**Athlete App – Coding Challenge**

The App is a collection of CRUD APIs using Spring Boot. It includes Spring Data JPA (using H2 In-memory DB) with Spring Validator and Also JUnit Jupiter (Junit 5) + Mockito + MockMVC for Unit tests.

The Coding challenge results can be accessed and downloaded from my personal Github account at this link:

<https://github.com/faisalmoin79/athlete-app/tree/coding-challenge>

I have a **main** branch with original source code and a **'coding-challenge**' branch which has my latest changes.

I started this test challenge late yesterday after work and completed today. This [link shows all the commits](https://github.com/faisalmoin79/athlete-app/commits/coding-challenge) I made to the code.

All the code changes I made can be seen in this [pull request](https://github.com/faisalmoin79/athlete-app/pull/1/files).

Some additional notes:

1. The application is a completely working app and I used POSTman to test all endpoints and error validations.
2. I used Spring Data with H2 (in memory database) for saving Entities.
   * There is a src/main/resources/data.sql which has scripts to load initial data
   * I used H2-Console to view data changes with the API

Text

Description automatically generated with low confidence

1. I added Swagger UI to the project where you can see the API endpoints and try them out. When testing i used POSTman (**by importing Swagger Collection - that is attached)**

Graphical user interface, application

Description automatically generated

1. I used Spring validator for validating beans

Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

1. I used MockedMVC for Testing Controllers and Junit5+Mockito for Service

The Unit Test coverage for this app is 83.4% with 100% for the Controller API:

A screenshot of a computer

Description automatically generated with medium confidence