**PROJECT CONTROL DOCUMENT**

1. **REQUIREMENTS**

Our node.js Website provides

1. Customer should be able to sign up for our website
2. Customer should be able to login to the cloth store website
   1. Users should be able to see the products/clothes even if they don't log in
   2. But when they log in, user details and the cart details should be updated
   3. When a user is not logged in and tries to add something to the cart, then the user should be prompted to log in.
   4. The temporary item which the user selected to add to his/her cart before the login prompt should be automatically added to the cart, once the user logs in
3. User who logged in as an admin should be able to add products to the Database
   1. Use a form to fill Brand, color, attach multiple images, offer applicable, real cost etc. and submit for each cloth.
4. Customer sees all clothes available in the store as a list
5. Customer can filter the products on clicking on the docked tab on left of the page (E.g.: Men, Women may be main option and pyjamas, night wear, party wear might be the sub category)
6. Each product should be displayed as a card with the following properties
   1. 1-4 images (slides on hover)
   2. Seller
   3. Description of the product in one line
   4. Real Price
   5. Offer
   6. Net Price

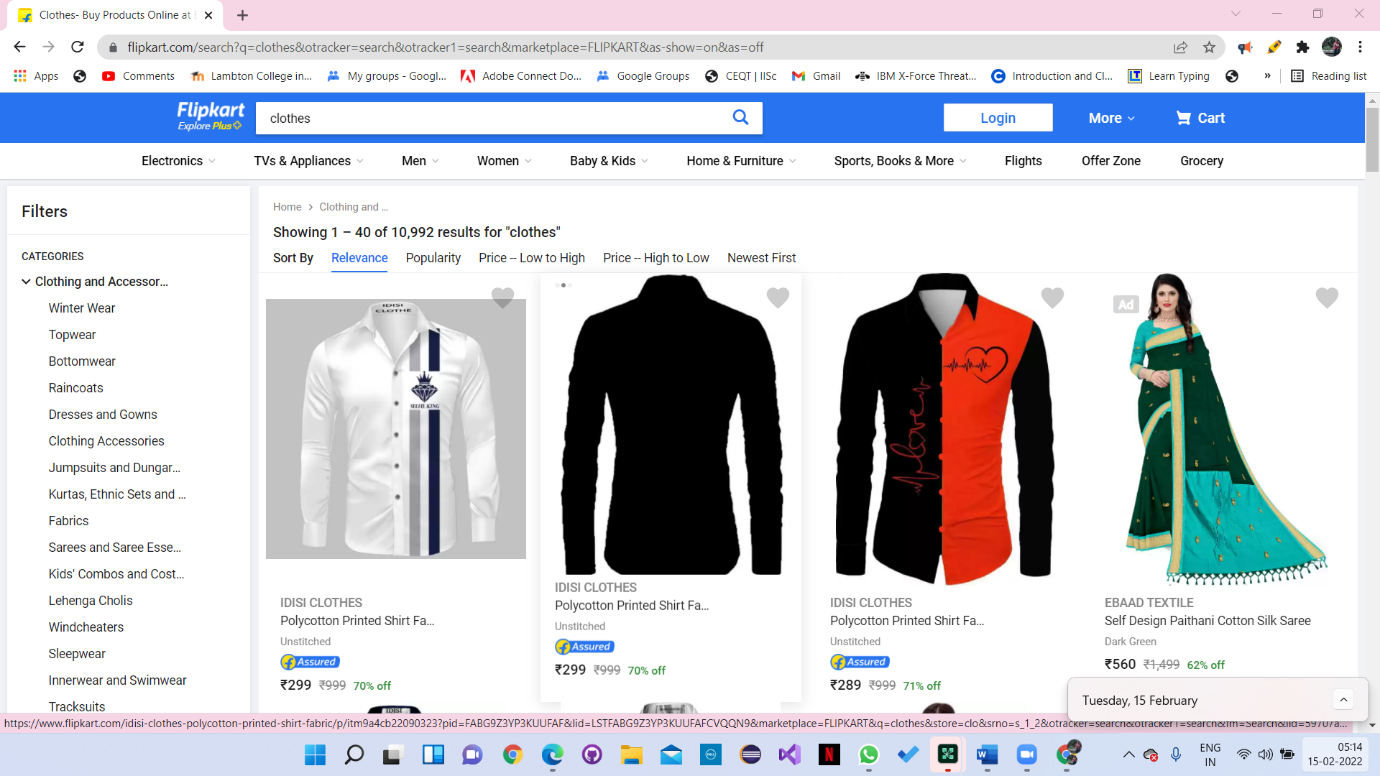
**Hidden details (by which we will filter/display the list in home page)**

