



Reece Doyle

T1A3 – TERMINAL APPLICATION



- ▶ Lake Macquarie Premier League App

Planning and Implementation



Planning and Implementation

The image shows a Trello board titled "Lake Macquarie Premier League". The board has four columns: "User Stories", "Features", "Doing", and "Done".

- User Stories:**
 - As an Administrator of LMPL, I want a simple app that will store the entire season draw for our competition. I want a simple method of storing results in a CSV file that I can use to update the team ladder in excel.
 - As a Player, I want a simple method to see where my team is playing for the season so I can effectively plan my week around where I need to be for the games.
 - As a fan, I want to know which teams have made the cut, and be able to see quickly what games are being played for the Round, so I can see where I need to be to watch the games.
- Features:**
 - + Add a card
- Doing:**
 - + Add a card
- Done:**
 1. TEAMS: Print out the Teams that have made the competition, displayed in team colours.
 2. GAMES THIS WEEK: See the games happening within the upcoming week.
 3. ROUNDS: See the games for any of the 14 round season.
 4. TEAM DRAW: All 14 rounds of games for a particular team
 5. ENTER GAME RESULTS: Into a CSV file
 6. REMOVE RESULTS: If a mistake is made while entering results, remove them based on the Game number (1-56)
 7. VIEW RESULTS IN TABLE: Displays results.csv in a pretty table!

Features

The features of my app include:

- Home Screen: Where you navigate to all the other features. It generates a results.csv file. There's a large ASCII logo.
- Print Teams: Prints a list of teams participating in the competition.
- Games this week: Prints the games occurring in the calendar week.
- Print Draw of a Team: Print only the games for a particular team.
- Enter Results: Enter the results of the games in the results.csv.
- Remove Results: Stores the results you want to remove in a list and overwrites the results.csv with the rows you selected removed.
- View Results: Prints the results.csv in a pretty table!
- Exit Screen: A Farewell and another ASCII logo.

Lake Macquarie Premier League App

- A command line application designed to help with the administration tasks of a hypothetical amateur football league in my hometown of Lake Macquarie.
- I prompted ChatGPT to generate the team names, the draw, and the rounds. I created many versions of csv files before I settled on what I needed for the app

Lake Macquarie Premier League App

- A command line application designed to help with the administration tasks of a hypothetical amateur football league in my hometown of Lake Macquarie.
- I prompted ChatGPT to generate the team names, the draw, and the rounds. It create many versions of csv files before I settled on what I needed for the app



You

Can you please come up with the names for 8 football teams. They're based around suburbs of Lake Macquarie NSW Australia. I want the team mascots to all have alliteration e.g. Warners Bay Wanderers



ChatGPT

Certainly! Here are eight football teams based around suburbs of Lake Macquarie, NSW, Australia, with alliterative mascots:

1. Belmont Buccaneers
2. Charlestown Challengers
3. Eelebana Eagles
4. Glendale Guardians
5. Mount Hutton Hurricanes
6. Speers Point Spartans
7. Swansea Swans
8. Tingira Tigers



You

Add Boolaroo and Warners Bay to the list



ChatGPT

Certainly! Here's an updated list with Boolaroo and Warners Bay included:

1. Belmont Buccaneers
2. Boolaroo Bulldogs
3. Charlestown Challengers
4. Eelebana Eagles
5. Glendale Guardians
6. Mount Hutton Hurricanes
7. Speers Point Spartans
8. Swansea Swans
9. Tingira Tigers
10. Warners Bay Wanderers

Lake Macquarie Premier League App

- A command line application designed to help with the administration tasks of a hypothetical amateur football league in my hometown of Lake Macquarie.
- I prompted ChatGPT to generate the team names, the draw, and the rounds. It create many versions of csv files before I settled on what I needed for the app

please append this to csv

 ChatGPT
Certainly! Here is the appended data for Rounds 5 to 7, formatted into a CSV file:

