Project Description:
In this project, you will create an online bookstore API that allows users to view, search, and purchase books. The API will be built using FastAPI and the book data will be stored in MongoDB.

Requirements:
Book Model: You will create a Pydantic model for the book data that includes the following fields: title, author, description, price, and stock.

MongoDB Connection: You will establish a connection to MongoDB using PyMongo. The connection should be asynchronous.

Data Validation: You will use Pydantic to validate the incoming book data before it is stored in MongoDB. The API should return an appropriate error message if the data fails validation.

Aggregation: You will use MongoDB's aggregation framework to retrieve the following data:

- The total number of books in the store
- The top 5 bestselling books
- The top 5 authors with the most books in the store

Indexing: You will create appropriate indexes for the MongoDB collection to optimize query performance.

Operators: You will use MongoDB's query operators to implement the following search functionality:

- Search for books by title
- Search for books by author
- Search for books by price range

Asynchronous Programming: All database operations should be done asynchronously to ensure the API remains responsive and performant.

API Endpoints: You will create the following API endpoints:

- GET /books: Retrieves a list of all books in the store
- GET /books/{book_id}: Retrieves a specific book by ID
- POST /books: Adds a new book to the store
- PUT /books/{book_id}: Updates an existing book by ID
- DELETE /books/{book_id}: Deletes a book from the store by ID
- GET /search?title={}&author={}&min_price={}&max_price={}: Searches for books by title, author, and price range

Deliverables:

- GitHub repository containing the source code for the project
- Documentation on how to run and use the API with names of all the team members. (README.txt)
- A brief video on the project, including the design choices and implementation details.

Grading Criteria:

- Implementation of all required features and functionality
- Proper use of FastAPI and MongoDB
- Proper use of Pydantic models and data validation
- Proper use of MongoDB aggregation and indexing
- Proper use of MongoDB query operators
- Proper use of asynchronous programming
- Code readability and organization
- Documentation quality

Good luck with the project!