Team 1:

Uveimar Sandoval: usandova@gmu.edu

Farith Bascope: fbascope@gmu.edu

Daniel Curtis: dvurtis7@gmu.edu

Rajeep Karki: rkarki6@gmu.edu

Sammy Miller: smille77@gmu.edu

Omar Naseem: [onaseem@gmu.edu](mailto:onaseem@gmu.edu)

Joshua Reyes: jreyes36@gmu.edu

December 5, 2023

CS 321: Software Engineering

Professor Wes Masri

GTA: Bhargavi Janga

Project Deliverable 4 – Testing

**Technology Used**

Testing Framework: pytest

* Testing framework for Python that simplifies the process of writing and executing tests.
* Provides clean syntax, fixtures, and extensive support for test discovery and automation.

**Unit Testing:**

* test\_first()
  + This method is for displaying the first player's name from the NFL database.
  + Gets player’s name using the SQL query: "SELECT Name FROM NFL."
  + Expected "Josh Allen."
* test\_second()
  + Method to display a list of players and verify the index of "Kyle Allen."
  + To get a list of player names using the SQL query: "SELECT Name FROM NFL."
  + Checks if "Kyle Allen" is at index 1 in the result list.
* test\_many()
  + Method to display list of players and verify the index of "Josh Allen" and "Shane Buechele."
  + Gets a list of player names using the SQL query: "SELECT Name FROM NFL."
  + Checks if "Josh Allen" is at index 0 and "Shane Buechele" is at index 2 in the result list.
* test\_defense()
  + This method corresponds to displaying a list of players and verifying the player at index 35.
  + It retrieves a list of player names using the SQL query: "SELECT Name FROM NFL."
  + It checks if the player at index 35 is "Kameron Cline."
* test\_specialTeams()
  + This method is to displaying a list of players and verifying the player at index 70.
  + Gets list of player names using the SQL query: "SELECT Name FROM NFL."
  + Checks if the player at index 70 is "Tyler Bass."

The following test cases confirm extract\_RosterData() is functional with multiple NFL teams.

* test\_teamMIA()
  + - Method to test the data extraction for the Miami Dolphins team.
    - It calls the **extract\_RosterData** function with the URL for the Miami Dolphins' roster.
    - It then asserts that the first item in the extracted data list matches the expected value: "Tua Tagovailoa QB 25 6' 1"
* test\_teamMIADef()
  + - Method corresponds to testing the data extraction for the Miami Dolphins' defense.
    - It calls the **extract\_RosterData** function with the URL for the Miami Dolphins' roster.
    - It asserts that the 41st item in the extracted data list matches the expected value: "Emmanuel Ogbah DE 29 6' 4".
* test\_teamNE(), test\_teamNEDef(), test\_teamNYJ(), test\_teamNYJDef, test\_teamBAL(), test\_teamBALDef()
  + - Methods to test data extraction for other NFL teams (e.g., New England Patriots, New York Jets, Baltimore Ravens) in a similar manner to above methods like **test\_teamMIA** and **test\_teamMIADef**.
    - These methods test different teams and positions by calling the **extract\_RosterData** function with the appropriate URLs and asserting the extracted data.
* test\_teamCIN(), test\_teamCINDef()
  + - Methods to test data extraction for the Cincinnati Bengals and their defense.
    - They follow the same pattern as the other test methods by calling the **extract\_RosterData** function with the respective URLs and asserting the extracted data.
* test\_playerOne()
  + Method to retrieve a players stats from the website
  + Tests that the received data is in the format
  + [James Cook, 507, 1, 21]
* test\_playerTwo()
  + Method to retrieve players stats from the website
  + Tests that the received data is in the format
  + [Damien Harris, 462, 0, 17]
* test\_playerThree()
  + Method to retrieve players stats from the website
  + Tests that the received data is in the format
  + [Ty Johnson, 160, 2, 12]
* test\_position()
  + Method to display a player's position based on the SQL query: "SELECT Name, Position FROM NFL."
  + Gets the player's name and position.
  + Checks if the position is "QB" and asserts True or False.
* test\_QB()
  + Method to display a list of players with the position "QB."
  + Gets player names using the SQL query: "SELECT Name FROM NFL WHERE position ='QB'."
  + Checks if the list of players with the "QB" position matches the expected list: ["Josh Allen", "Kyle Allen", "Shane Buechele"].
* test\_Josh()
  + Method that will retrieve the data that is stored on Josh Allen
  + Querys’ the database using the SQL Query “SELECT \* FROM Stats WHERE name=”Josh Allen”
  + Asserts that the data is [762, 3, 0]
* test\_Kyle()
  + Method that will retrieve the data that is stored on Josh Allen
  + Querys’ the database using the SQL Query “SELECT \* FROM Stats WHERE name=”Kyle Allen”
  + Asserts that the data is [13, 0, 0]
* test\_StatsArizona()
  + Method to verify roster and statistics data were inserted correctly from [www.espn.com](http://www.espn.com)
  + Runs insertRoster\_DB and runs SQL query "SELECT PassingYards, RushingYards FROM NFL WHERE Name = Joshua Dobbs"
  + checks if roster and statistics are correct for specific players
* The following have similar behavior to test\_StatsArizona() but for different teams:
  + test\_StatsRavens
  + test\_StatsFalcons
  + test\_StatsBills
  + test\_StatsPanthers
  + test\_StatsBengals
  + test\_StatsBears
  + test\_StatsBrowns
  + test\_StatsCowboys
  + test\_StatsBroncos
  + test\_StatsLions
  + test\_StatsTexans
  + test\_StatsPackers
  + test\_StatsColts
  + test\_StatsRams
  + test\_StatsJaguars
  + test\_StatsVikings
  + test\_StatsChiefs
  + test\_StatsSaints
  + test\_StatsRaiders
  + test\_StatsGiants
  + test\_StatsChargers
  + test\_StatsEagles
  + test\_StatsDolphins
  + test\_Stats49ers
  + test\_StatsPatriots
  + test\_StatsSeahawks
  + test\_StatsJets
  + test\_StatsBuccaneers
  + test\_StatsSteelers
  + test\_StatsCommanders
  + test\_StatsTitans

