Excellent — a **clothing eCommerce website** with **Next.js**, **MongoDB**, **Cloudinary**, **User/Admin roles**, **cash on delivery**, with **categories like kids, men, women, winter, summer**, **special offers** — let me provide you with a **complete, clean project structure**, **database schema**, and **Next.js folder setup** for **JavaScript** (not TypeScript).

Here’s **professional architecture** for you:

**✅ High-Level Features**

| **Feature** | **Description** |
| --- | --- |
| **User** | View products, filter by category/season, cart, purchase (COD) |
| **Admin** | Add/edit/delete products, manage categories/offers/orders |
| **Authentication** | JWT-based Auth (Login/Register) |
| **Image Upload** | Via **Cloudinary** |
| **Database** | **MongoDB** (Products, Orders, Users, Categories, Offers) |
| **Payments** | Cash on Delivery (No online payment for now) |

**📁 Recommended File Structure (Next.js App Directory)**

bash

CopyEdit

ecommerce-clothing/

├── app/

│ ├── layout.js

│ ├── page.js # Home Page

│ ├── products/ # All product-related routes

│ ├── cart/ # Cart page

│ ├── login/ # Login page

│ ├── register/ # Register page

│ ├── admin/ # Admin dashboard, protected route

│

├── components/

│ ├── Navbar.js

│ ├── Footer.js

│ ├── ProductCard.js

│ ├── CategoryFilter.js

│ ├── SpecialOfferBanner.js

│ └── AdminSidebar.js

│

├── context/

│ ├── AuthContext.js

│ └── CartContext.js

│

├── lib/

│ ├── dbConnect.js # MongoDB connection

│ └── cloudinary.js # Cloudinary config

│

├── models/

│ ├── User.js

│ ├── Product.js

│ ├── Order.js

│ ├── Category.js

│ └── Offer.js

│

├── pages/api/

│ ├── auth/ # Login, register, etc.

│ ├── products/ # CRUD for products

│ ├── orders/ # Place/view orders

│ └── upload/ # Cloudinary upload endpoint

│

├── public/

│

├── styles/

│ ├── globals.css

│ └── variables.css # Theme/Global variables

│

├── utils/

│ ├── authMiddleware.js # Protect admin routes

│ └── generateToken.js # JWT functions

│

├── .env.local # Environment variables (Mongo URI, Cloudinary)

└── package.json