* 1. Gender
  2. Type of clothing

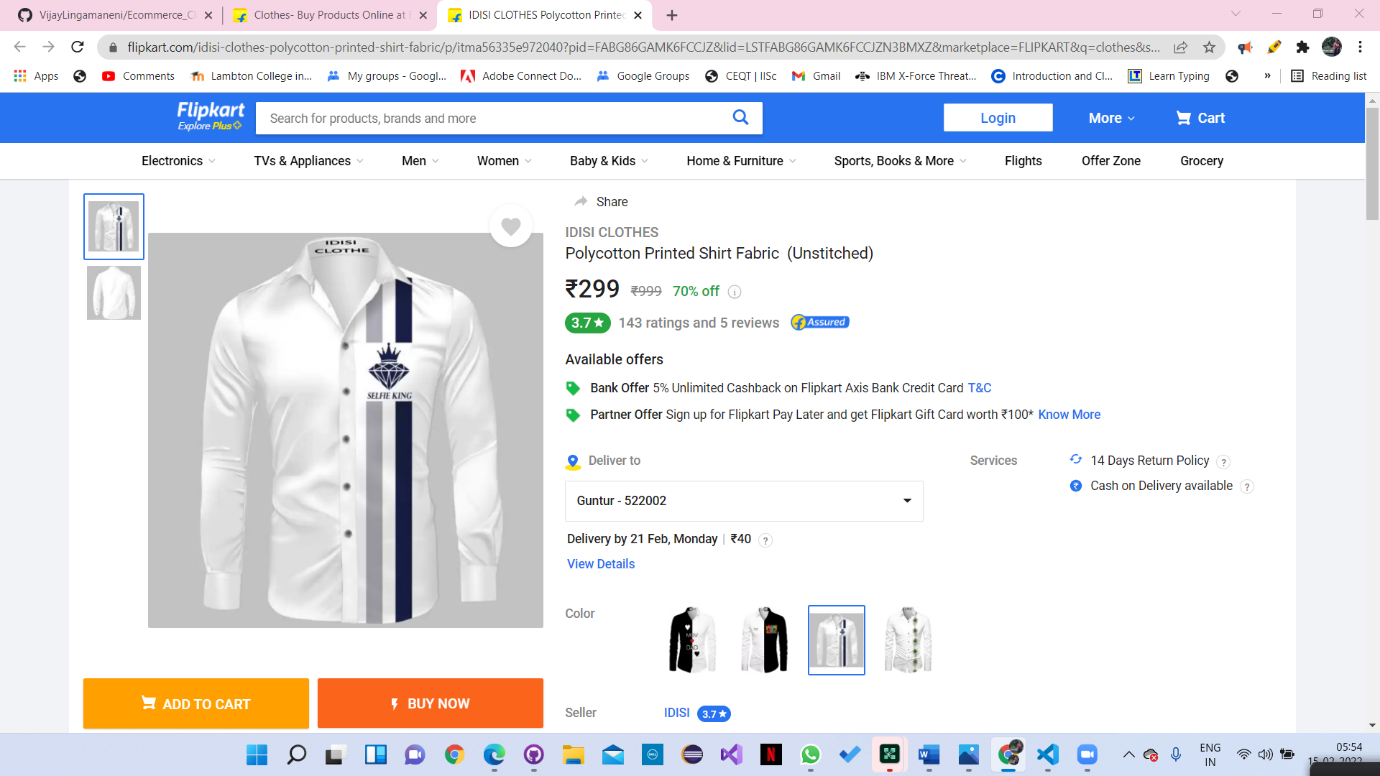
1. On clicking on each product, the customer should be able to the details of the product
   1. Brand
   2. Color
   3. Fabric
   4. Price
   5. Cart Symbol
   6. Should be able to zoom an image
   7. Seller etc.
2. Customer should be able to increase and decrease the quantity(q) of each product and the net price displayed should be increased
3. Customer can add one or more products into their basket
4. Customer should be able to edit the quantity of the products in the basket and the price should be updated accordingly
5. Customer should be able to see no of items on top of the cart if there are any products in the cart
6. Customer should be able to retrieve the products in their basket on clicking on the cart symbol on the top right corner
7. Customer should be able to delete the products in their basket
8. Total amount of the cart should be displayed/updated each time customer adds or deletes a product from it
9. Customer should be able to log out of the website

