**Course Learning App**

The Course Learning App is a platform designed to connect students and teachers. It allows teachers to create and manage courses, lessons, and topics while tracking performance. Students can enroll in courses, engage with the content at their own pace, and provide feedback, fostering a dynamic learning environment.

#### **Authentication**

* Users can create accounts using their **name**, **email**, and **password**.
* Two user roles:
  + **Student**
  + **Teacher**

#### **Teacher Features**

1. **Content Management**
   * Add, edit, and delete:
     + **Courses**: Includes title and description.
     + **Lessons**: Building blocks of a course.
     + **Topics**: Detailed sections or quizzes within lessons.
   * Modify or remove any content at any time.
2. **Performance Tracking**
   * View analytics for courses, such as:
     + Number of students who viewed the course.
     + Likes and feedback received.
   * Understand content performance and make improvements.
3. **Interaction with Students**
   * Track student engagement.
   * Monitor students who follow their courses.

#### **Student Features**

1. **Course Interaction**
   * Browse and enroll in courses.
   * View course lessons and topics with automatic progress tracking.
2. **Feedback and Engagement**
   * Like courses.
   * Provide feedback.
   * Participate in quizzes to test understanding.
3. **Teacher Interaction**
   * Follow teachers whose courses they like.

**Notes:**

**Stack Requirements:**

* Node.js
* Express.js
* MongoDB
* Typescripts
* Mongoose

**Error Handling:**

Implement proper error handling throughout the application. Use global error handling middleware to catch and handle errors, providing appropriate error responses with status codes and error messages.

**Error Response Object Should include the following properties:**

* **success → false**
* **message → Error Type → Validation Error, Cast Error, Duplicate Entry**
* **error message**
* **stack**

**Errors response will be like this;-**



The success response will be like this

{

"success": true,

"statusCode":200,

"message": "Users created successfully",

"data": {}

}  
  
**Retrieve paginated and filtered cow listings: ( You do not need to implement pagination as we implemented, you can do as you want )**

Route: /api/v1/course?

Query parameters: (Case Insensitive)

* page: The page number for pagination (e.g., ?page=1).
* limit: The number of cow listings per page (e.g., ?limit=10).
* searchTerm: The search query string for searching English (e.g., ?query=English). (Search Fields should be Subject, Level, Title)

### **Get All**

