Blue Ribbon - Backend Development Internship Task

Your task is to develop a NestJS-based backend application that exposes REST API endpoints for managing a sporting club. The system should handle members, sports, and sport subscriptions. Use Supabase as the PostgreSQL database for this project.

Each member has a first and last name, gender (male or female only), birthdate, and subscription date. Members may also have family members linked to them, and each family member can be associated with only one central member.

Each sport has a name, a subscription price, and an allowed gender (male, female, or mix). Members can subscribe to multiple sports, with each subscription being either group or private (note: this does not affect the subscription price).

> 📝 Note:
> **If you make any assumptions during the development of the task, please make sure to document them. This can be either in code comments or in your submission email.**

Your application should expose endpoints to:

1. Create a sport
2. Get all sports

> 💡 **Bonus**: These sports will be displayed on our mobile app's home screen and requested thousands of times. Can you implement this endpoint efficiently?

3. Update a sport
4. Delete a sport
5. Create a member
6. Get a member's details
7. Update a member
8. Delete a member
9. Subscribe a member to a sport

> 💡 **Bonus**: Can you ensure that a member cannot have more than one subscription to the same sport?

10. Unsubscribe a member from a sport

Extra Bonus Points

- Writing **unit tests** is a good practice and ensures good code quality. Write test cases that verify the correctness of your functions.
- Implementing **input validation** (e.g. using DTOs with `class-validator` or Joi schema-based validation) is also considered a plus.

Submission Instructions

Remove the `node_modules` folder and send your project folder in `.zip` format to tae@blueribbon-inc.com

Provide clear instructions on how to set up and run your code.

> 💡 **Bonus:** Instead of sending a zipped folder of your project, you can publish your task on a public GitHub repository with a **clear README** that includes:
>
> 1. Project setup instructions
> 2. An API summary
>
> Then send the link to your repository in the email.

Evaluation Criteria

- Code readability
- Efficient and correct handling of database operations
- Accuracy and completeness of the implemented functionality