Problem Statement: Product Catalog Management System

Objective

Develop a full-stack MERN (MongoDB, Express, React, Node.js) application that enables users to view a catalog of 100 products stored in a MongoDB database. The frontend must display the products in a paginated table using Material UI. The pagination should be applied only on the frontend, and all data should be fetched at once from the backend.

Functional Requirements

1. Backend API (Express + MongoDB)

- Create a GET endpoint to fetch all product entries in one request (no backend pagination).

- Use MongoDB to store at least 100 product entries, each with the following fields:

  - name: String

  - price: Number

  - category: String

  - inStock: Boolean

2. Data Seeding

- Create a script to generate and insert 100 sample products into the MongoDB database.

3. Frontend UI (React + Material UI)

- Build a user-friendly table interface to display product data.

- Fetch all product data from the backend using Axios.

- Apply pagination logic entirely on the frontend using Material UI.

- The table must include the following header columns:

Problem Statement: Product Catalog Management System

- Name
- Price
- Category
- In Stock

Technical Constraints

- Use Mongoose for modeling MongoDB schema.
- Use Axios for API calls in React.
- Store environment variables like DB URI in a .env file.
- Use create-react-app for initializing the React project.

Deliverables

- A working backend server (server.js) with product route that returns all products.
- A MongoDB collection named 'products' with 100 seeded entries.
- A React application with a paginated table of products (pagination handled on the frontend).
- Integration of Material UI for responsive table design and pagination.
- A GitHub repository or deployable ZIP folder of the full project.