**Project Overview:**

This project implements a RESTful API for managing a structured product catalogue system. The API is developed using Java and integrated with a database, preferably MongoDB, to handle the storage of nested product data. The API allows users to perform CRUD operations on products and search for products based on various filters.

**Product Entity Structure:**

The Product entity has the following attributes:

* **id**: A unique identifier for the product.
* **name**: The name of the product.
* **description**: A text description of the product.
* **price**: The price of the product.
* **categories**: An array of categories (strings) the product belongs to.
* **attributes**: An array of key-value pairs (objects) for additional attributes such as size, color, brand, etc.
* **availability**: An object containing availability information:
  + **inStock**: A boolean indicating if the product is in stock.
  + **quantity**: An integer representing the available quantity.
* **ratings**: An array of objects representing user ratings, each with:
  + **userId**: A unique identifier for the user who gave the rating.
  + **rating**: A numerical rating value.
  + **comment**: An optional text comment on the rating.

**Database Integration:**

The database schema is designed to handle nested arrays and objects. For MongoDB, nested documents are utilized. For SQL databases, JSON data types or relational tables can be used to represent nested structures.

**API Endpoints Documentation:**

1. **Add Product**
   * Endpoint: POST /products
   * Request Body Format: See above
   * Response Format: Status: 201 Created, Body: "Product added successfully."
2. **Get Product by ID**
   * Endpoint: GET /products/{id}
   * Response Format: Status: 200 OK, Body: Product object JSON
3. **Get All Products**
   * Endpoint: GET /products
   * Response Format: Status: 200 OK, Body: List of Product objects JSON
4. **Update Product**
   * Endpoint: PUT /products/{id}
   * Request Body Format: Same as Add Product
   * Response Format: Status: 200 OK, Body: "Product updated successfully."
5. **Delete Product**
   * Endpoint: DELETE /products/{id}
   * Response Format: Status: 200 OK, Body: "Product deleted successfully."
6. **Search Products**
   * Endpoint: GET /products/search
   * Query Parameters:
     + name (string): Name of the product
     + category (string): Category of the product
     + attributeKey (string): Key of an attribute
     + attributeValue (string): Value of an attribute
   * Response Format: Status: 200 OK, Body: List of Product objects JSON
7. **Rate Product**
   * Endpoint: POST /products/{id}/rate
   * Request Body Format: See above
   * Response Format: Status: 200 OK, Body: "Product rated successfully."

**Data Model Diagram:**

The Product entity consists of the following attributes:

* id: string
* name: string
* description: string
* price: number
* categories: array of strings
* attributes: array of key-value pairs (objects)
* availability: object containing inStock (boolean) and quantity (number)
* ratings: array of objects containing userId (string), rating (number), and comment (string)

**Database Setup and Seeding Instructions:**

* **Install MongoDB:** Install MongoDB on your system if not already installed.
* **Start MongoDB Server:** Start the MongoDB server using the appropriate command for your operating system.
* **Database Configuration:** Update the application.properties file with MongoDB connection settings.
  + spring.data.mongodb.host=localhost
  + spring.data.mongodb.port=27017
  + spring.data.mongodb.database=product-catalog
* **Database Seeding (Optional):** You can manually insert initial data into the database using MongoDB shell or a MongoDB GUI tool like MongoDB Compass. Alternatively, you can create a data initialization script to seed the database programmatically.

Example initialization script:

db.products.insertMany([

{

"id": "1",

"name": "Product 1",

"description": "Description of Product 1",

"price": 10.99,

"categories": ["Category 1", "Category 2"],

"attributes": [{"key": "Color", "value": "Red"}, {"key": "Size", "value": "Large"}],

"availability": {"inStock": true, "quantity": 100},

"ratings": [{"userId": "user1", "rating": 4, "comment": "Good product"}]

},

{

"id": "2",

"name": "Product 2",

"description": "Description of Product 2",

"price": 20.99,

"categories": ["Category 2", "Category 3"],

"attributes": [{"key": "Color", "value": "Blue"}, {"key": "Size", "value": "Medium"}],

"availability": {"inStock": true, "quantity": 50},

"ratings": [{"userId": "user2", "rating": 5, "comment": "Excellent product"}]

}

])