

A decorative graphic on the left side of the slide consists of two overlapping parallelograms. The front one is blue and the back one is a light mint green. They are positioned diagonally, with the blue one partially covering the green one.

# 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

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 []
```

# 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 a user types in a non-prominent player, the website can still show the user the stats of that player

```
41 ons.TemplateNotFound: player.html
42 10:06,308 [INFO] 127.0.0.1 Search query: Tom Brady
43 10:19,262 [INFO] 127.0.0.1 Search query: Baker Mayfield
44 25:44,365 [INFO] 127.0.0.1 Search query: Baker Mayfield
45 25:48,186 [INFO] 127.0.0.1 Search query: Baker Mayfield
46 25:54,313 [INFO] 127.0.0.1 Search query: Odell
47 26:30,420 [INFO] 127.0.0.1 Search query: Mike Evans
48 26:32,157 [INFO] 127.0.0.1 Search query: Mike Evans
49 26:42,473 [INFO] 127.0.0.1 Search query: Tom Brady
50 26:53,586 [INFO] 127.0.0.1 Search query: Drew Brees
51 32:51,068 [INFO] 127.0.0.1 Search query: Odell
52 33:04,167 [INFO] 127.0.0.1 Search query: tyreek hill
53 33:11,821 [INFO] 127.0.0.1 Search query: Jerry Rice
54 33:23,378 [INFO] 127.0.0.1 Search query: Randy Moss
55 34:52,294 [INFO] 127.0.0.1 Search query: Drew Brees
56 34:55,159 [INFO] 127.0.0.1 Search query: Drew Brees
57 39:15,175 [INFO] 127.0.0.1 Search query: Baker Mayfield
58 39:32,937 [INFO] 127.0.0.1 Search query: Odell
59 15:29,334 [INFO] 127.0.0.1 Search query: Baker Mayfield
60 15:46,832 [INFO] 127.0.0.1 Search query: Odell
61 15:57,828 [INFO] 127.0.0.1 Search query: Nick Chubb
62 44:34,647 [INFO] 127.0.0.1 Search query: Nick Chubb
```

```
from app import create_app, db
from app.models import Player
```

```
def seed_players():
    app = create_app()
```

```
    with app.app_context():
        db.drop_all()
        db.create_all()
```

```
players = [
    # QBs
    Player(full_name="Tom Brady", position="QB", team="NE/TB", seasons=22),
    Player(full_name="Peyton Manning", position="QB", team="IND/DEN", seasons=20),
    Player(full_name="Drew Brees", position="QB", team="SD/NO", seasons=10),
    Player(full_name="Aaron Rodgers", position="QB", team="GB/NYJ", seasons=10),
    Player(full_name="Brett Favre", position="QB", team="ATL/GB/NYJ/MIN", seasons=20),
    Player(full_name="Patrick Mahomes", position="QB", team="KC", seasons=6),
    Player(full_name="Josh Allen", position="QB", team="BUF", seasons=6),
    Player(full_name="Joe Burrow", position="QB", team="CIN", seasons=4),
    Player(full_name="Matt Ryan", position="QB", team="ATL/IND", seasons=8),
    Player(full_name="Matthew Stafford", position="QB", team="DET/LAR", seasons=11),
    Player(full_name="Kirk Cousins", position="QB", team="WAS/MIN/ATL", seasons=7),
    Player(full_name="Baker Mayfield", position="QB", team="CLE/CAR/LAR/TI", seasons=5),
    Player(full_name="Jalen Hurts", position="QB", team="PHI", seasons=4),
    Player(full_name="Lamar Jackson", position="QB", team="BAL", seasons=3),

    # RBs
```

# Static Analysis

- I used Bandit on my Static Analysis

```
(venv) premallpatell@DESKTOP-IAHMI9:/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-IAHMI9:/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



- [illegible]

```
py-spy record -o profile.svg --pid 3985
```

```
all
__bootstrap (threading.py:1038)
__bootstrap_inner (threading.py:1073)
run (threading.py:1048)
process_request_thread (socketserver.py:692)
finish_request (socketserver.py:362)
__init__ (socketserver.py:701)
handle (werkzeug.serving.py:391)
handle (http.server.py:436)
handle_one_request (http.server.py:424)
run_wsgi (werkzeug.serving.py:363)
execute (werkzeug.serving.py:324)
__call__ (flask/app.py:1498)
wsgi_app (flask/app.py:1473)
full_dispatch_request (flask/app.py:880)
dispatch_request (flask/app.py:865)
player_detail (app/routes.py:211)
get_player_details (app/routes.py:231)
get_pr_carcer_stats (app/routes.py:119)
reference_web_scraper/player_game_log.py:69          reference_web_scraper/player_game_log.py:70      +S-py:134
request (requests/api.py:73)                        feed (b45/init.py:476)                          +S-py:135
request (requests/api.py:59)                         feed (b45/init.py:661)                            +S-py:9
request (requests/sessions.py:589)                   feed (b45/builder/htmlparser.py:455)              +S-py:44
send (requests/sessions.py:703)                      feed (html/parser.py:112)                         +S-py:136
send (requests/adapters.py:644)                     feed (html/parser.py:172) ..ser.py:174           +S-py:137
open (urllib3/connectionpool.py:787)                 ..rser.py:357) ..r.py:433                       +S-py:138
connect (urllib3/connectionpool.py:404)               ..r.py:382) ..r.py:298                          +S-py:9
open (urllib3/connectionpool.py:393)                  ..r.py:893) ..r.py:1065                           +S-py:139
connect (urllib3/connection.py:812)                  ..r.py:812) ..r.py:1065                           +S-py:44
..stname (urllib3/connection.py:975)                ..r.py:812) ..r.py:1065                           +S-py:155
ocket (urllib3/util/ssl.py:461)                    ..r.py:812) ..r.py:1065                           +S-py:1360
                                                    ..r.py:812) ..r.py:1065                           +S-py:1381
                                                    ..r.py:812) ..r.py:1065                           +S-py:935
```



# 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-IHAHMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ python3 -m venv .venv
(venv) premallpatell@DESKTOP-IHAHMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ source venv/bin/activate
(venv) premallpatell@DESKTOP-IHAHMI9:/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-IHAHMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ cat /proc/sys/kernel/randomize_va_space
2
(venv) premallpatell@DESKTOP-IHAHMI9:/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-IAHMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ python3 -m venv .venv
premallpatell@DESKTOP-IAHMI9:/mnt/c/Users/prema/Documents/CSCI-408/Nfl-Capstone$ source venv/bin/activate
(venv) premallpatell@DESKTOP-IAHMI9:/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