**Integration Testing:**

* test\_database\_integration()
  + test\_database\_integration() tests the integration of the database.py file with the parser.py file.
  + Functions tested: insertRoster\_DB(), main(), extract\_RosterData()
  + Checks if roster and statistic data are inserted correctly by checking if certain names of players are in the database.
* test\_parser\_integration1()
  + test\_parser\_integration() tests the integration of the parser.py file methods
  + gets results by running the extract\_RosterData() method which also runs the extract\_source() method
  + Checks if correct roster data is in the result depending on the URL given to extract\_RosterData
* The following have similar behavior to test\_parser\_integration1() with different URLs
  + test\_parser\_integration2()
  + test\_parser\_integration3()
  + test\_parser\_integration4()
* test\_parser2\_integration1()
  + test\_parser2\_integration() tests the integration of the parser2.py file methods
  + gets results by running the the init() method in parser2.py, which runs extract\_positions() and scrape\_table()
  + Checks if the result is a list and has a length > 0
* The following have similar behavior to test\_parser2\_integration1() with different URLs
  + test\_parser2\_integration2()
  + test\_parser2\_integration3()
  + test\_parser2\_integration4()

**System Testing:**

* System Test 1 - Scenario: View all players
  + Test to see if user can see a list of all players
  + User has the option to see a player's stats
  + Checks if the list is not empty
* System Test 2- Scenario: View player stats
  + Test to see if user can see a player's stats
  + Checks if the player's stats are not empty
* System Test 3 - Scenario: Filter by Name
  + Test to see if user can filter by name and get the player with that name
  + User can see the player's stats
  + Checks if the player matches the name
* System Test 4 - Scenario: Filter by position
  + Test to see if user can filter by position and get a list of players that play that position
  + User has the option to see a player's stats
  + Checks if the players match the position
* System Test 5 - Scenario: Filter by age
  + Test to see if user can filter by age and get a list of players that are that age
  + User has the option to see a player's stats
  + Checks if the players match the age
* System Test 6 - Scenario: Filter by height
  + Test to see if user can filter by height and get a list of players that are that height
  + User has the option to see a player's stats
  + Checks if the players match the height

**Contributors:**

* Sammy Miller
* Joshua Reyes