

Assignment 1: Building a database that will store and display data about Formula 1

Barry Denby

January 27, 2023

1 Assignment Information

| | |
|------------------------|------------------------------------|
| Course: | MSCC/MSCBD |
| Stage / Year: | 1 |
| Module: | Cloud Platforms & Applications |
| Semester: | 1 |
| Assignment: | 3 of 3 |
| Date of Issue: | 2023-02-13 |
| Assignment Deadline: | 2023-03-26 @ 23:55 (End of week 6) |
| Assignment Submission: | Upload to Moodle |
| Assignment Weighting: | 10% of Module |

2 Introduction

NOTE: read the whole assignment brief first before implementing it contains very important information

In this assignment you will be tasked with building your first PaaS application which will be a database that stores information about F1 drivers and teams. You will be required to deal with users who login and logout. This will function a little like Wikipedia in that anyone who is logged in can change anything, anyone who is logged out can change nothing.

In this application users should be able to add and update drivers and teams that are in the database. They should also be able to perform comparisons of either teams or drivers. You will also be required to do some filtering. With filtering however you will be limited on the types of queries you can perform due to the NoSQL nature of the database. It is recommended that you structure your datastore models in such a way to make the required queries easier if possible.

NOTE: This is an individual assignment. It is not a group assignment. You can discuss ideas/algorithms but you cannot share code/documentation.

3 Submission and Penalties

You are required to submit two separate components to the Moodle

- An archive containing your complete Google App Engine Python project. The accepted archive formats are: zip, rar, 7z, tar.gz, tar.bz2, tar.xz. The use of any other archive format will incur a 10% penalty before grading.
- A PDF containing documentation of your code. **If you do not provide documentation your code will not be marked.** Copying and pasting code into a PDF does not count as documentation.

There are also a few penalties you should be aware of

- Code that fails to compile will incur a 30% penalty before grading. At this stage you have zero excuse to produce non compiling code. I should be able to open your project and be able to compile and run without having to fix syntax errors.
- The use of libraries outside the SDK will incur a 20% penalty before grading. You have all you need in the standard SDK. I shouldn't have to figure out how to install and use an external library to get your app to work
- **An omission of a git repository attached to your email address that is registered for GCD will result in your application and documentation not being graded.**
- The standard late penalties will also apply

You are also required to submit as part of your archive a working Git repository.

- When I unpack your archive there should be a .git directory as part of it.
- This should be a fully working **local** git archive. It should not require access to a remote repository
- You are not permitted to upload your work to Github, Gitlab, or any other publicly visible git repository (assignment will be marked as a zero if it is)
- If you need a remote git repository the only permitted one is the college provided Gitlab which can be found at gitlab.griffith.ie
- There must be a minimum of seven commits in the git repository.

Very Important: Take note of the groups listed below. These are meant to be completed in order. Groups must be completed in full before the next group will be evaluated. Completed will mean that all tasks in the groups are visible and testable. If a single one is not visible and testable further groups will not be considered. e.g. if there are four tasks in Group 1 and task 3 is skipped or not visible or testable then Groups 2, 3 and 4 will be ignored. Documentation

will be treated separately irrespective of how many Groups you have completed.

You should also be aware that I will remove marks for the presence of bugs anywhere in the code and this will incur a deduction of between 1% and 10% depending on the severity. If you have enough of these bugs it is entirely possible that you may not score very many marks overall. I want robust bug free code that also validates all user input to make sure it is sensible in nature. Please be aware of the major bugs section. If any of these bugs are present in your application you will lose 12% for each one up to a maximum of 36%

4 Plagiarism

Be aware that we take plagiarism very seriously here. Plagiarism is where you take someone else's work and submit it as if it was your own work. There are many different ways plagiarism can happen. I will list a few here (this is not exhaustive):

- Finding something similar online (full implementation or tutorial) that does the same job and submit that.
- Finding something similar online (full implementation or tutorial) and transcribing (i.e. copying it out by hand)
- Working together on an individual assignment and sharing code together such that all implementation look the same.
- Getting a copy of someone else's code and submitting/transcribing that
- Paying someone to do your assignment. **NOTE: if you are caught participating in either side of such a transaction upto 5 years after you graduate you can be stripped of your degree.**
- Logging into someone else's Moodle account, downloading their assignment and uploading it to your own Moodle account.