CSV	Copy code
Round,Game Number,Home Team,Away Team,Day of the Week,Date,Time,Location 5,17,Eleebana Eagles,Speers Point Spartans,Saturday,2 Dec 23,15:00,Eleebana 5,18,Eleebana Eagles,Swansea Silverbacks,Saturday,9 Dec 23,15:00,Eleebana 5,19,Speers Point Spartans,Warners Bay Wanderers,Saturday,2 Dec 23,15:00 5,20,Boolaroo Bulldogs,Warners Bay Wanderers,Saturday,9 Dec 23,11:00,Boolaroo 6,21,Boolaroo Bulldogs,Warners Bay Wanderers,Saturday,9 Dec 23,11:00,Boolaroo 6,22,Glendale Guardians,Charlestown Cobras,Sunday,11 Feb 24,12:30,Glendale 6,23,Eleebana Eagles,Warners Bay Wanderers,Sunday,7 Jan 24,12:30,Eleebana 6,24,Charlestown Cobras,Speers Point Spartans,Sunday,10 Dec 23,12:30,Charlestown 7,25,Belmont Bandits,Charlestown Cobras,Friday,19 Jan 24,18:00,Belmont 7,26,Glendale Guardians,Eleebana Eagles,Sunday,17 Dec 23,12:00,Glendale 7,27,Charlestown Cobras,Warners Bay Wanderers,Saturday,16 Dec 23,13:30,Charlestown 7,28,Swansea Silverbacks,Eleebana Eagles,Saturday,13 Jan 24,15:00,Swansea	 Copy code

Feel free to use this appended CSV format for your needs. If you have any more requests or if there's anything else I can assist you with, feel free to let me know!

 ChatGPT
Feel free to use this appended CSV format for your needs. If you have any more requests or if there's anything else I can assist you with, feel free to let me know!

 You
please merge the last two csv files

 ChatGPT
Certainly! Here is the merged data for Rounds 1 to 7, formatted into a single CSV file:

CSV	Copy code
Round,Game Number,Home Team,Away Team,Day of the Week,Date,Time,Location 1,1,Belmont Bandits,Boolaroo Bulldogs,Friday,3 Nov 23,18:00,Belmont Park 1,2,Charlestown Cobras,Eleebana Eagles,Saturday,4 Nov 23,11:00,Charlestown 1,3,Boolaroo Bulldogs,Charlestown Cobras,Saturday,4 Nov 23,11:00,Charlestown 1,4,Charlestown Cobras,Boolaroo Bulldogs,Saturday,4 Nov 23,11:00,Charlestown 1,5,Eleebana Eagles,Speers Point Spartans,Saturday,4 Nov 23,11:00,Eleebana 1,6,Speers Point Spartans,Eleebana Eagles,Saturday,4 Nov 23,11:00,Eleebana 1,7,Boolaroo Bulldogs,Charlestown Cobras,Saturday,4 Nov 23,11:00,Charlestown 1,8,Charlestown Cobras,Boolaroo Bulldogs,Saturday,4 Nov 23,11:00,Charlestown 1,9,Eleebana Eagles,Speers Point Spartans,Saturday,4 Nov 23,11:00,Eleebana 1,10,Speers Point Spartans,Eleebana Eagles,Saturday,4 Nov 23,11:00,Eleebana 1,11,Boolaroo Bulldogs,Charlestown Cobras,Saturday,4 Nov 23,11:00,Charlestown 1,12,Charlestown Cobras,Boolaroo Bulldogs,Saturday,4 Nov 23,11:00,Charlestown 1,13,Eleebana Eagles,Speers Point Spartans,Saturday,4 Nov 23,11:00,Eleebana 1,14,Speers Point Spartans,Eleebana Eagles,Saturday,4 Nov 23,11:00,Eleebana 1,15,Boolaroo Bulldogs,Charlestown Cobras,Saturday,4 Nov 23,11:00,Charlestown 1,16,Charlestown Cobras,Boolaroo Bulldogs,Saturday,4 Nov 23,11:00,Charlestown 1,17,Eleebana Eagles,Speers Point Spartans,Saturday,4 Nov 23,11:00,Eleebana 1,18,Speers Point Spartans,Eleebana Eagles,Saturday,4 Nov 23,11:00,Eleebana 1,19,Boolaroo Bulldogs,Charlestown Cobras,Saturday,4 Nov 23,11:00,Charlestown 1,20,Charlestown Cobras,Boolaroo Bulldogs,Saturday,4 Nov 23,11:00,Charlestown 1,21,Eleebana Eagles,Speers Point Spartans,Saturday,4 Nov 23,11:00,Eleebana 1,22,Speers Point Spartans,Eleebana Eagles,Saturday,4 Nov 23,11:00,Eleebana 1,23,Boolaroo Bulldogs,Charlestown Cobras,Saturday,4 Nov 23,11:00,Charlestown 1,24,Charlestown Cobras,Boolaroo Bulldogs,Saturday,4 Nov 23,11:00,Charlestown 1,25,Belmont Bandits,Charlestown Cobras,Friday,19 Jan 24,18:00,Belmont 1,26,Glendale Guardians,Eleebana Eagles,Sunday,17 Dec 23,12:00,Glendale 1,27,Charlestown Cobras,Warners Bay Wanderers,Saturday,16 Dec 23,13:30,Charlestown 1,28,Swansea Silverbacks,Eleebana Eagles,Saturday,13 Jan 24,15:00,Swansea	 Copy code

