

FURNIHUB

Database Description:

Furnihub is a project that will allow customers to easily buy furniture based on their preferences and sellers can sell their products. We need a database for storing and manipulating this huge amounts of data and we are using MongoDB for that. In our project, People can register as sellers by providing information needed. After successful registering seller can login to the website with their predefined credentials. Seller can add products and their necessary information for selling the product. The necessary information like cost, quantity, material type, size, capacity, color etc,. Seller can view the products added and can filter products with their attributes. Seller can also update the product details whenever it is needed. Customers can also register on the website by providing the necessary information. They can login to the website after completing the registration process. The customer has access to all products added by multiple sellers. The database collections we have in this application are Admin, Seller, Customer, Product, Orders and Payment.

Admin collection contain information about the administrators who will have the privilege to remove any product or any seller.

```
{
  "_id": ObjectId("admin_id"),
  "admin_username": "admin_username",
  "email": "admin_email",
  "password": "admin_password",
  "role": "admin" // You can assign different roles if needed
}
```

Customer collection holds all the information about the customers who get registered into the website. It also stores the order history which consists of the order ids of the orders customer has placed in the past

```
{
  "_id": ObjectId("customer_id"),           // Unique identifier for the customer.
  "username": "customer_username",         // The username of the customer.
  "email": "customer_email",               // The email address of the customer.
  "mobile_number": "customer_mobile",      // The mobile number of the customer.
  "address": "customer_address",           // The address of the customer.
  "order_history": [
    ObjectId("order_id"),                  // the id of the first order
    ObjectId("order_id"),                  // the id of the second order
    ObjectId("order_id")                   // the id of the third order
  ]
}
```

Seller collection holds all the information about the sellers who get registered into the website. Along with the personal information this collection stores the ids of the products sell by the seller.

```
{
  "_id": ObjectId("seller_id"),           // Unique identifier for the seller.
  "seller_name": "seller_name",          // The name or business name of the seller.
  "email": "seller_email",               // The email address associated with the seller.
  "contact_number": "seller_contact",    // The contact number of the seller.
  "products": [ ObjectId("product_id"),  // the id of first product
                  ObjectId("product_id"), // the id of second product
                  ObjectId("product_id")  // the id of third product
                ]
}
```

Product collection contain the details about the products available in the website. Attributes like color, brand, material, dimensions, cost etc,.

```
{
  "_id": ObjectId("product_id"),          // Unique identifier for the product.
  "name": "product_name",                 // The name of the product.
  "cost": 200,                             // The cost of the product.
  "dimensions": "product_dimensions",     // Dimensions of the product.
  "color": "product_color",               // The color of the product.
  "brand": "Samsung",                     // The brand or company name associated with the seller.
  "material_type": "product_material",    // The type of material used in the product.
  "weight": 10,                           // The weight of the product.
  "seller_id": ObjectId("seller_id"),     // Unique identifier for the seller of the product.
  "rating": 4.5                           // The rating of the product, based on customer feedback.
}
```

Order collection stores the orders made by the customers. It stores the different products and their respective quantities come under the order. Order details like the total cost, order date, order status etc,.

```
{
  "_id": ObjectId("order_id"),             // Unique identifier for the order.
  "customer_id": ObjectId("customer_id"),  // Unique identifier for the customer who placed the order.
  "products": [                             // An array representing the products included in the order.
    {
      "product_id": ObjectId("product_id"), // Unique identifier for the first product in the order.
      "quantity": 2                          // The quantity of the first product in the order.
    },
    {
      "product_id": ObjectId("another_product_id"), // Unique identifier for another product in the order.
      "quantity": 1                                // The quantity of the second product in the order.
    }
  ],
  "order_date": ISODate("order_date"),     // The date and time when the order was placed.
  "total_cost": 500,                       // The total cost of the order.
  "order_status": "processing"              // The status of the order (e.g., "processing," "shipped," "delivered").
}
```

Payment collection contains the payment details of a booking. It stores order which the payment has made and attributes like payment method, date of payment, status etc,.

```
{
  "_id": ObjectId("payment_id"),           // Unique identifier for the payment.
  "order_id": ObjectId("order_id"),        // Unique identifier for the associated order.
  "payment_date": ISODate("payment_date"), // The date and time when the payment was made.
  "payment_method": "credit_card",         // The payment method used (e.g., "credit_card," "paypal").
  "amount": 500,                           // The payment amount.
  "status": "completed"                     // The status of the payment (e.g., "completed," "pending").
}
```

Database Model:



