



408-Capstone

By: Prem Patel



Table Of Contents

- FrontEnd/BackEnd
- Database
- Static Analysis
- Dynamic Analysis
- Memory Protection / Mitigation
- Secure Environment Configuration
- Documentation & Research

FrontEnd

Home Search Players Login Register

Search Players

 Search

Search Players

Baker Mayfield Search

Results for "Baker Mayfield" (Live API)



Baker Mayfield — Quarterback (Tampa Bay Buccaneers)

Data from TheSportsDB API



Jalen Mayfield — Guard (Atlanta Falcons)

Data from TheSportsDB API

Baker Mayfield



Sport / League: American Football

Team: Tampa Bay Buccaneers

Position: Quarterback

Jersey #: 6

Nationality: United States

Birth Date: 1995-04-14

Birth Place: Austin, Texas

Height: 6 ft 1 in (1.85 m)

Weight: 215 lb (98 kg)

College: N/A

Career Stats (Pro-Football-Reference)

Seasons	8
Games Played	120
Passing Yards	27554
Passing TDs	191
Interceptions	96
Rushing Attempts	354
Rushing Yards	1486
Rushing TDs	11

Career Summary

Baker Reagan Mayfield (born April 14, 1995) is an American football quarterback for the Cleveland Browns of the National Football League (NFL). Mayfield began his college football career as a walk-on player for the Texas Tech Red Raiders. He is notable for being the first walk-on true freshman quarterback to start a season opener at a BCS school.

Front-End: HTML and CSS

BackEnd

Back-End: Flask and Python

- Run.py
- Routes definitions with routes.py
- Configurations with config.py
- Display stats with models.py

```
import datetime
from flask import Blueprint, render_template, request, current_app, abort
from . import db
from .models import SearchLog
import requests

main_bp = Blueprint("main", __name__)

# ----- TheSportsDB helpers (for search + profile) -----
def search_players_api(name: str):
    """Search players via TheSportsDB (API id, name, position, team, thumb)."""
    key = current_app.config.get("THESPORTSDB_API_KEY")
    base = current_app.config.get("THESPORTSDB_BASE_URL")
    url = f"{base}/{key}/searchplayers.php"

    try:
        resp = requests.get(url, params={"p": name}, timeout=5)
        resp.raise_for_status()
    except requests.RequestException as e:
        current_app.logger.error("TheSportsDB API error: %s", e)
        return []

    data = resp.json() or {}
    players = data.get("player") or []
    for player in players:
        player["date"] = datetime.datetime.fromtimestamp(player["date"])
    return players
```

Database

I originally had a database where career stats of very prominent players for every single position, but I was told to use an API for the career stats because that way if an user types in non-prominent player, the website can still show the user the stats of that player

```
1  ons.TemplateNotFound: player.html
2  10:06,308 [INFO] 127.0.0.1 Search query: Tom Brady
3  10:19,262 [INFO] 127.0.0.1 Search query: Baker Mayfield
4  25:44,365 [INFO] 127.0.0.1 Search query: Baker Mayfield
5  25:48,186 [INFO] 127.0.0.1 Search query: Baker Mayfield
6  25:54,313 [INFO] 127.0.0.1 Search query: Odell
7  26:30,420 [INFO] 127.0.0.1 Search query: Mike Evans
8  26:32,157 [INFO] 127.0.0.1 Search query: Mike Evans
9  26:42,473 [INFO] 127.0.0.1 Search query: Tom Brady
10 26:53,586 [INFO] 127.0.0.1 Search query: Drew Brees
11 32:51,068 [INFO] 127.0.0.1 Search query: Odell
12 33:04,167 [INFO] 127.0.0.1 Search query: tyreek hill
13 33:11,821 [INFO] 127.0.0.1 Search query: Jerry Rice
14 33:23,378 [INFO] 127.0.0.1 Search query: Randy Moss
15 34:52,294 [INFO] 127.0.0.1 Search query: Drew Brees
16 34:55,159 [INFO] 127.0.0.1 Search query: Drew Brees
17 39:15,175 [INFO] 127.0.0.1 Search query: Baker Mayfield
18 39:32,937 [INFO] 127.0.0.1 Search query: Odell
19 15:29,334 [INFO] 127.0.0.1 Search query: Baker Mayfield
20 15:46,832 [INFO] 127.0.0.1 Search query: Odell
21 15:57,828 [INFO] 127.0.0.1 Search query: Nick Chubb
22 44:34,647 [INFO] 127.0.0.1 Search query: Nick Chubb
```

Static Analysis

- I used Bandit on my Static Analysis

```
(venv) premallpatell@DESKTOP-IHAIMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ bandit -r app -f html -o bandit_report.html
[main] INFO profile include tests: None
[main] INFO profile exclude tests: None
[main] INFO cli include tests: None
[main] INFO cli exclude tests: None
[main] INFO running on Python 3.12.3
[html] INFO HTML output written to file: bandit_report.html
(venv) premallpatell@DESKTOP-IHAIMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$
```

Category	Issues
Critical	None
High Severity	None
Moderate Severity	Timeout missing on requests
Low Severity	HTML parsing, safe usage
Hardcoded Secret	None
External Library Issues	None

Dynamic Analysis

- For Dynamic Analysis I used strace, tcpdump, cpdump, py-spy, and mprof to do the Dynamic Analysis
 - Strace: System calls
 - Tcpdump: network traffic
 - Py-spy: Run-time Performance

```
y-spy record -o profile.svg --pid 3305
```

Memory Protection / Mitigation

ASLR and DEP/NX protections were enabled because the website can not operate if the system is air-gapped because of the APIs used from SportsDB and Pro-Football-Reference

```
(venv) premallpatell@DESKTOP-IHAIMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ python3 -m venv .venv
(venv) premallpatell@DESKTOP-IHAIMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ source venv/bin/activate
(venv) premallpatell@DESKTOP-IHAIMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ python3 run.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.26.104.200:5000
Press CTRL+C to quit
^C(venv) premallpatell@DESKTOP-IHAIMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ cat /proc/sys/kernel/randomize_va_space
2
(venv) premallpatell@DESKTOP-IHAIMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ readelf -l /usr/bin/python3 | grep GNU_STACK
GNU_STACK      0x0000000000000000 0x0000000000000000 0x0000000000000000 *
```

Secure Environment Configuration

The webapp was ran in an isolated environment using WSL with its own python virtual environment.

```
premallpatell@DESKTOP-IHAHM19:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ python3 -m venv .venv
premallpatell@DESKTOP-IHAHM19:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ source .venv/bin/activate
(venv) premallpatell@DESKTOP-IHAHM19:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ python3 run.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://172.26.104.200:5000
Press CTRL+C to quit
*
```



Challenged Faced

The challenges that I faced was trying to figure which option to for the project because it took me a while that I only needed to focus on the core components and the options given to us by Professor Tibs were just suggestions and not actual options that we had to do.



Lesson Learned

The biggest thing I learned during this final project was implementing API keys to the website from SportsDB and Pro-Football-Reference because it did not work at start and it took me a long time to figure out the issue and apply the solutions to the code . It took me about 3 hours for the APIs to work. Another lesson I learned that if you going to use an API, just pay for it because it will save you so much time and hassle.



Sources

APIs - SportsDB and Pro-Football-Reference

AI Source - Chat-GPT 5.1

The AI tool(s) used were Chat-GPT. This tool was used long-code changes and Ideas. Key prompts included (edit this to simple for the user to use). The AI output was evaluated by me working the website deciding what I need to add or change. My edits to the AI's output involved the core components of the final project.]



Class Remarks

This was my favorite class by far this semester and huge thanks to Professor Tibs for making this a fun and exciting class, this is the class where I learned the most out of all the class I have taken here at The Citadel.

The End