**INSPIRATIONS OF THE UI DESIGN:**

**Home Page after logging in:**

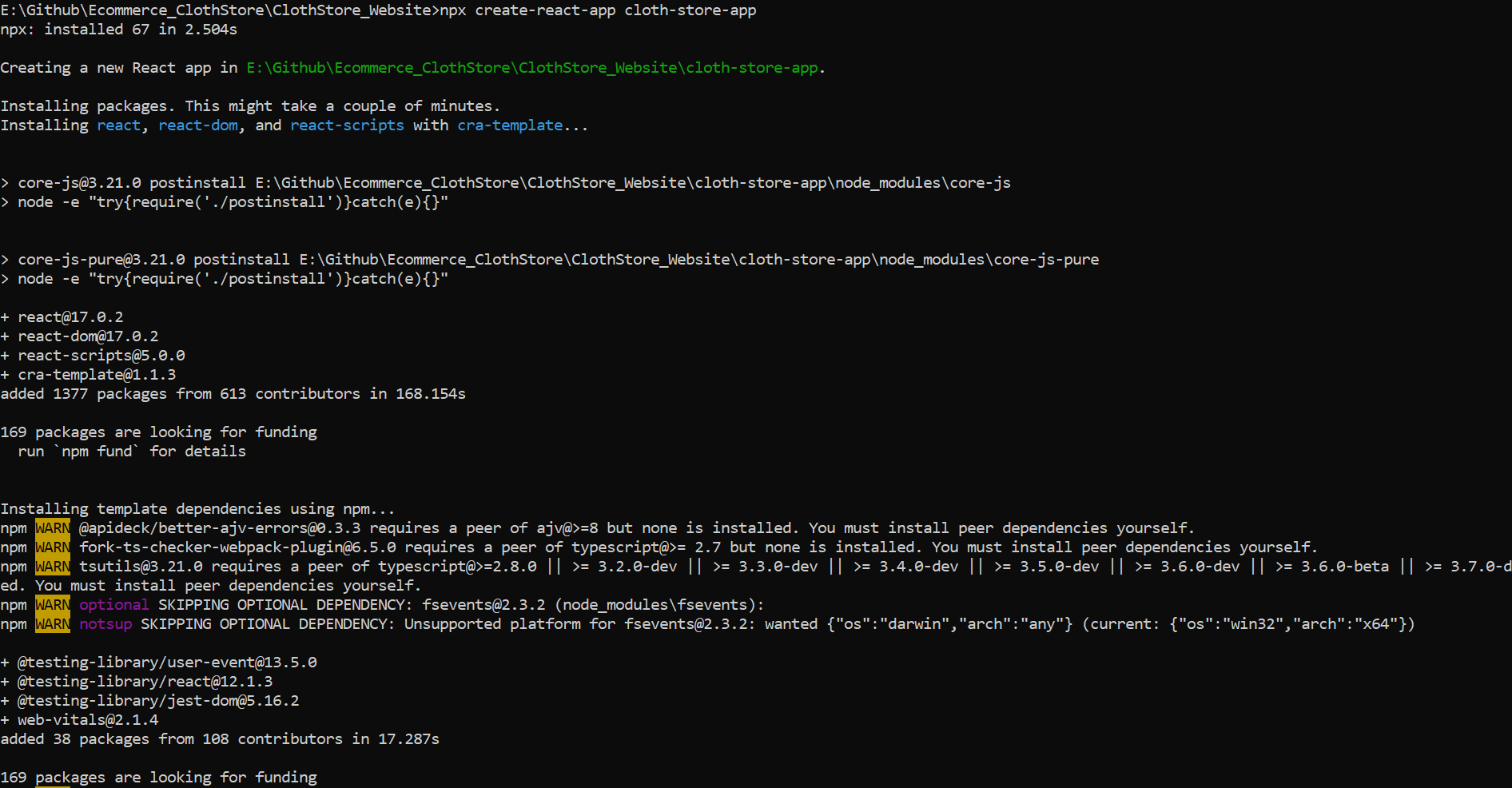
****

**Detail Page of a selected item**

****

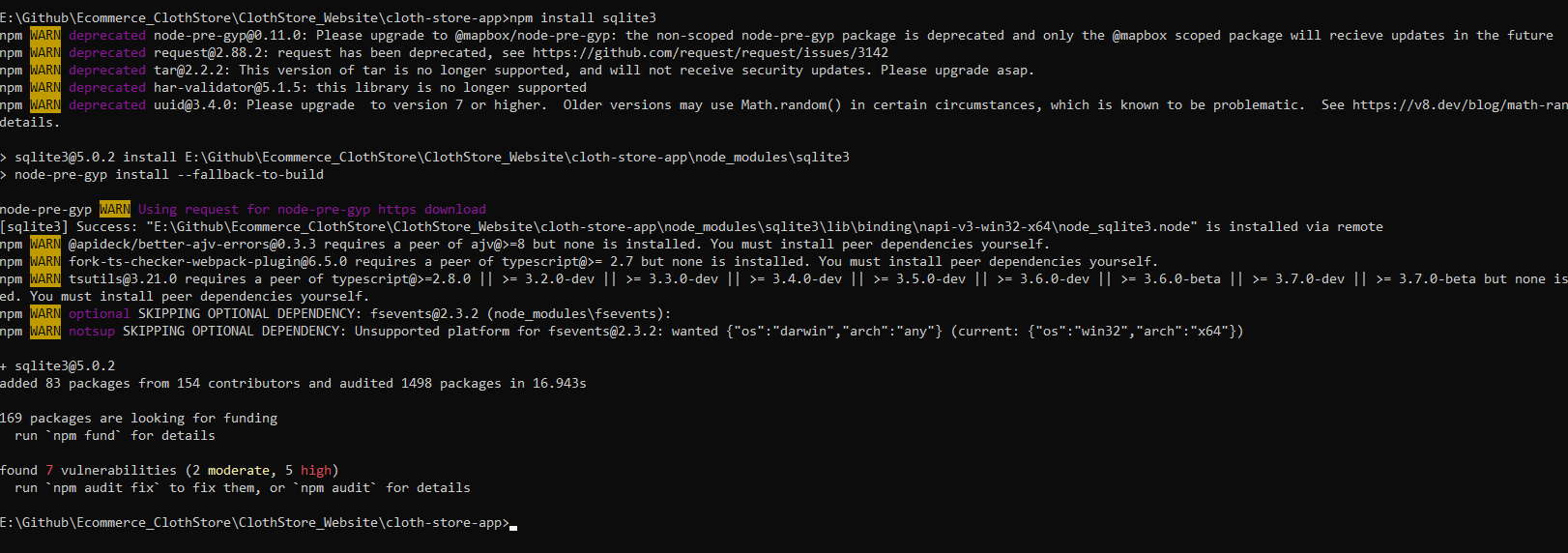
1. **SETUP SQLITE IN LOCAL DEV**

**Create a New react app:** npx create-react-app cloth-store-app

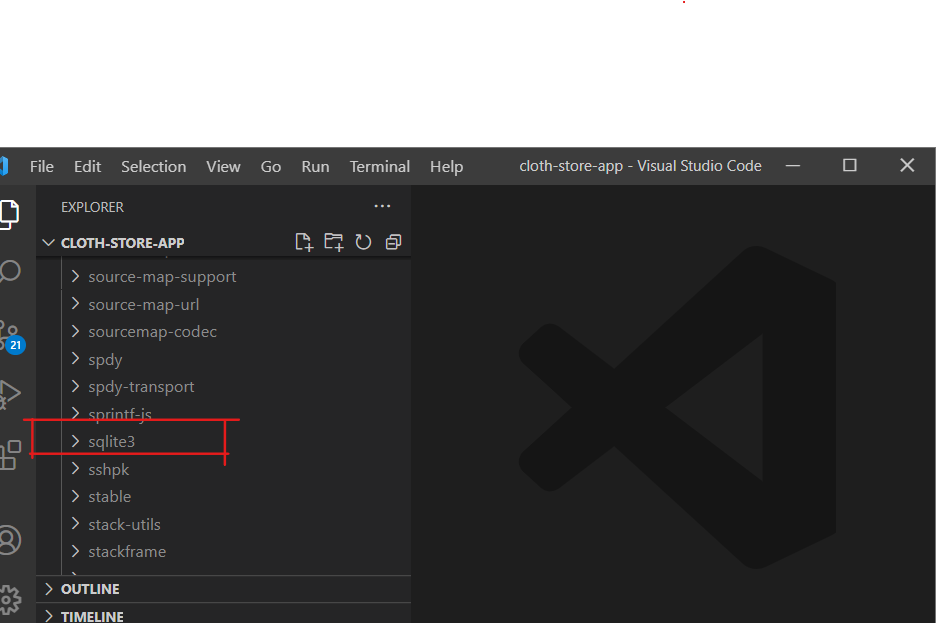


