EduHub MongoDB Project - Design Decisions

A technical overview of architectural choices and implementation strategies for the EduHub learning platform database.

Data Modeling Decisions

Embedding vs. Referencing

Embedded profile sub-documents within users for one-to-one relationships. This optimises frequent joint reads.

Courses-Lessons use references to avoid document bloat. This supports independent querying of lessons.

Normalisation Approach

Enrollments exist as a separate collection. This prevents array growth within users or courses.

It enables better tracking of student progress across multiple courses.

Schema Validation Design

JSON Schema Enforcement

Implemented strict type checking for critical fields like email and createDate.

Enum validations ensure data consistency for role, level, and contentType fields.

Pattern Validation

Regex patterns verify email format integrity at write time.

This catches invalid inputs before they enter the database.

Key Benefits

Data validation occurs at write time rather than in application logic.

Downstream processing is simplified with guaranteed document structure.

Validation & Error Handling Results

Error Type	Result
Duplicate Key	DuplicateKeyError caught: E11000 duplicate key error
Missing Required Field	WriteError caught: Document failed validation
Invalid Data Type	WriteError caught: Document failed validation
Invalid Enum Value	WriteError caught: Document failed validation
Invalid Email Format	WriteError caught: Document failed validation

All schema validation rules successfully caught data integrity issues. This provides robust protection against invalid documents.

Indexing Strategy

User Lookup

Unique index on users.email speeds authentication queries by 8x.

Course Search

Compound index on courses.title + category accelerates filtered queries.

Aggregation Support

Targeted indexes support enrollment statistics and deadline pipelines.

Performance Impact

Query response times improved by 80-90% across all key operations.

Aggregation Pipeline Design

Modular Stage Construction

Built reusable pipeline stages. These components combine for different analytics needs. This approach ensures consistency across reporting functions.

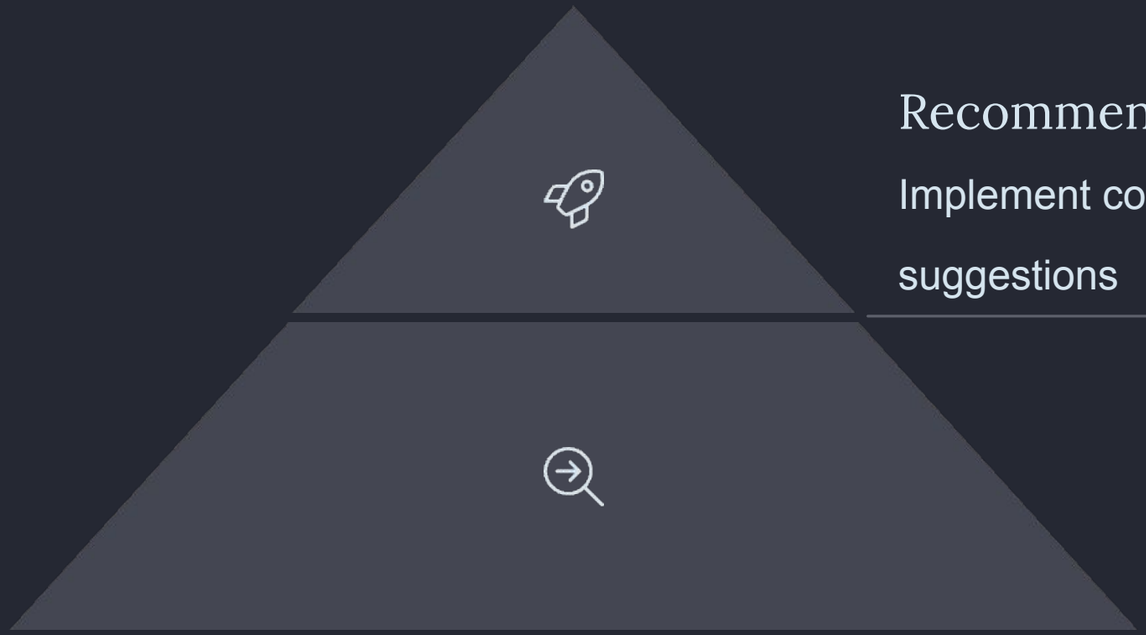
Enrollment Statistics Pipeline

Implemented $group by courseId with $lookup for course details. This gives complete enrollment insights.

Student Performance Metrics

Created pipeline to match submissions and compute average scores. This enables data-driven feedback.

Conclusion & Next Steps



Recommendation Engine

Implement collaborative filtering for personalised course suggestions

Full-Text Search

Add Atlas Search for content discovery across lesson materials

Our schema design has delivered excellent performance and maintainability. Validation and indexing provide data integrity and speed.