch Data

- ☐ Integrate REST API endpoints to fetch product data.
 - ☐ Add functionality for paginated product grids, filters, and search bars.
-

☒ Backend (Node.js with Express)

1. ☐ Create Routes

- ☒ /api/products : Fetch product list.
- ☐ /api/products/:id : Fetch product details.
- ☐ /api/contact : Handle contact form submissions.

2. ☐ Integrate PostgreSQL Database

- ☒ Set up database connection on Railway.
 - ☐ Update API routes to fetch data from the database.
 - ☐ Use Prisma ORM for schema and queries.
-

☒ Database (PostgreSQL)

1. ☐ Define Tables

- ☒ products : id, name, description, price.
 - ☐ Add category, brand, image_url, video_url.
- ☐ categories : id, name.
- ☐ brands : id, name.

2. ☐ Seed Initial Data

- ☐ Add sample product data to the database.
 - ☐ Populate categories and brands.
-

☒ Testing

1. ☐ Write Integration Tests:

- ☐ /api/products endpoint.
- ☐ /api/products/:id endpoint.
- ☐ /api/contact endpoint.

2. ☐ Verify frontend for:

- ☐ Correct data display.
 - ☐ Mobile responsiveness.
-

📦 Deployment

1. ☒ **Frontend:** Deployed to Netlify.
 2. ☒ **Backend:** Deployed to Railway.
 3. ☐ Verify production functionality:
 - ☐ `/api/products` fetches product list.
 - ☐ `/api/products/:id` fetches product details.
 - ☐ `/api/contact` submits forms successfully.
-

Next Steps

- Test functionality end-to-end.
- Ensure CORS configuration works for Netlify → Railway integration.