Go to the project directory and execute the following command

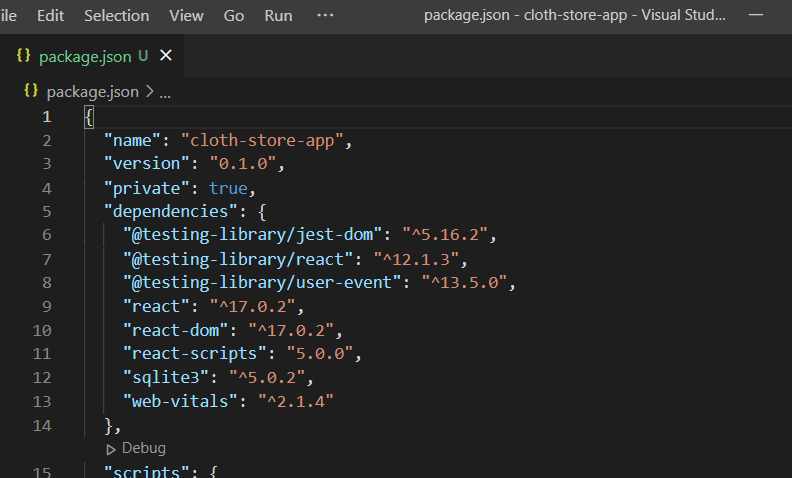
**Command:** npm install sqlite3

****

node\_modules should be updated and you should be able to see sqlite3



You should see package.json also updated accordingly



