Day 1 - Practical Lab (Student Version)

Library Management System

Task 1: Environment Setup (15 minutes)

Steps:

1. Verify MongoDB container is running

```
docker ps | grep mongodb-course
```

2. Connect to MongoDB

```
docker exec -it mongodb-course mongosh
```

3. Create library database

```
use library_db
```

Task 2: Books Management (30 minutes)

a) Add new books:

Insert the following books into the collection:

```
db.books.insertMany([
    {
        title: "Introduction to Algorithms",
        author: "Dr. Mohamed Ahmed",
        isbn: "978-111222333",
        price: 85,
        publishYear: 2023,
        categories: ["algorithms", "computer science"],
        language: "Arabic",
        pages: 450,
        inStock: true
    },
    {
        title: "Computer Networks",
        author: "Dr. Sara Mahmoud",
        isbn: "978-444555666",
        price: 70,
        publishYear: 2022,
        categories: ["networks", "technology"],
        language: "Arabic",
        pages: 380,
        inStock: false
    },
    {
        title: "Information Security",
        author: "Dr. Khaled Abdullah",
        isbn: "978-777888999",
        price: 95,
        publishYear: 2023,
        categories: ["security", "technology"],
        language: "Arabic",
        pages: 520,
        inStock: true
    }
])
```

b) Verify insertion:

```
// Display all books
db.books.find().pretty()

// Count available books
db.books.countDocuments({inStock: true})
```

Task 3: Search and Query (25 minutes)

Execute the following queries:

1. **Find books by Dr. Mohamed Ahmed:**
2. **Find only available books:**
3. **Find books priced more than 80:**

```
// Write your query here
```

4. **Find books in "technology" category:**
5. **Show only title and author for books published in 2023:**
6. **Find books with more than 400 pages:**
7. **Find books published between 2022 and 2023:**
8. **Count books by each author:**

```
// Hint: You may need aggregation for this
```

Task 4: Data Updates (20 minutes)

Required tasks:

1. **Increase all book prices by 10%:**

```
// Use updateMany with $mul operator
```

2. **Update "Computer Networks" book status to available:**

3. **Add "discount" field with value true for books priced over 90:**

4. **Add "reference" category to "Introduction to Algorithms":**

```
// Use $push operator
```

5. **Remove "networks" category from all books that have it:**

```
// Use $pull operator
```

6. **Update the page count of "Information Security" to 550:**

Task 5: Members Management (20 minutes)

a) Add new members:

```
db.members.insertMany([
    {
        name: "Ahmed Mohamed Ali",
        email: "ahmed.ali@student.iti.gov.eg",
        membershipType: "student",
        joinDate: new Date("2023-09-01"),
        borrowedBooks: [],
        maxBooksAllowed: 3,
        isActive: true
    },
    {
        name: "Fatma Hassan Mahmoud",
        email: "fatma.hassan@teacher.iti.gov.eg",
        membershipType: "teacher",
        joinDate: new Date("2022-01-15"),
        borrowedBooks: ["978-111222333"],
        maxBooksAllowed: 5,
        isActive: true
    },
    {
        name: "Omar Khaled Salem",
        email: "omar.salem@student.iti.gov.eg",
        membershipType: "student",
        joinDate: new Date("2023-03-10"),
        borrowedBooks: ["978-444555666", "978-777888999"],
        maxBooksAllowed: 3,
        isActive: false
    }
])
```

b) Member queries:

1. **Find all active students:**

2. **Find members with borrowed books:**

```
// Write your query here (check array length > 0)
```

3. **Find teachers who can borrow more than 3 books:**

4. **Update Omar's status to active:**

5. **Add a new borrowed book to Ahmed's record:**

```
// Use $push to add "978-777888999" to Ahmed's borrowedBooks
```

Task 6: Advanced Queries (25 minutes)

Pattern Matching and Complex Queries:

1. **Find books with titles starting with "Computer":**

```
// Use regular expression
```

2. **Find all members with email ending in "student.iti.gov.eg":**

```
// Use regular expression
```

3. **Find books in both "technology" and "security" categories:**

```
// Use $all operator
```

4. **Find members who joined in 2023:**

```
// Use date range query
```

5. **Find books that are either expensive (>90) OR out of stock:**

```
// Use $or operator
```

6. **Find books that are NOT in "algorithms" category:**

```
// Use $ne or $nin operator
```

Task 7: Final Challenge (20 minutes)

Create a comprehensive library report:

1. **Total number of books in library**

2. **Number of available vs unavailable books**

```
// Write separate queries for each
```

3. **Average book price**

```
// Use aggregation framework
```

4. **Most expensive and cheapest book**

```
// Write separate queries with sort and limit
```

5. **Books sorted by price (descending)**

6. **List of all unique authors**

```
// Use distinct() method
```

7. **Count of books by publication year**

```
// Use aggregation to group by publishYear
```

8. **Members with most borrowed books**

```
// Find member(s) with highest number of borrowed books
```