Inventory Management Project

📌 Overview:

This project has a FastAPI backend running on port **8000** and a React frontend running on port **3000**. Since both run on different servers, the frontend communicates with the backend through API endpoints like /products, and these endpoint names must match exactly on both sides. To allow the frontend to send requests to the backend, CORS is enabled in FastAPI; otherwise, the browser blocks communication between different ports. The backend uses PostgreSQL as the local database, and SQLAlchemy ORM converts Python classes into SQL tables, making database operations easy. The frontend sends CRUD requests, FastAPI handles them, and PostgreSQL stores the data. Overall, this is a simple Inventory Management System where React handles the UI, FastAPI handles APIs, and PostgreSQL stores all product records.

📌 Details:

This is a full-stack **Inventory Management System** built using:

- **FastAPI** (Backend)
- **PostgreSQL** (Database)
- **React.js** (Frontend)

Backend runs on **http://localhost:8000** and Frontend runs on **http://localhost:3000**. CORS is enabled so that both servers can communicate.

📌 Project Structure

```
📁project/
│
├─── backend/
│   ├─── main.py # FastAPI main file
│   ├─── database.py # DB engine + session configuration
│   ├─── DB_ORM_Model.py # ORM table mapping using SQLAlchemy
│   ├─── models.py # Pydantic models for request/response
│   └─── venv/ # Virtual environment (after creation)
```

```
│
└──── frontend/
├──── src/ # React source code
├──── package.json
└──── node_modules/ # Created after npm install
```

📌 Backend Setup (FastAPI + PostgreSQL)

1️⃣ Create Virtual Environment

Open terminal inside **backend** folder:

CODE : python -m venv venv
Activate it:

- Windows: venv\Scripts\activate

2️⃣ Install Backend Dependencies

After activating venv:

CODE : pip install fastapi uvicorn sqlalchemy psycopg2 pydantic

3️⃣ Setup PostgreSQL Locally

Make a database named : InventoryDB
Check database.py — it already contains the connection string:
“postgresql://postgres:root@localhost:5432/InventoryDB”

(Modify if your PG password is different.)

4️⃣ Start the Backend Server
RUN : uvicorn main:app –reload
Backend will now run at: http://localhost:8000

📌 Frontend Setup (React)

Go to the **frontend** folder:
Install node modules: npm install
Start React App: npm start
Frontend will run at: http://localhost:3000

📌 Connecting Frontend & Backend (CORS)

Since React (3000) and FastAPI (8000) run on different ports, the browser blocks requests. To fix this, CORS is enabled in main.py of baceknd:

“allow_origins=["http://localhost:3000"]”

This allows the frontend to send GET, POST, PUT, DELETE requests to backend. React calls must match backend routes like:
GET /products

GET /products/{id}

POST /products

PUT /products/{id}

DELETE /products/{id}

📌 ORM + Database Explanation

- DB_ORM_Model.py creates SQL tables using **SQLAlchemy ORM**.
- models.py defines request models using **Pydantic**.
- No raw SQL required — ORM converts Python classes → SQL tables.
- main.py reads/writes data using db.query().

📌 How the App Works

1. Start backend (FastAPI)
2. Start frontend (React)
3. React sends API requests to FastAPI
4. FastAPI interacts with PostgreSQL using SQLAlchemy
5. Responses return back to React UI