Getting the code

If you have git, you can do “git clone <https://github.com/aelxw/mlse.git>” to get the code for this application.

Python Information

* The application should work with version 3.6.2+
  + I initially was using version 3.6.2, but upgraded to 3.7.1 and it still worked fine
  + I just downloaded Anaconda through this link: <https://repo.anaconda.com/archive/Anaconda3-2018.12-Windows-x86_64.exe>
* The required packages are in **/server/requirements.txt**
  + “pip install -r requirements.txt” should import the packages from this file.
  + If this doesn’t work, the packages may need to be installed individually with “pip install …”
* The two main files for the backend are: **REST\_API.py** and **Models.py**
  + REST\_API.py runs a WSGI server that takes REST calls
  + Objects from Models.py are imported by REST\_API.py and used for matching
  + You can look at these files to see the backend logic of the application

Front end

* Only two files are used for the front end: **/client/code.gs** and **/client/index.html**
* Log in with a designated google account and go to <https://script.google.com/home> to create a google apps script project. Paste the code from these two files into the project (these files are not local)
* **code.gs** contains the google functions that creates the Google form, Spreadsheet, etc.
* **index.html** is a single file with the HTML and JavaScript code for the application.
  + It is using AngularJS
* Go to <https://script.google.com/d/1qg9g0zGXTTsqZour6fZO75ZWY1JQJcdTgYJZMwe78flsg41VzIFzNOz9/edit?usp=sharing> to see how it’s currently set up
* To deploy, click “Publish” -> “Deploy as web app…” -> “Update” -> click the “latest code” link or go to the “Current web app URL”
* Make sure that the server is running for the application to work correctly
  + Run/double click **/server/run\_REST\_API.bat**

Configuration

* /server/run\_REST\_API.bat needs to be changed according to where python is located on the local computer.
  + Change “C:\Users\Alexander\AppData\Local\Anaconda3\python.exe” to the location of python.exe
  + Change “C:\Users\Alexander\mlse\server\REST\_API.py” to the location of the REST\_API.py file
* Running this file should start the server

Hosting

* There are **two** locations where the hostname and port must be specified (server side and client side):
  + Server side: **/server/REST\_API.py line 60**: “http\_server = WSGIServer((**'localhost'**, **2000**), app)”
    - Change **‘localhost’** to the necessary hostname
    - Change **2000** to the necessary port
  + Client side: **/client/index.html line 331**: “const server = "**http://localhost:2000**";”
    - Change **http://localhost:2000** to the necessary socket address

Database

* Everything is in the /server/database folder.
  + I’m not sure if what’s in the folder will work by default (it might). However, if there are errors, download <https://sqlite.org/2019/sqlite-tools-win32-x86-3270100.zip>, extract it, and run sqlite3.exe. Make sure that the files are copied to the /server/database folder
* The SQLite database is the **mlse.db** file
* I use VSCode to view/query the database. The installation should have a Browser.exe file to view the database as well.
* There are two tables in the SQLite database
  + Team
    - Name (team name)
    - Logo (url link to a picture of the team icon)
    - Division (NHL/NBA)
  + Prev (previous rank for each employee) – **irrelevant now because we aren’t using historical data**
    - Email (email of employee)
    - Rank (rank employee got in the last matching round)
    - Division (NHL/NBA)