Features





Home Screen

Home Screen



Welcome to the Lake Macquarie Premier League App for 23/24!

Enter 1 to view THIS YEAR'S TEAMS.
Enter 2 to view THIS WEEK'S GAMES.
Enter 3 to view a ROUND.
Enter 4 to view a TEAM'S DRAW.
Enter 5 to ENTER the RESULTS from a game.
Enter 6 to REMOVE the RESULTS from a game.
Enter 7 to VIEW the RESULTS.
Enter 8 to exit.

Enter your selection: |

Home Screen Code

```
from colored import fore, back, attr
from lmpf_functions import (
    print_teams, print_this_round, print_round, team_draw,
    enter_results, edit_results, view_results
)
import pyfiglet as pfg
import csv
```

Main.py

- Imports at the top
- Try: Except: creating results.csv
- The menu: _____

```
file_name = "results.csv"

try:
    # open the file in read mode
    results = open(file_name, "r")
    results.close()
    # print("In try block")
    # if it throws an error, the file doesn't exist
    # if no error, the files exists
except FileNotFoundError:
    # Now, we know the file doesn't exist
    # Create a new file
    results = open(file_name, "w")
    results.write("Game Number,Home Team,Home Goals,Away Team,Away Goals\n")
    results.close()
    # print("In except block")

def create_menu():
    print(f"\n{fore('blue')} Enter 1 to view THIS YEAR'S TEAMS. {attr('reset')}")
    print(f"\n{fore('blue')} Enter 2 to view THIS WEEK'S GAMES. {attr('reset')}")
    print(f"\n{fore('blue')} Enter 3 to view a ROUND. {attr('reset')}")
    print(f"\n{fore('blue')} Enter 4 to view a TEAM'S DRAW. {attr('reset')}")
    print(f"\n{fore('blue')} Enter 5 to ENTER the RESULTS from a game. {attr('reset')}")
    print(f"\n{fore('blue')} Enter 6 to REMOVE the RESULTS from a game. {attr('reset')}")
    print(f"\n{fore('blue')} Enter 7 to VIEW the RESULTS. {attr('reset')}")
    print(f"\n{fore('blue')} Enter 8 to exit.\n")
    choice = input(f"\n{fore('red')}{back('16')} Enter your selection: {attr('reset')} ")
    return choice
```

Home Screen Code

```
users_choice = ""

while users_choice != "8":
    users_choice = create_menu()
    if (users_choice == "1"):
        print_teams()
    elif (users_choice == "2"):
        print_this_round()
    elif (users_choice == "3"):
        print_round()
    elif (users_choice == "4"):
        team_draw()
    elif (users_choice == "5"):
        enter_results(file_name)
    elif (users_choice == "6"):
        edit_results()
    elif (users_choice == "7"):
        view_results()
    elif (users_choice == "8"):
        continue
    else:
        print("Invalid Input")
```

Main.py

- User Choice loop:
- Farewell and Logo at the bottom

```
# App Fareweall and Logo
print("\n")
print(f"\{fore('red')}\{back('white')}\nThank you for using the Lake Macquarie Premier League App! \{attr('reset')\}")
print("\n")
text = pfg.print_figlet(text = "Thank You !\n\nL . M . P . L", font = "standard", colors = "red")
```

Exit Screen

Enter your selection: 8

Thank you for using the Lake Macquarie Premier League App!