I've had to deal with many cases of plagiarism over the last ten years so I can spot it and diagnose it easily, so don't do it. To prevent plagiarism include but not limited to the following:

- Do all your code by yourself
- Don't share your code with anyone, particularly if anyone looks for a copy of your code for reference.
- Don't post your code publicly online. Remember the use of GitHub, Gitlab, BitBucket etc is prohibited.
- If you need to find information online only query about very specific problems you have don't look for a full assignment or howto.
- Change the default password on your Moodle account. The default password can be determined if someone is connected to you through social media or they get one or two details from you.
- If you need to refer to anything online your only permitted source to reference is StackOverflow.
- Please note that AI tools such as OpenAI or ChatGPT will count as plagiarism. Their use is strictly prohibited

Be aware that if you submit your assignment you accept that you understand what plagiarism is and that your assignment is not plagiarised in any way.

Also be aware that if you are caught for plagiarism you will not get another opportunity or a second chance to resubmit the assignment.

If you see the words "**pending review**" in your assignment feedback it is 99% likely that you will be called to a plagiarism meeting.

5 Coding Tasks (80%)

- Group 1 tasks (20%)
 1. Application that has a working login/logout service
 2. Datastore objects that will represent a driver and a team
 - Drivers should have an Age, Total Pole Positions, Total Race wins, Total points scored, Total world titles, Total Fastest Laps, and the team they drive for.
 - Teams should have Year founded, Total Pole Positions, Total Race wins, Total Constructors titles, Finishing position in previous season
 3. Separate pages for adding a driver and a team to the database
 4. Add in a form to query drivers. It should work on all attributes and should only return drivers that are either equal to or above any of the numerical values entered. It should be possible to search on one or more attributes.
- Group 2 tasks (40%)
 5. Add in a form to query teams. It should work on all attributes and should only return teams that are either equal to or above any of the numerical values entered. It should be possible to search on one or more attributes.
 6. Both query forms should be accessible to logged out users.
 7. Each driver should be shown as a hyperlink which when clicked brings the user to a separate page showing the information for that team or driver.
 8. Each team should be shown as a hyperlink which when clicked brings the user to a separate page showing the information for that team or driver.
- Group 3 tasks (60%)
 9. Add the ability to edit a driver and update their attributes
 10. Add the ability to edit a team and update their attributes
 11. add the ability to delete a driver or team from the database
 12. add in a separate page for comparing two drivers against each other
- Group 4 tasks (80%)
 13. add in a separate page for comparing two teams against each other
 14. For the comparison of drivers show a 2 column table of their stats with the better stat highlighted green. Team should not be compared. Everything except age the higher number should be highlighted. Age should be lower.
 15. For the comparison of teams show a 2 column table of their stats with the better stat highlighted green. Everything except finishing position and year the higher number should be highlighted green. For finishing position and year the lower number should be highlighted.

16. UI design: UI that is easy and intuitive to use.

- Major bugs (presence of any one of these will be a 12% reduction in mark up to a maximum of 36%)
 - It is possible to add two drivers with the same name
 - It is possible to add two teams with the same name
 - editing a driver edits the wrong driver
 - editing a team edits the wrong team
 - deleting a driver deletes the wrong driver
 - deleting a team deletes the wrong team
 - it is possible to trigger a compare with 0 or 1 drivers
 - it is possible to trigger a compare with 0 or 1 teams

6 Documentation Brackets (20%)

NOTE: Documentation should be around 1,500 words in length total

1. (0 to 15%): Document every method in your code from a high level perspective. i.e. give an overview of what the method does. Do not copy and paste code you will be penalised for this.
2. (15 to 20%): Document every datastructure and model you have used in your code and why you chose them.