**ארכיטקטורה ותכנון פתרון — חלק ראשון**

**תיאור כללי של המערכת:**

המערכת בנויה מ-2 מסכים עיקריים:  
1️⃣ **רשימת קניות** — שליפת קטגוריות ומוצרים מ-SQL Server.  
2️⃣ **סיכום הזמנה** — שמירת נתוני הזמנה במסד NoSQL.

**📂 מבנה הפרויקט (ברמה לוגית):**

bash

CopyEdit

/client -- React + Redux Toolkit

/server-net -- .NET 8 + EF Core + SQL Server (API 1)

/Controllers

/Services

/Entities

/DbContext

/server-node -- Node.js (Express / NestJS) + NoSQL (API 2)

/routes

/controllers

/models

**שלב ראשון: צד לקוח - React + Redux Toolkit**

**📌 תכנון State:**

* קטגוריות: categories: Category[]
* מוצרים: products: Product[]
* סל קניות: cart: CartItem[]
* פרטי הזמנה: orderDetails: { fullName, address, email }

**📌 מבנה המסכים:**

1. **ShoppingPage.jsx**
   * שליפת קטגוריות ומוצרים מה-API של .NET
   * סינון מוצרים לפי קטגוריה
   * שינוי כמות
   * כפתור "הוסף לסל"
2. **OrderSummaryPage.jsx**
   * הצגת הסל
   * טופס הזמנה (שם, כתובת, מייל)
   * כפתור "אשר הזמנה" => קריאה ל-API Node.js

**📌 Redux Toolkit:**

* Slice לקטגוריות ומוצרים
* Slice לסל קניות
* Slice להזמנה
* AsyncThunk לשליפות ושמירות

**שלב שני: צד שרת ראשון - .NET 8 + EF Core + SQL Server**

**📌 מבנה בסיס הנתונים (SQL Server):**

**טבלה: Categories**  
| Id | Name |

**טבלה: Products**  
| Id | Name | CategoryId (FK) | Price |

**📌 Endpoints:**

bash

CopyEdit

GET /api/categories => מחזיר את כל הקטגוריות

GET /api/products => מחזיר את כל המוצרים (או לפי קטגוריה עם Query param)

**📌 מחלקות עיקריות:**

* Entities (Category, Product)
* DbContext
* Repositories / Services
* Controllers

**שלב שלישי: צד שרת שני - Node.js / NestJS + MongoDB או Elasticsearch**

**✅ אני ממליץ MongoDB. Elasticsearch מיותר פה.**

**MongoDB Collection: Orders**

json

CopyEdit

{

"fullName": "User Name",

"address": "Full Address",

"email": "email@example.com",

"items": [

{ "productId": 1, "productName": "Milk", "quantity": 2 }

],

"createdAt": ISODate()

}

**📌 Endpoints:**

bash

CopyEdit

POST /api/orders => שומר את ההזמנה במסד הנתונים

**שלב רביעי: חיבורים בין חלקים**

| **צד לקוח** | **שרת .NET 8** | **שרת Node.js** |
| --- | --- | --- |
| שליפת מוצרים | GET /api/categories |  |
|  | GET /api/products |  |
| שליחת הזמנה |  | POST /api/orders |

**דוגמה לפלואו:**

1️⃣ לקוח עולה => נשלפים קטגוריות ומוצרים מהשרת .NET.  
2️⃣ המשתמש בוחר מוצרים => נוספים לסל ב-Redux.  
3️⃣ המשתמש ממלא פרטי טופס ולוחץ אשר => POST אל Node.js ששומר ב-MongoDB.