Feature 1: Print Teams

Print Teams

Enter your selection: 1

Congratulations to the following teams who've won selection in the Lake Macquarie Premier League for the 23/24 season:

Belmont Bandits
Boolaroo Bulldogs
Charlestown Cobras
Eleebana Eagles
Glendale Guardians
Speers Point Spartans
Swansea Silverbacks
Warners Bay Wanderers

Print Teams Code

```
import re
import csv
import datetime
from colored import fore, back, attr
from prettytable import from_csv
```

Impl_functions.py

- Imports at the top
- file_name declaration
- Teams and Mascots lists
- Print team function

```
def print_teams():
    print("\nCongratulations to the following teams who've won selection "
          "in the Lake Macquarie Premier League for the 23/24 season: \n")

    print(*teams, sep="\n")
    print("\n")
```

```
file_name = "results.csv"

teams = [
    f"\{fore('white')\}\{back('red')\}Belmont Bandits\{attr('reset')\}",
    f"\{fore('white')\}\{back('blue')\}Boolaroo Bulldogs\{attr('reset')\}",
    f"\{fore('white')\}\{back('yellow')\}Charlestown Cobras\{attr('reset')\}",
    f"\{fore('white')\}\{back('131')\}Eleebana Eagles\{attr('reset')\}",
    f"\{fore('white')\}\{back('56')\}Glendale Guardians\{attr('reset')\}",
    f"\{fore('white')\}\{back('172')\}Speers Point Spartans\{attr('reset')\}",
    f"\{fore('white')\}\{back('22')\}Swansea Silverbacks\{attr('reset')\}",
    f"\{fore('white')\}\{back('13')\}Warners Bay Wanderers\{attr('reset')\}"
]

team_mascots = [
    f"\{fore('white')\}\{back('red')\}Bandits\{attr('reset')\}",
    f"\{fore('white')\}\{back('blue')\}Bulldogs\{attr('reset')\}",
    f"\{fore('white')\}\{back('yellow')\}Cobras\{attr('reset')\}",
    f"\{fore('white')\}\{back('131')\}Eagles\{attr('reset')\}",
    f"\{fore('white')\}\{back('56')\}Guardians\{attr('reset')\}",
    f"\{fore('white')\}\{back('172')\}Spartans\{attr('reset')\}",
    f"\{fore('white')\}\{back('22')\}Silverbacks\{attr('reset')\}",
    f"\{fore('white')\}\{back('13')\}Wanderers\{attr('reset')\}"
]
```



Feature 2: Games this week

Games this week

Enter your selection: 2

Today's date is Sunday 24 December 2023

It is week #51 for the year

This week's games are:

Round 8, Game 29, Belmont Bandits vs Speers Point Spartans Friday, 19/1/24, 18:00

Round 8, Game 30, Boolaroo Bulldogs vs Swansea Silverbacks Saturday, 20/1/24, 12:00

Round 8, Game 31, Eleebana Eagles vs Warners Bay Wanderers Saturday, 20/1/24, 15:00

Round 8, Game 32, Glendale Guardians vs Charlestown Cobras Sunday, 21/1/24, 14:00

