

Day 8 Coding Assignment – Indexing, Aggregation, and MongoDB Atlas

Duration: 50–60 Minutes

Focus Areas: Indexing, Query Optimization, Aggregation Framework, and MongoDB Atlas

Scenario Background

BookVerse is now scaling, and you are tasked with optimizing database performance and enabling analytics using MongoDB's advanced features.

You will implement indexes, run aggregation pipelines, and connect your local database to MongoDB Atlas.

User Stories and Tasks

User Story 1 – Indexing and Query Optimization

As a developer, I need to optimize frequently used queries to improve database performance.

Tasks:

1. Identify frequently queried fields (e.g., `genre`, `authorId`, `ratings.score`).
2. Create indexes on these fields using the `createIndex()` method.
3. Compare query performance using the `explain("executionStats")` command before and after creating indexes.

4. Drop an unnecessary index using `dropIndex()` and note its impact.

Concepts Covered: Indexing, query optimization, performance analysis

User Story 2 – Aggregation Framework

As a data analyst, I need to generate reports about books and their ratings using MongoDB's aggregation pipeline.

Tasks:

1. Calculate the average rating per book using `$unwind`, `$group`, and `$avg`.
2. Retrieve the top 3 highest-rated books.
3. Count the number of books published per genre.
4. Find authors who have more than 2 books published.
5. Display the total reward points (sum of all ratings) received by each author.

Concepts Covered: Aggregation stages, grouping, projection, sorting

User Story 3 – MongoDB Atlas Connection

As a backend engineer, I need to deploy and connect the database to MongoDB Atlas for cloud management.

Tasks:

1. Create a free cluster in MongoDB Atlas.
2. Create a database named `BookVerseCloudDB`.
3. Import your local collections (Authors, Books, Users) into Atlas.
4. Connect to your cluster using the connection string in a Node.js script or MongoDB Compass.

Concepts Covered: Cloud database setup, connection configuration, data import

Expected Deliverables

- Screenshots showing:
 - Index creation and `explain` output comparison
 - Aggregation pipeline results
 - MongoDB Atlas dashboard and successful connection
 - A `.json` or `.js` file with aggregation queries and index creation commands
-

Self-Evaluation Checklist

Criteria	Self-Rating (✓/✗)	Notes
Indexes created and analyzed with explain plans		
Query optimization results interpreted correctly		
Aggregation pipelines constructed using correct stages		
Results correctly reflect analytical objectives		
Atlas setup and connection completed successfully		

Bonus Challenge (Optional)

1. Create a compound index on { `genre: 1`, `publicationYear: -1` } and analyze its impact.
2. Build a stored aggregation query to return the top-rated author by average book score.