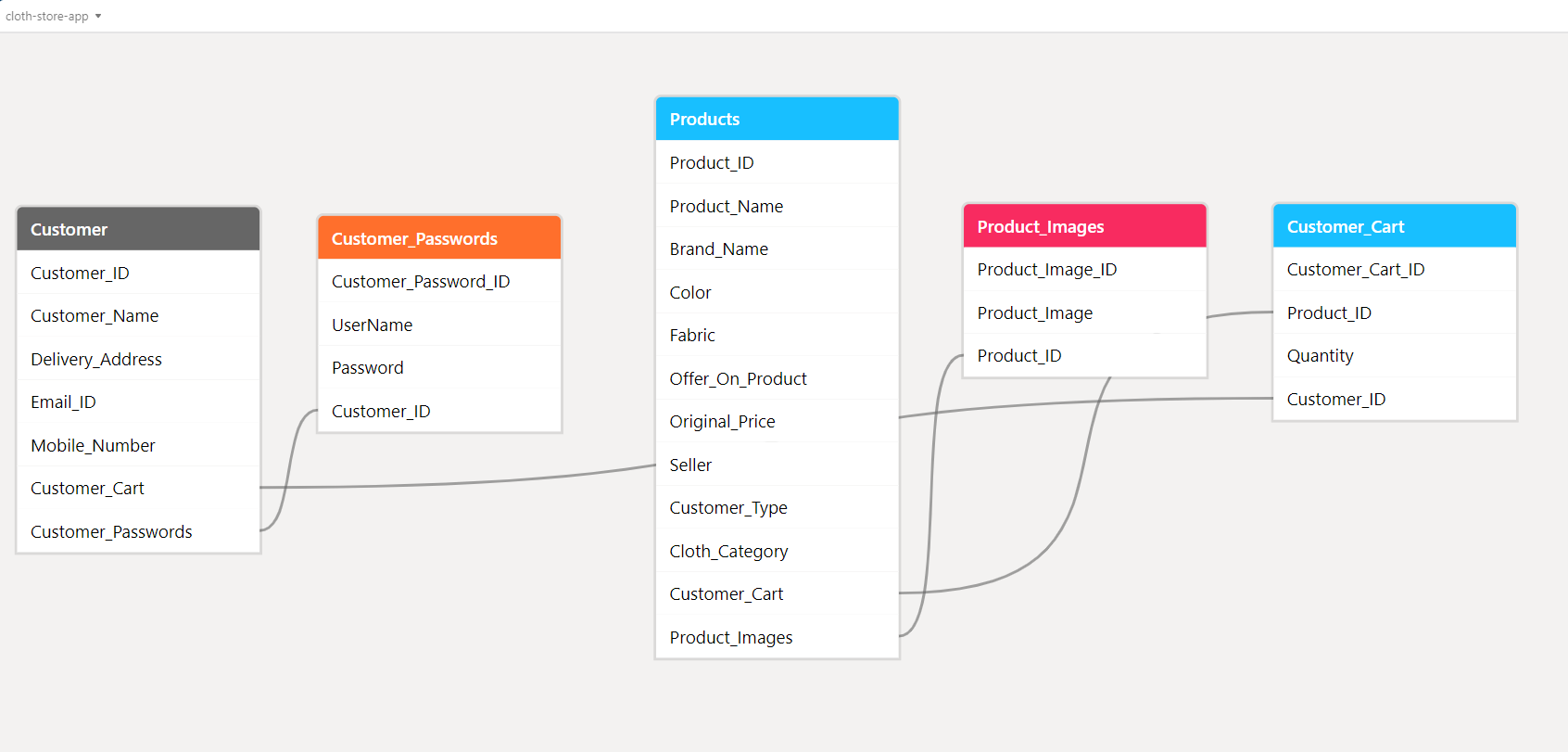
1. **DATA BASE DESIGN**

Database Design is Available in the following link. Anybody Can Sign up with google and view this DB schema and sample data.

**Database Schema:**

[Database Design for ClothStore: Customer - Airtable](https://airtable.com/appZ6uENLYpNwMgb9/tblvlakS7Yioxjy9X/viwCtjXyqEzRZn0Dt?blocks=bliZ55pLb7Xo9saSG)

**For those who can’t access the link:**

****

**Sample Table Data:**

[Database Design for ClothStore: Customer - Airtable](https://airtable.com/appZ6uENLYpNwMgb9/tblvlakS7Yioxjy9X/viwCtjXyqEzRZn0Dt?blocks=bipGGvZ1tTyyTNvQk)

