E-Commerce Fashion Platform

Introduction :

Our project is an e-commerce fashion website developed with the MERN stack, featuring MongoDB as the database. Designed to rival popular platforms like Myntra, our site offers a seamless shopping experience. MongoDB's flexibility and scalability make it the ideal choice to manage the dynamic and diverse fashion-related data efficiently. Our goal is to provide users with a visually appealing, user-friendly platform for exploring and purchasing the latest fashion trends. Welcome to a cutting-edge online marketplace where style meets technology.

Entities and Attributes for Each Entity :

|  |  |
| --- | --- |
| Users | * UserID (ObjectId) * Username (String) * Email (String) * Password (String, hashed) * FirstName (String) * LastName (String) * DateOfBirth (Date) * AddressID (ObjectId, referencing Addresses collection) * CartID (ObjectId, referencing Cart collection) |
| Products | * ProductID (ObjectId) * Name (String) * Description (String) * Price (Number) * Size (String) * Color (String) * Brand (String) * ImageURL (String) * StockQuantity (Number) * CategoryID (ObjectId, referencing Categories collection) |
| Categories | * CategoryID (ObjectId) * Name (String) * Description(String) |
| Orders | * OrderID (ObjectId) * UserID (ObjectId, referencing Users collection) * Products (Array of Objects, each containing ProductID, Quantity, Price) * OrderDate (Date) * TotalAmount (Number) |
| Reviews | * ReviewID (ObjectId) * UserID (ObjectId, referencing Users collection) * ProductID (ObjectId, referencing Products collection) * Rating (Number) * Comment (String) * ReviewDate (Date) |
| Addresses | * AddressID (ObjectId) * UserID (ObjectId, referencing Users collection) * Street (String) * City (String) * State (String) * ZipCode (String) * Country (String) |
| Cart | * CartID (ObjectId) * UserID (ObjectId, referencing Users collection) * Items (Array of Objects, each containing ProductID, Quantity) |

Relationship between Entities :

|  |  |
| --- | --- |
| 1. Users: | * One-to-One with Addresses: One user has one address. * One-to-One with Cart: One user has one shopping cart. |
| 1. Products: | * Many-to-One with Categories: Many products belong to one category. |
| 1. Categories: | * One-to-Many with Products: One category can have many products. |
| 1. Orders: | * Many-to-One with Users: Many orders can belong to one user. * Many-to-Many with Products: Many orders can contain many products. |
| 1. Reviews: | * Many-to-One with Users: Many reviews are written by one user. * Many-to-One with Products: Many reviews are written for one product. |
| 1. Addresses: | * One-to-One with Users: One address belongs to one user. |
| 1. Cart: | * One-to-One with Users: One shopping cart belongs to one user. * Many-to-Many with Products: Many products can be in many shopping carts. |

Indexes:

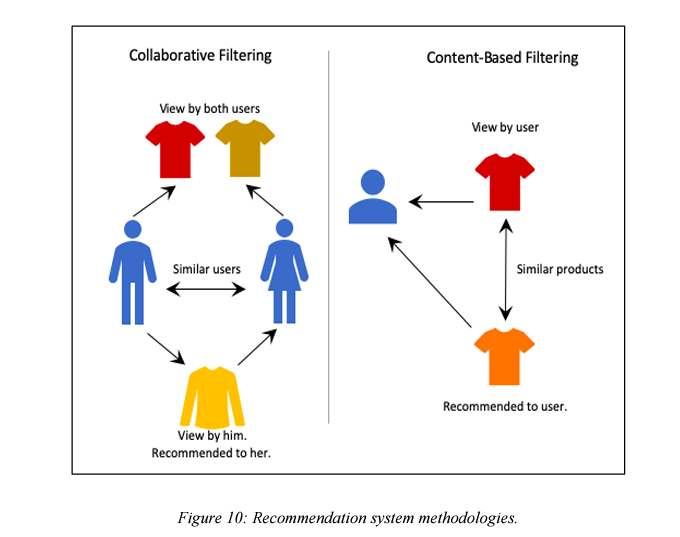
Fields that will be indexed for performance optimization.

Example:

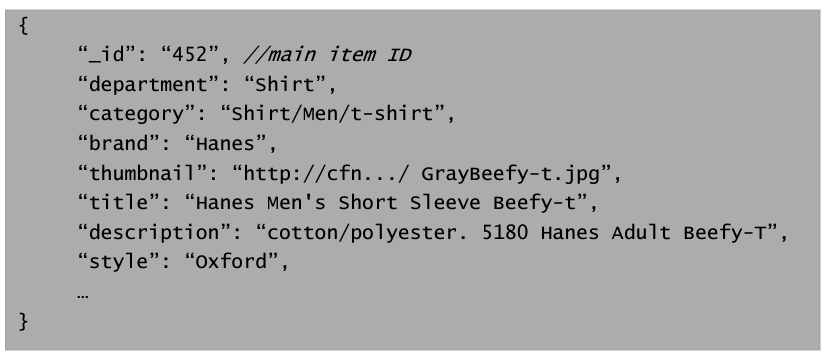
Index on Product ID for faster retrieval.

Index on User ID for quick user-related queries.

Recommended System :



Example of json file for a product :



Sample DataSets :

<https://frosch.cosy.sbg.ac.at/datasets/json/clothing/-/blob/main/clothing.json?ref_type=heads>

RESTful API :

can also include the RESTful APIs of the below referrenced website .

Referrence :

1. Myntra : <https://www.myntra.com/>
2. H & M : <https://www2.hm.com/en_in/index.html>