**YouTube Channel Server Application**

This server application is designed to manage and provide data related to YouTube channels. It contains three collections in a MongoDB database to store and manipulate data for aliens (channel details), users, and comments. The application is built using **Node.js**, **Express**, and **MongoDB**, with **Mongoose**

**Collections**

**1. Aliens**

* Stores details about YouTube channels and their subscription information.
* **Fields:**
* name (String, required): Name of the customer.
* tech (String, required): Technology used by the customer.
* sub (Boolean, required): Subscription status.
* channel (String, required): Name of the YouTube channel.
* subscriber (Number, optional): Number of subscribers (default: 0).
* country (String, optional): Country of origin (default: "Unknown").

**2. Users**

* Stores details about users interacting with the customers or channels.
* **Fields:**
  + - email (String, required, unique): User's email address.
    - age (Number, required): User's age.
    - alienId (ObjectId, required): Reference to an Alien document.

**3. Comments**

* Stores comments made by users on aliens (channels).
* **Fields:**
  + - text (String, required): Comment text.
    - alienId (ObjectId, required): Reference to an Alien document.
    - userId (ObjectId, required): Reference to a User document.
    - createdAt (Date, optional): Timestamp for the comment (default: current time).

**Routes**

The application supports the following routes:

**Aliens Routes**

**http://Localhost:5000/aliens**

* **POST** /aliens  
  Add a new alien (channel) to the database.
* **GET** /aliens  
  Fetch all aliens (channels).
* **GET** /aliens/:id  
  Fetch a specific alien by ID.
* **PATCH** /aliens/:id  
  Update name fields based on the Id
* **DELETE** /aliens/:id  
  Remove a specific alien from the database.

And with validations if the input does not match the specification it will throw the e

**Users Routes**

**http://Localhost:5000/users**

* **POST** /users  
  Add a new user to the database.
* **GET** /users  
  Fetch all users.
* **GET** /users/:id  
  Fetch a specific user by ID.

**Comments Routes**

**http://Localhost:5000/comments/:alienid**

* **POST** /comments  
  Add a new comment to the database.
* **GET** /comments  
  Fetch all comments.
* **GET** /comments/:id  
  Fetch a specific comment by ID.