Diligent - E-Commerce Website
Technical Architecture Document

Project overview

A lightweight MERN stack e-commerce website with:
- React frontend (Tailwind + Material UI)
- Express.js backend (REST API)
- MongoDB Atlas cloud database
- Basic cart state management (frontend) and sample backend endpoints

Core requirements

- Browse products
- View product details
- Add / remove items to cart
- Lightweight backend to serve product data and cart endpoints (stateless)
- Responsive, styled UI using Tailwind + Material UI components

High-level architecture

```
[Browser]
  -> React (SPA) --XHR/Fetch--> Express API
  -> React localStorage for cart persistence (simple state + persistence)
```

Backend

- Node.js + Express
- Endpoints:

```
  GET  /api/products            -> list of products (supports pagination & search)
  GET  /api/products/:id        -> product details
  POST /api/cart/checkout       -> mock checkout endpoint
  GET  /api/health              -> health check
```

- MongoDB Atlas used for product storage. Connection string stored in .env as MONGO_URI.

Frontend

- React app bootstrapped with a simple structure:

```
  /src
    /components (ProductList, ProductCard, ProductDetails, Cart)
    /pages (Home, ProductPage, CartPage)
```

  Uses Tailwind for layout; Material UI for a couple of interactive components (AppBar, Buttons, Icons)

State management

- Local state via React useState/useContext for cart
- Cart persisted to localStorage for session survival
- Backend is stateless for carts (checkout endpoint accepts cart payload)

Security & deployment notes

- Never commit real MongoDB credentials. Use .env and environment variables on deployment.
- For deployment: host backend on Heroku / Render / Railway; frontend on Vercel/Netlify; use MongoDB Atlas.
- Enable CORS on backend for the frontend origin.
- Add rate-limiting and input validation for production readiness.

Deliverables

1) Codebase: frontend + backend, runnable locally
2) Architecture PDF (this doc)
3) Prompts log used to generate code & docs

Setup (local dev)

1. Backend: