

# Matt G's F1 database

# Who am I?

- Graduated from University with a BSc in Business Information technology in 2006
- Unable to properly establish an IT career
- In 2013 switched my focus to helping the autistic community as a volunteer.
- Now looking to return to the industry
- Did things in a way that was logical to me but also taking into account lessons learned from Project 1.

# Concept

- Recognised that I needed to be less rigid in my approach, but still play to my strengths as much as possible.
- Part of playing to my strengths was using a pre-existing interest of mine in Formula 1.
- Another part was realising that my strengths were more in the front end than the back end, so to get the front end fit for purpose as soon as possible - that way if things went wrong with the back end, I still had a project that wasn't completely hopeless.

# Sprint plan

- First, set up project management board.
- Then make rough first start on front end.
- Then work on the back end enough so that the API is functioning.
- Then complete front end so that it's fit for purpose.
- Set up and run back end tests.
- Work on documentation through the project rather than all in one go.
- Leave fatjar and presentation till the end.



# Consultant journey

- VS Code(HTML, CSS, Javascript)
- Eclipse(Java/Springboot)
- Jira(project management board)
- Github(Repository storage)

# CI

- Mainly stuck to main/master branch
- Had to merge master with main branch.
- Briefly set up a “calendar test” branch to experiment with some Javascript stuff - eventually deleted that branch though

# Testing

- Not quite 80% but a lot better than my first project.

The screenshot shows the Eclipse IDE interface for a project named 'F1-database'. The top toolbar includes menus like File, Edit, Navigate, Search, Project, Run, Window, and Help. Below the toolbar, the Package Explorer on the left shows the project structure. The main editor displays the 'pom.xml' file, which is a Maven POM for a Spring Boot application. The XML content includes project details like groupId, artifactId, version, and a list of dependencies. The bottom of the IDE features a 'Failure Trace' panel and a 'Coverage' table.

Test Results Summary:

Test Case	Runner	Time (s)
F1DatabaseApplicationTest	JUnit 5	0.161
F1DatabaseControllerIntegrationTest	JUnit 5	0.568
F1DatabaseControllerUnitTest	JUnit 5	0.103
F1DatabaseServiceTest	JUnit 5	0.019

Overall Test Summary:

Runs	Errors	Failures
25/25 (4 skipped)	0	0

Failure Trace:

Failure Trace is empty.

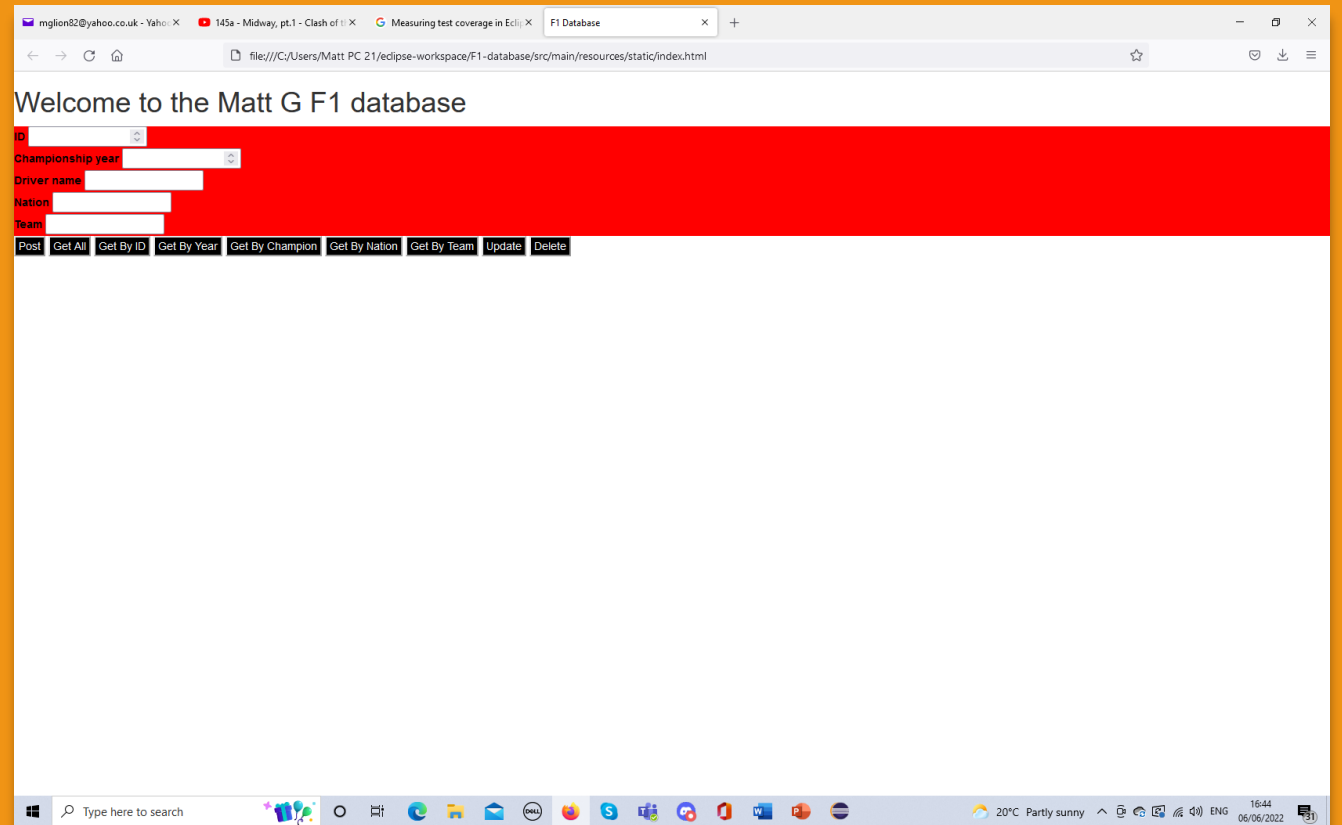
Coverage Table:

Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
F1-database	75.4 %	1,110	362	1,472

The pom.xml content is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.0</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.qa</groupId>
  <artifactId>F1-database</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>F1-database</name>
  <description>Week 8 project</description>
  <properties>
    <java.version>11</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-validation</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
  </dependencies>
</project>
```

# Demonstration Introduction



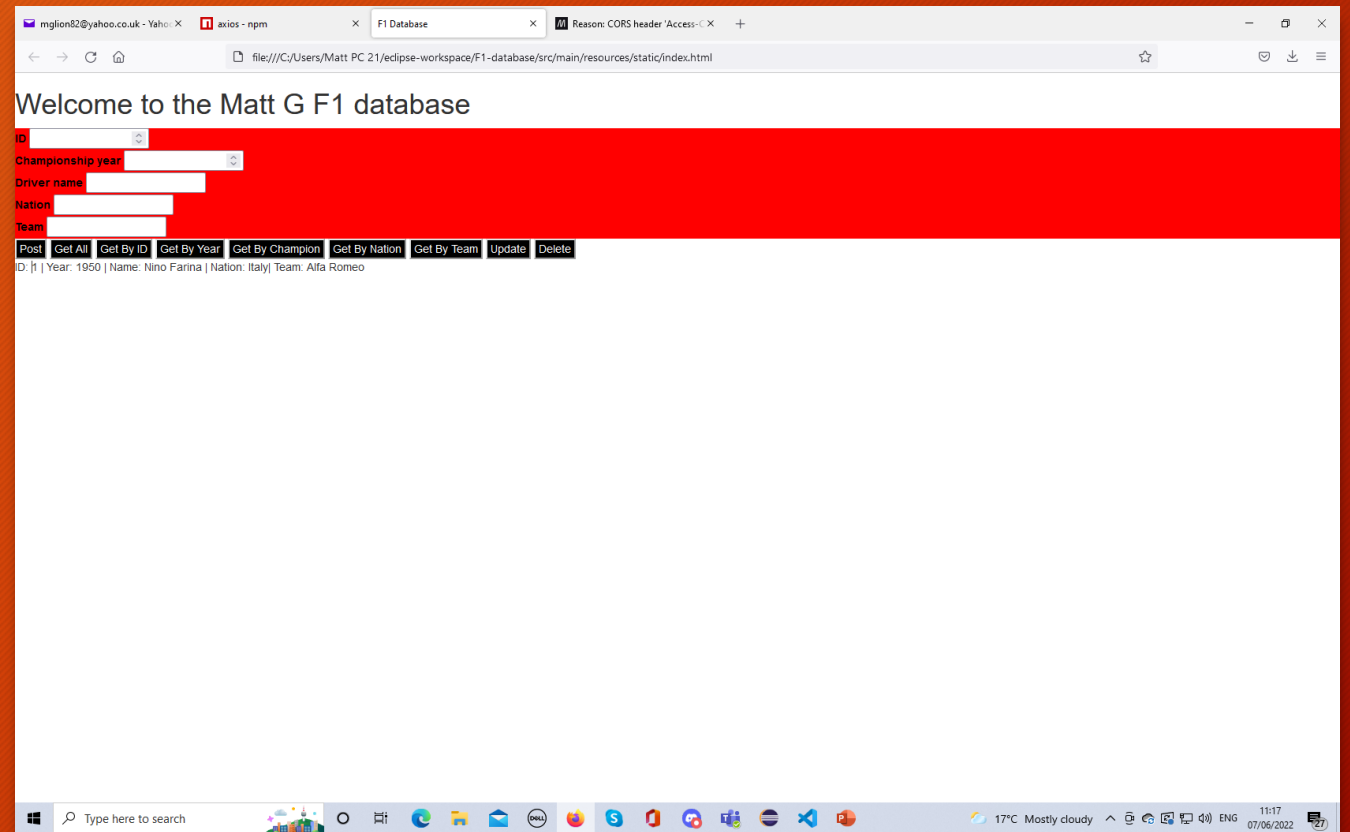


# Front end features

- Buttons for all potential operations.
- Input fields for all potential values that the user may wish to enter or search for.
- Included ID field because even though that won't be on public display - this project wasn't made with a regular user in mind, rather an administrator of a database who is going to need to be able to search for entries by ID.

# User story

- User enters an entry relating to the 1950 World champion.
- User presses “Get All” and the entry is displayed.



# Sprint review

- Completed back end, HTML and CSS
- Completed fatjar, ERD, project management board and this presentation.
- Last minute Javascript glitches I wasn't able to resolve in time.
- Wanted to include year picker and display flags - found Bootstrap classes for these features but couldn't get them implanted in time.

# Sprint retrospective

- Due to last minute issues still a failure but an improvement on project 1
- Could have double checked Javascript earlier - a lot of the issues I did manage to sort at the last minute I should have spotted earlier.



# Conclusion

- Re-established my strengths and weaknesses.
- Don't get complacent about anything.
- Still put a lot of effort in.
- Needed more help than expected

# Questions?