Games this week Code Lmpl.csv

- Info for the whole season is stored in a csv
- Week is the field this function utilises

```
Week,Round,Game Number,Home Team,Away Team,Day,Date,Time
44,1,1,Belmont Bandits,Boolaroo Bulldogs,Friday,3/11/23,18:00
44,1,2,Charlestown Cobras,Eleebana Eagles,Saturday,4/11/23,12:00
44,1,3,Glendale Guardians,Speers Point Spartans,Saturday,4/11/23,15:00
44,1,4,Swansea Silverbacks,Warners Bay Wanderers,Sunday,5/11/23,14:00
45,2,5,Boolaroo Bulldogs,Charlestown Cobras,Friday,10/11/23,18:00
45,2,6,Eleebana Eagles,Glendale Guardians,Saturday,11/11/23,12:00
45,2,7,Speers Point Spartans,Swansea Silverbacks,Saturday,11/11/23,15:00
45,2,8,Warners Bay Wanderers,Belmont Bandits,Sunday,12/11/23,14:00
46,3,9,Belmont Bandits,Charlestown Cobras,Friday,17/11/23,18:00
46,3,10,Boolaroo Bulldogs,Eleebana Eagles,Saturday,18/11/23,12:00
46,3,11,Glendale Guardians,Swansea Silverbacks,Sunday,19/11/23,14:00
46,3,12,Speers Point Spartans,Warners Bay Wanderers,Friday,24/11/23,18:00
47,4,13,Charlestown Cobras,Glendale Guardians,Friday,1/12/23,18:00
47,4,14,Eleebana Eagles,Speers Point Spartans,Saturday,2/12/23,12:00
47,4,15,Swansea Silverbacks,Belmont Bandits,Sunday,3/12/23,14:00
47,4,16,Warners Bay Wanderers,Boolaroo Bulldogs,Friday,8/12/23,18:00
48,5,17,Belmont Bandits,Eleebana Eagles,Friday,15/12/23,18:00
48,5,18,Boolaroo Bulldogs,Glendale Guardians,Saturday,16/12/23,12:00
```

Games this week Code

- Prints the date
- week_num declared and printed
- Reads the csv and only prints games within the current week of the year
- No Games between week 6 and 43

```
def print_this_round():
    from datetime import date
    # Show actual date
    my_date = datetime.date.today()
    today = date.today()
    print(today.strftime(f"\nToday's date is %A %d %B %Y"))
    # Using isocalendar() function
    year, week_num, day_of_week = my_date.isocalendar()
    print(f"\nIt is week #{str(week_num)} for the year\n")
    print("This week's games are: ")

    found_games = False # Variable to track if any games are found

    with open("lmp1.csv", "r") as f:
        reader = csv.reader(f)
        for row in reader:
            if row[0] == str(week_num):
                print("\n" "Round " + row[1] + "," +
                      " " + "Game" + " " + row[2] + "," +
                      " " + row[3] + " " + "vs" + " " +
                      row[4] + " " + row[5] + "," +
                      " " + row[6] + "," + " " + row[7])
                print("\n")
                found_games = True # Set to True if at least one game is found

    if not found_games and week_num in range(6, 43):
        print("No Games this week.")
```

Games this week test Code

Main.py

- Commenting out the datetime package
- New declaration for week_num

- week_num_to_test

```
import csv
# import datetime

def print_this_round():
    # from datetime import date
    # # Show actual date
    # my_date = datetime.date.today()
    # print(my_date)
    # # Using isocalendar() function
    # year, week_num, day_of_week = my_date.isocalendar()
    # print("Week #" + str(week_num))
    # print("This week's games are: ")

    week_num = week_num_to_test

    found_games = False # Variable to track if any games are found

    with open("lmp.csv", "r") as f:
        reader = csv.reader(f)
        for row in reader:
            if row[0] == str(week_num):
                print("\n" "Round " + row[1] + "," +
                      " " + "Game" + " " + row[2] + "," +
                      " " + row[3] + " " + "vs" + " " +
                      row[4] + " " + row[5] + "," +
                      " " + row[6] + "," + " " + row[7])
                print("\n")
            found_games = True # Set to True if at least one game is found

    if not found_games and week_num in range(6, 43):
        print("No Games this week.")

# Test the function with a specific week_num value
week_num_to_test = 10 # Change this to the desired week number
print_this_round()
```



Feature 3: Print a Round

Print a Round

Enter your selection: 3

There are 14 rounds in the LMPL Competition for this year.

Each team plays each other twice, once at Home, and once Away.

