**Author: Dionysios Tachtsidis**

**Title: Bulls and Cows**

**Functional description**

This program allows a user to play a bulls and cows game, and get all the games available in the database or a single game by ID.

The program uses a database to store the Game data.

The program allows the user to perform the actions through POST and GET requests through postman.

The program uses the console to display some “safety” messages.

### Prioritization

The biggest priority for that project, is to get the game logic algorithm to work.

For those purposes, at first, I wrote a bulls and cows algorithm that ran on its own (FILE INCLUDED as BnCTest).

After that, we prioritize the database communication part.

In order to achieve that, I reused some code found in the materials, and configured it to the project’s needs.

Lastly, through the controller class, I set up paths for POST and GET requests that handle the user interaction.

**KNOWN PROBLEMS**

**IF THE USER DECIDES TO INPUT VALID DATA (unique numbers in guess), THE PROGRAM WORKS AS INTENDED.**

**DATABASE PASSWORD IS LEFT BLANK**

* The tests were not running and they were giving a NullPointerException.
* I left all but one the tests within the project. The test I left was commented out and was left there to showcase by understanding of the way of the implementation of the tests part.
* Alongside that, a class next to main was added, again is commented right now, that was to handle the tests.
* The algorithm has a known issue. If the user inputs the same number more than once, the Cows counter adds it more than once as a Cow.
* Possible fix: Test user input for uniqueness. Since that happens through a post request, instead of data, the request would return an Error code.

**FILES INCLUDED**

1. Project file with my project.
2. SQL scripts for the DB and the testDB.
3. Project file for the algorithm I created.