**View Below Excel for sample table data:**



1. **INSTALLING AND SETTING UP DB BROWSER FOR SQLite**

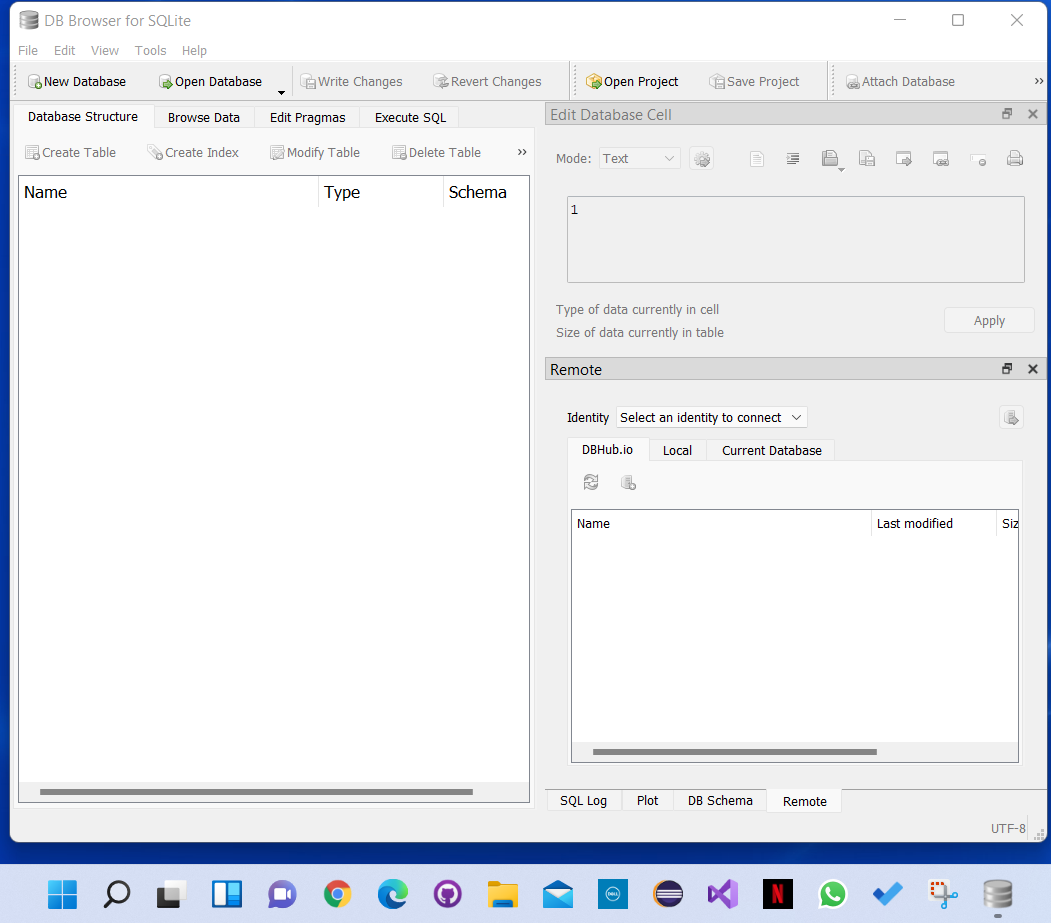
**Installation:**

Download suitable version for your system from the following link.

<https://sqlitebrowser.org/dl/>

**SET UP:**

Open DB Browser for SQLite



1. **Create a new database:**

Clicked on the New Database on the top left and saved it as file in the directory

**E:\Github\Ecommerce\_ClothStore\Database\ClothStore.db**

1. **Create Tables one by one: (By executing the following query)**
2. **Customer Table:**

CREATE TABLE "Customer" (

"Customer\_ID" INTEGER NOT NULL UNIQUE,

"Customer\_Name" TEXT NOT NULL,

"Delivery\_Address" TEXT,

"Email\_ID" NUMERIC NOT NULL,

"Mobile\_Number" TEXT NOT NULL,

PRIMARY KEY ("Customer\_ID" AUTOINCREMENT)

);

1. **Customer\_Password Table**

CREATE TABLE "Customer\_Password" (

"Customer\_Password\_ID" INTEGER NOT NULL UNIQUE,

"Customer\_ID" INTEGER NOT NULL,

"UserName" TEXT NOT NULL UNIQUE,

"Password" TEXT NOT NULL,

FOREIGN KEY("Customer\_ID") REFERENCES "Customer"("Customer\_ID"),

CONSTRAINT "PK\_Customer\_Password" PRIMARY KEY("Customer\_Password\_ID" AUTOINCREMENT)

);

1. **Product Table**

CREATE TABLE "Customer\_Products\_Cart" (

"Customer\_Product\_Cart\_ID" INTEGER NOT NULL UNIQUE,

"Product\_ID" INTEGER NOT NULL,

"Customer\_ID" INTEGER NOT NULL,

"Quantity" NUMERIC NOT NULL,

CONSTRAINT "FK\_Customer\_Product\_Cart\_Customer" FOREIGN KEY("Customer\_ID") REFERENCES "Customer"("Customer\_ID"),

FOREIGN KEY("Product\_ID") REFERENCES "Products"("Product\_ID"),

CONSTRAINT "PK\_Customer\_Cart" PRIMARY KEY("Customer\_Product\_Cart\_ID" AUTOINCREMENT)

);

