

Got it 

You want **simple, clean, beginner-friendly architecture — no code, only file structure + pages + database schema**, for a **clothing e-commerce site** using:

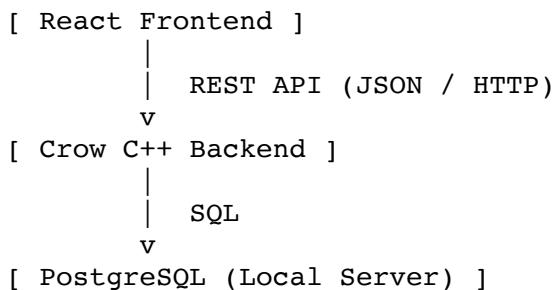
- **Backend:** C++ (Crow)
- **Frontend:** React
- **Database:** PostgreSQL (local server)
- **Features:** Home, Men, Women, Cart (+ recommended pages)

Let's keep this **clear and practical**, like something you'd actually build.



1

Overall Simple Architecture



2

Backend Structure (C++ + Crow)



backend/

```
backend/
├── CMakeLists.txt
├── main.cpp
└── config/
    └── db_config.json
db/
├── connection.h
└── connection.cpp
routes/
├── home_routes.cpp
├── product_routes.cpp
├── cart_routes.cpp
└── order_routes.cpp
models/
├── User.h
├── Product.h
└── Category.h
Cart.h
Order.h
utils/
└── response_helper.h
```

◆ Purpose

- routes/ → API endpoints
 - models/ → data structures
 - db/ → PostgreSQL connection
 - config/ → local DB credentials
-



3

Frontend Structure (React)



frontend/

```
frontend/
└── public/
    └── index.html
└── src/
    ├── api/
    │   └── api.js
    ├── components/
    │   ├── Navbar.jsx
    │   ├── Footer.jsx
    │   ├── ProductCard.jsx
    │   └── CartItem.jsx
    ├── pages/
    │   ├── Home.jsx
    │   ├── Men.jsx
    │   ├── Women.jsx
    │   ├── Cart.jsx
    │   ├── ProductDetails.jsx
    │   └── Checkout.jsx
    ├── context/
    │   └── CartContext.jsx
    └── App.jsx
    └── main.jsx
```



4

Website Pages (Simple & Recommended)

Page	Purpose
Home	Featured products, categories
Men	Men clothing items
Women	Women clothing items
Product Details	Size, color, add to cart
Cart	Selected products
Checkout	Place order

Optional later:

- Login / Register
 - Admin dashboard
-



6

How Everything Connects (Simple Flow)

React Page (Men / Women / Cart)

↓

API request (Axios)

↓

Crow Route (C++)

↓

PostgreSQL Query

↓

JSON Response

↓

React UI Update
