Voici un **récapitulatif complet et propre** de tout ce que tu as mis en place pour déployer ton projet **MERN (MongoDB, Express, React, Node.js)** localement avec **Docker**, sans docker-compose.

**✅ Objectif :**

Déployer en local une application MERN avec :

* Backend Express/Node
* Base de données MongoDB
* Frontend React
* Docker, sans docker-compose
* Réseau Docker personnalisé

**📁 Structure du projet**

/ecommerce-api/ ← backend Express

/client/ ← frontend React

**⚙️ Étapes réalisées**

**1. 🧱 Backend (Express + MongoDB)**

**🔨 Dockerisation de l'API**

* Création d’un Dockerfile :

FROM node:18

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

EXPOSE 5000

CMD ["npm", "start"]

**🔧 .env configuré avec alias Docker :**

MONGO\_URI=mongodb://mongo:27017/ecommerce

PORT=5000

**🐳 Commandes exécutées :**

docker network create backend-test-network

docker volume create mongo-data

docker run -d --name db-container --network backend-test-network --network-alias mongo -v mongo-data:/data/db mongo:4.4

docker build -t ecommerce-backend ./ecommerce-api

docker run -d --name express-app --network backend-test-network -p 5000:5000 --env-file .env ecommerce-backend

**2. 💻 Frontend React (Vite ou CRA)**

**🚧 Création du dossier :**

npx create-react-app client --template typescript

**🔧 Mise à jour de Message.tsx :**

* Utilisation de VITE\_API\_URL :

fetch(`${import.meta.env.VITE\_API\_URL}/api/products`)

**.env côté React :**

VITE\_API\_URL=http://localhost:5000

PORT=3000

**🐳 Dockerfile dans /client/ :**

FROM node:18 AS build

WORKDIR /app

COPY package\*.json ./

RUN npm install

COPY . .

RUN npm run build

FROM nginx:alpine

COPY --from=build /app/dist /usr/share/nginx/html

EXPOSE 80

**🐳 Commandes :**

docker build -t ecommerce-frontend ./client

docker run -d --name frontend-app --network backend-test-network -p 3000:80 ecommerce-frontend

**3. 📦 Optimisation .dockerignore**

**Dans /ecommerce-api/.dockerignore et /client/.dockerignore :**

node\_modules

.git

.vscode

.DS\_Store

\*.log

build

dist

**✅ Résultat final**

| **Service** | **Adresse** |
| --- | --- |
| Frontend | [http://localhost:3000](http://localhost:3000/) |
| Backend API | [http://localhost:5000](http://localhost:5000/) |
| MongoDB | Connecté en interne via mongo |

Tous les conteneurs sont sur le même réseau Docker backend-test-network, ce qui permet la communication entre eux via leurs noms/alias.

Tu veux que je te prépare un petit script .sh ou .bat pour tout relancer automatiquement ?