

: Online Bookstore Database

Context:

An online bookstore wants to manage books, authors, and price data.

Collection: books

Sample Document:

```
{  
  "title": "The MongoDB Guide",  
  "author": "Ravi Joshi",
```

```
  "price": 499,  
  "category": "Database",  
  "ratings": [4, 5, 5, 3]  
}
```

Tasks:

1. Insert 5 books with details like title, author, price, category, and rating array.
2. Find all books priced under ₹500.
3. Update the price of a book titled "The MongoDB Guide" to ₹450.
4. Delete all books from category "Old Stock".
5. Use aggregation to calculate the average rating per book.

```
-- Step 1: Create / Use the Database  
use bookstore
```

```
-- Step 2: Create the books Collection (Automatically created on insert)  
db.createCollection("books")
```

```
-- Step 3: Insert 5 Books with Details
```

```
db.books.insertMany([  
  {  
    title: "The MongoDB Guide",  
    author: "Ravi Joshi",  
    price: 499,  
    category: "Database",  
    ratings: [4, 5, 5, 3]  
  },  
  {  
    title: "Learn Python Programming",  
    author: "Alice Williams",  
    price: 350,  
    category: "Programming",  
    ratings: [5, 4, 4, 5]  
  },  
  {  
    title: "Mastering JavaScript",  
    author: "Bob Smith",  
    price: 650,
```

```

        category: "Programming",
        ratings: [3, 4, 4, 4]
    },
    {
        title: "The Art of Web Design",
        author: "Charles Green",
        price: 750,
        category: "Design",
        ratings: [5, 5, 5, 5]
    },
    {
        title: "Old Stock Book",
        author: "Jane Doe",
        price: 200,
        category: "Old Stock",
        ratings: [2, 3, 2, 1]
    }
]
)

```

```

-- Step 4: Find All Books Priced Under ₹500
db.books.find({ price: { $lt: 500 } })

```

```

-- Step 5: Update the Price of a Book Titled "The MongoDB Guide" to ₹450
db.books.updateOne(
  { title: "The MongoDB Guide" },
  { $set: { price: 450 } }
)

```

```

-- Step 6: Delete All Books from Category "Old Stock"
db.books.deleteMany({ category: "Old Stock" })

```

```

-- Step 7: Use Aggregation to Calculate the Average Rating per Book
db.books.aggregate([
  {
    $project: {
      title: 1,
      avg_rating: { $avg: "$ratings" }
    }
  }
])

```