1. **Product\_Images Table**

CREATE TABLE "Product\_Images" (

"Product\_Image\_ID" TEXT NOT NULL UNIQUE,

"Product\_ID" INTEGER NOT NULL,

"Product\_Image" BLOB NOT NULL,

FOREIGN KEY("Product\_ID") REFERENCES "Products"("Product\_ID"),

CONSTRAINT "PK\_Product\_Images" PRIMARY KEY("Product\_Image\_ID")

);

1. **Customer\_Cart Table**

CREATE TABLE "Customer\_Products\_Cart" (

"Customer\_Product\_Cart\_ID" INTEGER NOT NULL UNIQUE,

"Customer\_ID" INTEGER NOT NULL,

"Product\_ID" INTEGER NOT NULL,

"Quantity" NUMERIC NOT NULL,

CONSTRAINT "FK\_Customer\_Product\_Cart\_Customer" FOREIGN KEY("Customer\_ID") REFERENCES "Customer"("Customer\_ID"),

FOREIGN KEY("Product\_ID") REFERENCES "Products"("Product\_ID"),

CONSTRAINT "PK\_Customer\_Cart" PRIMARY KEY("Customer\_Product\_Cart\_ID" AUTOINCREMENT)

);

1. **Import the data from the below excel sheets**

****

****

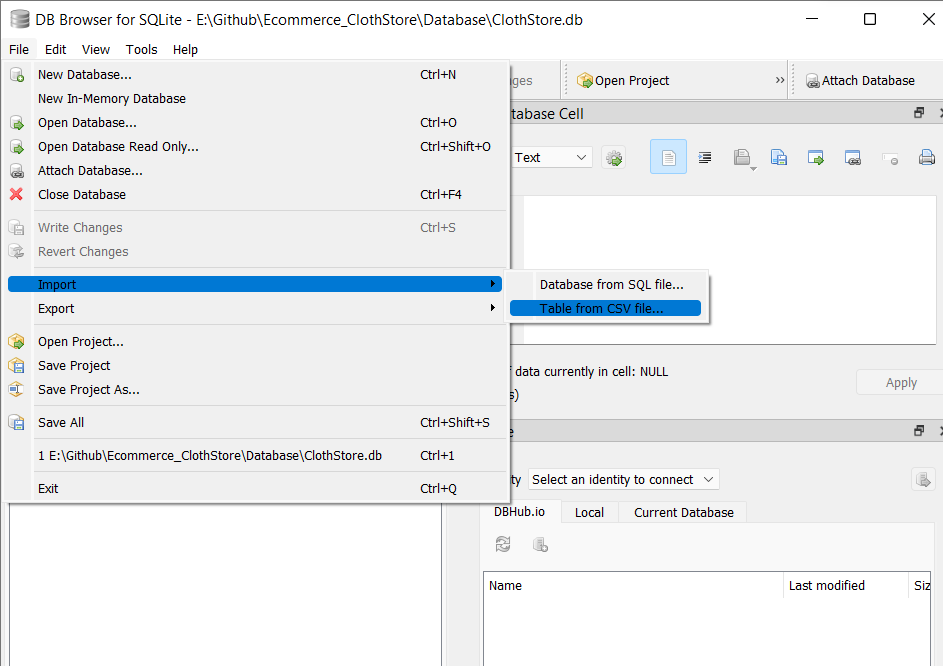
****

We will modify the product images from front end a little later

****

****

Using Select File>Import>Table from CSV file…



1. **SETTING UP MIDDLEWARE:**

Install the following packages for interaction and manipulation of data in the middleware

1. **Express:** for routing, dynamic rending with html pages, and responding to http requests

**npm install express**

1. **body-parser:** for handling json, raw, text etc.

**npm install body-parser**

1. **cookie-parser:** for parsing cookie-headers and populating req.cookies

**npm install cookie-parser**

1. **multer:** for handling multipart/form-data

**npm install multer**

1. **ALL SQL STATEMENTS:**
2. **JAVASCRIPT CODE**
3. **CONFIGURATION FILES: (JavaScript, HTML, CSS, SQL)**
4. **UI Prototype Designs and drawings**
5. **HTML, USER INTERFACE CODE**
6. **SCREENSHOTS OF FINISHED PRODUCT:**