Assignment 1:  
  
Case Study: Online Library Management System  
  
Scenario:  
You are tasked with building an Online Library Management System for a university. The system should allow students and faculty to borrow and return books, view available books, and track due dates. The system should also provide administrators with tools to manage the library's collection and user accounts.  
  
Requirements:  
  
1. Database Design:  
Design a MongoDB schema to represent the library's collection of books, users, and transactions (borrowing and returning books).  
Each book should have attributes such as title, author, ISBN, and availability status.  
Users should have attributes like name, email, and a list of borrowed books.  
Transactions should include information on which book was borrowed or returned, the user involved, and timestamps.  
  
2. Java Application:  
Develop a Java-based application to interact with the MongoDB database.  
Implement user authentication and authorization, allowing administrators to manage user roles and privileges.  
Users should be able to:  
a. View available books.  
b. Borrow a book (if available) and set a due date.  
c. Return a borrowed book.  
d. View their borrowed books and due dates.  
Administrators should be able to:  
a. Add, edit, or remove books from the library's collection.  
b. Add, edit, or remove user accounts.  
c. View transaction history.  
  
Business Logic:  
1. Implement business logic to enforce rules such as ensuring that a book is available before allowing a user to borrow it and tracking due dates.  
2. Implement logic to prevent users from borrowing more books than allowed based on their user type (student, faculty).  
  
Error Handling and Logging:  
Implement proper error handling and logging for the application to record important events and handle exceptions gracefully.

Assignment Details:  
  
1. Index Creation:  
Write a Java program that connects to a MongoDB database and creates a compound index on a specified collection. The index should involve multiple fields. After creating the index, demonstrate its impact on query performance with and without the index.  
  
2. Text Indexing:  
Build a Java program that demonstrates text indexing in MongoDB. Create a text index on a specific field within a collection and perform full-text search operations using MongoDB's text search capabilities.  
  
3. Geospatial Queries:  
Develop a Java application that performs geospatial queries in MongoDB. Allow users to find and display locations within a certain distance of a specified point on a map using MongoDB's geospatial indexing and queries.  
  
Note: You can choose any data set to implement the solutions for above questions.

Assignment 1 Detail:  
  
Problem Statement: Building a MongoDB-Powered Blogging Platform  
  
Background:  
You are tasked with building a simple blogging platform that allows users to create, read, update, and delete blog posts. MongoDB will be used to store and manage the blog data. Users should be able to register, log in, and perform CRUD operations on their blog posts.  
  
Assignment Tasks:  
  
Task 1: Database Setup  
Set up a MongoDB database to store user and blog post data. Define the necessary collections and schema for users and blog posts.  
  
Task 2: User Registration and Authentication (15 points)  
Implement user registration and authentication functionality. Users should be able to:  
Register with a unique username and password.  
Log in using their credentials.  
Securely hash and store passwords in the database.  
  
Task 3: Blog Post Management  
Implement the following features for managing blog posts:  
Create a new blog post.  
Read a list of all blog posts.  
Read a specific blog post by its ID.  
Update an existing blog post.  
Delete a blog post.  
Ensure that users can only edit or delete their own posts.  
  
Task 4: User Profile  
Create user profiles that display the user's information and a list of their blog posts.  
  
Task 5: Search and Filters  
Implement search functionality to allow users to search for blog posts by keywords or tags. Additionally, add filtering options to sort and view blog posts by categories or dates.  
  
Task 6: Comments  
Enable users to comment on blog posts. Comments should include the author's name, timestamp, and content. Implement a feature that allows users to edit and delete their comments.

Assignment 4 Details:  
  
Case Study Scenario:  
You are tasked with developing a data analytics platform that analyzes user data from a social media platform. The platform collects user interactions such as likes, comments, and shares on posts. You need to build a Java application that uses MongoDB aggregate operations to derive valuable insights from this user interaction data.  
  
Requirements:  
  
Data Collection:  
Create a MongoDB database to store user interaction data. Design a collection to store records of user interactions, including user IDs, post IDs, interaction types (e.g., "like," "comment," "share"), and timestamps.  
  
Data Generation:  
Generate mock user interaction data for testing and analysis. You can use libraries like Faker to create realistic user profiles and interactions.  
  
Java Application:  
Develop a Java application that connects to the MongoDB database and performs aggregate operations to derive insights from the data.  
Aggregate Queries:  
  
Implement the following aggregate queries using the MongoDB Java driver:  
  
Query 1: Calculate the total number of interactions for each post.  
Query 2: Find the top posts with the most likes.  
Query 3: Calculate the average number of comments per post.  
Query 4: Determine the users who have the highest engagement (likes, comments, shares) on their posts.  
Query 5: Identify posts that have received the most engagement overall (likes + comments + shares).

Assignment 5 Details:  
  
Problem Statement: Title: Building an Online Bookstore System with MongoDB and Java  
  
Background:  
You are tasked with developing an online bookstore system that allows customers to browse, search, and purchase books. MongoDB will be used to store and manage book, customer, and order data.  
  
Assignment Tasks:  
  
Task 1: Database Setup  
Set up a MongoDB database to store book, customer, and order data. Define the necessary collections and schema for books, customers, and orders.  
  
Task 2: Book Catalog  
Implement a book catalog that includes the following features:  
Display a list of available books.  
Allow customers to view detailed book information.  
Implement categories and filters for easy book discovery.  
Add the ability to search for books by title, author, or genre.  
  
Task 3: Shopping Cart  
Create a shopping cart feature that allows customers to:  
Add books to their cart.  
View and update the cart contents.  
Remove items from the cart.  
  
Task 4: User Authentication  
Implement user registration and authentication functionality. Users should be able to:  
Register with a unique email and password.  
Log in using their credentials.  
Securely hash and store passwords in the database.  
  
Task 5: Checkout and Orders  
Implement the checkout and order processing system:  
Allow customers to place orders.  
Store order information, including items, quantities, and total amounts.  
Provide order history for each customer.  
Ensure that users can only view their own order history.  
  
Task 6: Reviews and Ratings  
Add the ability for customers to leave reviews and ratings for books they have purchased.  
  
Task 7: Inventory Management  
Create an inventory management system to track book quantities. Ensure that books are marked as "out of stock" when their inventory reaches zero.