Day 15 Coding Assignment

MongoDB Data Modeling and CRUD Operations

Objective

The objective of this assignment is to design and implement a MongoDB database for an online book management platform named **BookVerse**. The assignment focuses on NoSQL data modeling, CRUD operations, and querying/filtering data using MongoDB.

User Story 1 - Database Design & Data Modeling

use BookVerseDB

db.createCollection("authors")

db.createCollection("books")

db.createCollection("users")

Authors Collection

```
> db.authors.find();
< {
    _id: ObjectId('65a0011111111111111111'),
    name: 'Isaac Asimov',
    nationality: 'American',
    birthYear: 1920
  }
  {
    _id: ObjectId('65a0022222222222222222'),
    name: 'J.K. Rowling',
    nationality: 'British',
    birthYear: 1965
  }
  {
    _id: ObjectId('65a0033333333333333333'),
    name: 'George R.R. Martin',
    nationality: 'American',
    birthYear: 1948
  }
```

Users Collection

```
> db.users.find()
< {
    _id: ObjectId('65b001111111111111111111'),
    name: 'Reethu',
    email: 'reethu@gmail.com',
    joinDate: 2025-08-01T00:00:00.000Z
  }
  {
    _id: ObjectId('65b002222222222222222222'),
    name: 'Neethu',
    email: 'neethu@gmail.com',
    joinDate: 2025-10-10T00:00:00.000Z
  }
  {
    _id: ObjectId('6984a12e8f12a68daec292a0'),
    name: 'Geethu',
    email: 'geethu@gmail.com',
    joinDate: 2026-02-05T13:54:54.784Z
  }
```

Books Collection

```
> db.books.find()
< {
    _id: ObjectId('6984a0f58f12a68daec2929b'),
    title: 'Foundation',
    genre: 'Science Fiction',
    publicationYear: 1952,
    authorId: ObjectId('65a001111111111111111111'),
    ratings: [
      {
        user: 'Reethu',
        score: 5,
        comment: 'Classic sci-fi!'
      }
    ]
  }
  {
    _id: ObjectId('6984a0f58f12a68daec2929c'),
    title: 'I, Robot',
    genre: 'Science Fiction',
    publicationYear: 1950,
    authorId: ObjectId('65a001111111111111111111'),
    ratings: [
      {
        user: 'Neethu',
        score: 4,
        comment: 'Thought-provoking'
      }
    ]
  }
  {
    _id: ObjectId('6984a0f58f12a68daec2929d'),
    title: "Harry Potter and the Sorcerer's Stone",
    genre: 'Fantasy',
    publicationYear: 1997,
    authorId: ObjectId('65a002222222222222222222'),
    ratings: [
      {
        user: 'Preethu',
        score: 5,
        comment: 'Magical!'
      }
    ]
  }
  {
    _id: ObjectId('6984a0f58f12a68daec2929e'),
    title: 'A Game of Thrones',
    genre: 'Fantasy',
    publicationYear: 1996,
    authorId: ObjectId('65a003333333333333333333'),
    ratings: [
      {
        user: 'Reethu',
        score: 5,
        comment: 'Epic fantasy'
      }
    ]
  }
```

User Story 2 - CRUD Operations

1.Insert New User

```
db.users.insertOne({
  name: "Geethu",
  email: "geethu@gmail.com",
  joinDate: new Date()
})
```

2.Retrieve All Science Fiction Books

```
db.books.find({ genre: "Science Fiction" })
```

3.Update Publication Year

```
db.books.updateOne(
  { title: "Foundation" },
  { $set: { publicationYear: 1952 } }
)
```

4. Delete One User

```
db.users.deleteOne({ name: "Preethu" })
```

5. Add Rating Using $push

```
db.books.updateOne(
  { title: "A Game of Thrones" },
  {
   $push: {
     ratings: {
       user: "Reethu",
       score: 5,
       comment: "Epic fantasy"
     }
   }
  }
)
```

User Story 3 - Querying & Filtering

1. Books Published After 2015

```
db.books.find({ publicationYear: { $gt: 2015 } })
```

```
> db.books.find({ publicationYear: { $gt: 2015 } })
< {
    _id: ObjectId('6984a0f58f12a68daec2929f'),
    title: 'The Winds of Winter',
    genre: 'Fantasy',
    publicationYear: 2025,
    authorId: ObjectId('65a003333333333333333333'),
    ratings: []
  }
```

2. Authors Who Wrote Fantasy Books

```
db.authors.find({
  _id: {
    $in: db.books.distinct("authorId", { genre: "Fantasy" })
  }
})
```

```
> db.authors.find({
    _id: {
      $in: db.books.distinct("authorId", { genre: "Fantasy" })
    }
  })
< {
    _id: ObjectId('65a002222222222222222222'),
    name: 'J.K. Rowling',
    nationality: 'British',
    birthYear: 1965
  }
  {
    _id: ObjectId('65a003333333333333333333'),
    name: 'George R.R. Martin',
    nationality: 'American',
    birthYear: 1948
  }
```

3. Users Joined in Last 6 Months

```
db.users.find({
  joinDate: {
    $gte: new Date(new Date().setMonth(new Date().getMonth() - 6))
  }
})
```

```
> db.users.find({
    joinDate: {
      $gte: new Date(new Date().setMonth(new Date().getMonth() - 6))
    }
  })
< {
    _id: ObjectId('65b00222222222222222222'),
    name: 'Neethu',
    email: 'neethu@gmail.com',
    joinDate: 2025-10-10T00:00:00.000Z
  }
  {
    _id: ObjectId('6984a12e8f12a68daec292a0'),
    name: 'Geethu',
    email: 'geethu@gmail.com',
    joinDate: 2026-02-05T13:54:54.784Z
  }
```

4. Books With Average Rating > 4

```
db.books.aggregate([
  { $unwind: "$ratings" },
  {
   $group: {
     _id: "$title",
     avgRating: { $avg: "$ratings.score" }
   }
  },
  { $match: { avgRating: { $gt: 4 } } }
])
```

```
> db.books.aggregate([
    { $unwind: "$ratings" },
    {
      $group: {
        _id: "$title",
        avgRating: { $avg: "$ratings.score" }
      }
    },
    { $match: { avgRating: { $gt: 4 } } }
  ])
< {
    _id: 'A Game of Thrones',
    avgRating: 5
  }
  {
    _id: 'Foundation',
    avgRating: 5
  }
  {
    _id: "Harry Potter and the Sorcerer's Stone",
    avgRating: 5
  }
```

Conclusion

This assignment successfully demonstrates MongoDB NoSQL data modeling, CRUD operations, and advanced querying techniques. Relationships were efficiently handled using references and embedded documents, making the database scalable and suitable for a real-world application.