# ASSIGNMENT - 2

Name: Tejasree Arepalli

Project Title: Smart Student Management System with MongoDB

## Features:

* • Database setup on MongoDB (local/Atlas)
* • Created collections: students and courses
* • Implemented full CRUD operations
* • Established one-to-many relationship (students to courses)
* • Created indexes for optimized search and unique fields
* • Used $lookup to fetch course details inside student records

## Code Implementation:

### Database Setup:

In Mongo Shell:  
use student\_management

### Collection Schemas:

* students Collection:

{  
 \_id: ObjectId,  
 name: String,  
 email: String,  
 age: Number,  
 enrolledCourses: [ObjectId],  
 createdAt: Date  
}

* courses Collection:

{  
 \_id: ObjectId,  
 courseName: String,  
 instructor: String,  
 credits: Number  
}

### CRUD Operations:

Create (Insert Students and Courses):

db.courses.insertMany([  
 { courseName: "Computer Science", instructor: "Dr. Smith", credits: 4 },  
 { courseName: "Mathematics", instructor: "Prof. Adams", credits: 3 }  
]);  
  
db.students.insertOne({  
 name: "John Doe",  
 email: "john@example.com",  
 age: 20,  
 enrolledCourses: [],  
 createdAt: new Date()  
});

Read (Fetch all students):

db.students.find()

Read (Students with course details using $lookup):

db.students.aggregate([  
 {  
 $lookup: {  
 from: "courses",  
 localField: "enrolledCourses",  
 foreignField: "\_id",  
 as: "enrolledCourseDetails"  
 }  
 }  
])

Update (Update student name or enrolled courses):

db.students.updateOne(  
 { email: "john@example.com" },  
 { $set: { name: "Johnathan Doe" } }  
);

Delete (Remove a student):

db.students.deleteOne({ email: "john@example.com" })

Delete (Remove a course):

db.courses.deleteOne({ courseName: "Mathematics" })

### Indexing:

Unique email in students:

db.students.createIndex({ email: 1 }, { unique: true })

Text index on courseName in courses:

db.courses.createIndex({ courseName: "text" })

### Relationships:

• One-to-Many relationship between students and courses.

• enrolledCourses field in students is an array of ObjectId referencing courses.

• Used $lookup to populate the enrolled course details dynamically.

## How It Works:

* • MongoDB stores student and course information in two separate collections.
* • Students can be enrolled in multiple courses.
* • When fetching students, the course details are joined using $lookup.
* • Indexes are used for:
* • - Ensuring email uniqueness for students
* • - Speeding up text-based searches on course names
* • Full CRUD operations allow managing both students and courses efficiently.