

# Code Review -Frontend - 2.0

Code Review for Sprint 3 frontend [Including functionalities: "coach", "ageGroup"]

Reviewer: Xintao Chen

Date: 2023/10/17

Summary: Provide a brief overview of the changes and their purpose in the codebase.

Code Review Report Download:

[CodeReviewOne\\_Sprint\\_3\\_Frontend.xlsx](#)

Code Review Feedback and Checklist:

Code Quality:

- ☒ The code follows coding standards and best practices.
- ☒ Variable and function names are descriptive and follow naming conventions.
- ☒ Code is well-organized with proper indentation.
- ☒ There are no commented-out code blocks or unused code.
- ☒ Error handling and edge cases are considered and handled gracefully.

Functionality:

- ☒ The code accomplishes the intended functionality as described in the requirements or user stories.
- ☒ All edge cases and error scenarios are considered and handled appropriately.
- ☒ The code is efficient and performs well under expected loads.

Testing:

- ☒ Unit tests are provided for new functionality or changes.
- ☒ Existing unit tests are updated to reflect the changes.
- ☒ Integration tests, if applicable, cover the new functionality or changes.
- ☒ Tests provide adequate coverage and verify edge cases.

Documentation:

- ☒ Inline code comments are clear and provide insights where necessary.
- ☒ README or documentation is updated to reflect any changes in setup, configuration, or usage.
- ☒ Any new or modified APIs are documented.
- ☒ User-facing changes are documented for end-users (if applicable).

Security:

- ☒ The code is reviewed for security vulnerabilities.
- ☒ Sensitive data is handled securely, and no secrets are hardcoded.
- ☒ Proper access controls and permissions are in place.

Performance:

- ☒ The code is optimized for performance.
- ☒ Database queries are efficient, and indexes are used where necessary.
- ☒ Any potential bottlenecks are identified and addressed.

Dependencies:

- ☒ Dependencies are up-to-date, and any security vulnerabilities are mitigated.
- ☒ Third-party libraries or packages are used judiciously and for valid reasons.

## Conclusion

In conclusion, this code sets a high standard for code quality, functionality, and maintainability overall. It is ready for integration and demonstrates the developer's commitment to excellence. It is a testament to the team's collective dedication to producing top-notch software. Great work, and keep up the excellent coding practices!