Dharmveer singh(23BCS14174)

Practice 3 - MongoDB

Title: E-commerce Catalog with Nested Document Structure in MongoDB

Objective:

Learn how to design and implement a MongoDB data model using nested documents to represent a real-world e-commerce catalog. This task strengthens your understanding of MongoDB's flexible document structure, schema design, and handling complex relationships inside a single collection.

Task Description:

Create a MongoDB collection to represent an e-commerce catalog. Each product document should include fields such as name, price, category, and an array of nested variants. Each variant should include details like color, size, and stock. Insert a few sample product documents demonstrating different variants. Then, implement queries to retrieve all products, filter products by category, and project specific variant details. Use MongoDB shell queries or Mongoose methods to show how nested documents can be accessed and manipulated effectively.

Code Implementation:

```
// Step 1: Initialize Project
// npm init -y
// npm install express mongoose body-parser

// Step 2: Import Dependencies
const express = require('express');
const mongoose = require('mongoose');
const bodyParser = require('body-parser');

// Step 3: Initialize Express App
const app = express();
app.use(bodyParser.json());

// Step 4: Connect to MongoDB
mongoose.connect('mongodb://localhost:27017/ecommerceDB', {
    useNewUrlParser: true,
    useUnifiedTopology: true
}).then(() => console.log('MongoDB Connected'))
  .catch(err => console.log(err));

// Step 5: Define Product Schema with Nested Variants
const variantSchema = new mongoose.Schema({
    color: String,
    size: String,
    stock: Number
});

const productSchema = new mongoose.Schema({
    name: { type: String, required: true },
    price: { type: Number, required: true },
    category: { type: String, required: true },
    variants: [variantSchema]
});

const Product = mongoose.model('Product', productSchema);

// Step 6: Routes

// GET all products
```

```
app.get('/products', async (req, res) => {
    try {
        const products = await Product.find();
        res.status(200).json(products);
    } catch (err) {
        res.status(500).json({ message: err.message });
    }
});

// GET products by category
app.get('/products/category/:category', async (req, res) => {
    try {
        const products = await Product.find({ category: req.params.category });
        res.status(200).json(products);
    } catch (err) {
        res.status(500).json({ message: err.message });
    }
});

// GET products by variant color
app.get('/products/by-color/:color', async (req, res) => {
    try {
        const products = await Product.find({ "variants.color": req.params.color });
        res.status(200).json(products);
    } catch (err) {
        res.status(500).json({ message: err.message });
    }
});

// Step 7: Start Server
app.listen(3000, () => console.log('Server running on port 3000'));
```

Expected Output:

• GET /products → Retrieve all products with variants
• GET /products/category/:category → Filter products by category
• GET /products/by-color/:color → Retrieve products having variants of a specific color

Output:


GET ▾ http://localhost:3000/products
Send
Body ▾
Request GET
Response 200
No body
"_id": "686f68ed2bf5384209b236b2",
"name": "Winter Jacket",
"price": 200,
"category": "Apparel",
"variants": [
"color": "Black",
"size": "S",
"stock": 8,
"_id": "686f68ed2bf5384209b236b3"
"color": "Gray",
"size": "M",
"stock": 12,
"_id": "686f68ed2bf5384209b236b4"
"__v": 0


GET ▾ http://localhost:3000/products/category/Electronics
Send
Body ▾
Request GET
Response 200
HTTP/1.1 200 OK (6 headers)
"_id": "685f63eb90ac2728b3f11082",
"name": "Smartphone",
"price": 699,
"category": "Electronics",
"__v": 0,
"variants": []
No body


GET ▾ http://localhost:3000/products/by-color/Blue
Send
Body ▾
Request GET
Response 200
HTTP/1.1 200 OK (6 headers)
"_id": "686f68ed2bf5384209b236af",
"name": "Running Shoes",
"price": 120,
"category": "Footwear",
"variants": [
"color": "Red",
"size": "M",
"stock": 10,
"_id": "686f68ed2bf5384209b236b0"
"color": "Blue",
"size": "L",
"stock": 5,
"_id": "686f68ed2bf5384209b236b1"
"__v": 0
No body