Assignment No. 11

Assignment Title:Create the collection Books having the following fields TITLE, DESCRIPTION, BY, URL,TAGS AND LIKES.
Implement the following Aggregation and Indexing Queries
1. Find the number of books published by john.
2. Find books which have minimum likes and maximum likes published by john.
3. Find the average number of likes of the books published by john.
4. Find the first and last book published by john..
5. Create an index on author name.
Display the books published by john and check if it uses the index which we have created

Queries:

Step 1: Create the "Books" Collection

```
db.createCollection("Books")
db.Books.insertMany([
        {
        TITLE: "Book 1",
        DESCRIPTION: "Description 1",
        BY: "John",
        URL: "url1",
        TAGS: ["tag1", "tag2"],
        LIKES: 100
        },
        {
        TITLE: "Book 2",
        DESCRIPTION: "Description 2",
        BY: "John",
        URL: "url2",
        TAGS: ["tag2", "tag3"],
        LIKES: 150
        },
        {
        TITLE: "Book 3",
        DESCRIPTION: "Description 3",
        BY: "Alice",
        URL: "url3",
        TAGS: ["tag1", "tag4"],
        LIKES: 80
        }
])
```

Step 2: Execute Aggregation Queries

```
db.Books.aggregate([
    {
    $match: { BY: "John" }
    },
    {
    $group: {
    _id: null,
    count: { $sum: 1 }
    }
    }
])
```

1. **Find books with the minimum and maximum likes published by John:**

   ```
   db.Books.aggregate([
       {
       $match: { BY: "John" }
       },
       {
       $sort: { LIKES: 1 }
       },
       {
       $group: {
       _id: "$BY",
       minLikes: { $first: "$LIKES" },
       maxLikes: { $last: "$LIKES" }
       }
       }
   ])
   ```

2. **Find the average number of likes of books published by John:**

   ```
   db.Books.aggregate([
       {
       $match: { BY: "John" }
       },
       {
       $group: {
       _id: "$BY",
       avgLikes: { $avg: "$LIKES" }
       }
       }
   ])
   ```

3. **Find the first and last book published by John:**

   ```
   db.Books.aggregate([
       {
       $match: { BY: "John" }
       },
       {
   ```

```
        $sort: { TITLE: 1 }
        },
        {
        $group: {
        _id: "$BY",
        firstBook: { $first: "$TITLE" },
        lastBook: { $last: "$TITLE" }
        }
        }
])
```

Step 3: Create an Index on Author Name

```
db.Books.createIndex({ BY: 1 })
```

Step 4: Display Books Published by John Using the Index

```
db.Books.find({ BY: "John" }).explain("executionStats")
```