Enter the round (1-14) you'd like to print: 1

These are the games in Round 1

Round 1, Game 1, Belmont Bandits vs Boolaroo Bulldogs, Friday, 3/11/23, 18:00

Round 1, Game 2, Charlestown Cobras vs Eleebana Eagles, Saturday, 4/11/23, 12:00

Round 1, Game 3, Glendale Guardians vs Speers Point Spartans, Saturday, 4/11/23, 15:00

Round 1, Game 4, Swansea Silverbacks vs Warners Bay Wanderers, Sunday, 5/11/23, 14:00

Print Teams Code

Lmpl_functions.py

- round_number and rounds list
- Convert input to str to search lmpl.csv
- Try and Except block throws error for ints outside of 1-14
- Prints Round in this format

```
def print_round():
    print("\nThere are 14 rounds in the LMPL Competition for this year.\n")
    print("Each team plays each other twice, once at Home, and once Away.\n")
    round_number = 0
    rounds = [
        "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14"
    ]

    while str(round_number) not in rounds:
        try:
            round_number = int(input("Enter the round (1-14) you'd like to print: "))
            if str(round_number) in rounds:
                with open("lmpl.csv", "r") as f:
                    reader = csv.reader(f)
                    round_selected = rounds[round_number - 1]
                    print(f"\nThese are the games in Round {round_selected}\n")
                    for row in reader:
                        try:
                            if row[1] == round_selected:
                                print(f"Round {row[1]}, Game {row[2]}, {row[3]} vs {row[4]}, "
                                      f"{row[5]}, {row[6]}, {row[7]}")

                            print("\n")
                        except IndexError:
                            print("Invalid")
                            break
                    except ValueError:
                        print("Invalid input! Must be a number between 1 and 14")
```



Feature 4: Print Draw for a Team

Print Draw for a Team

Enter your selection: 4

Please enter the number that corresponds to the team you want to see the draw for.

1. Belmont Bandits
2. Boolaroo Bulldogs
3. Charlestown Cobras
4. Eleebana Eagles
5. Glendale Guardians
6. Speers Point Spartans
7. Swansea Silverbacks
8. Warners Bay Wanderers

Print Draw for a Team

Enter your selection: 4

Please enter the number that corresponds to the team you want to see the draw for.

1. Belmont Bandits
2. Boolaroo Bulldogs
3. Charlestown Cobras
4. Eleebana Eagles
5. Glendale Guardians
6. Speers Point Spartans
7. Swansea Silverbacks
8. Warners Bay Wanderers

Team Number: 1

This is the Belmont Bandits draw for this season.

Round 1, Belmont Bandits vs Boolaroo Bulldogs, Friday, 3/11/23, 18:00

Round 2, Warners Bay Wanderers vs Belmont Bandits, Sunday, 12/11/23, 14:00

Round 3, Belmont Bandits vs Charlestown Cobras, Friday, 17/11/23, 18:00

Round 4, Swansea Silverbacks vs Belmont Bandits, Sunday, 3/12/23, 14:00

Round 5, Belmont Bandits vs Eleebana Eagles, Friday, 15/12/23, 18:00

Round 6, Belmont Bandits vs Glendale Guardians, Friday, 29/12/23, 18:00

Round 7, Belmont Bandits vs Swansea Silverbacks, Friday, 12/1/24, 18:00

Round 8, Belmont Bandits vs Speers Point Spartans, Friday, 19/1/24, 18:00

Round 9, Belmont Bandits vs Warners Bay Wanderers, Friday, 26/1/24, 18:00

Round 10, Belmont Bandits vs Boolaroo Bulldogs, Friday, 2/2/24, 18:00

Round 11, Warners Bay Wanderers vs Belmont Bandits, Sunday, 11/2/24, 14:00

Round 12, Belmont Bandits vs Charlestown Cobras, Friday, 16/2/24, 18:00

Round 13, Swansea Silverbacks vs Belmont Bandits, Saturday, 24/2/24, 15:00

Round 14, Belmont Bandits vs Eleebana Eagles, Friday, 2/3/24, 18:00

Print Teams Code

Lmpl_functions.py

- round_number and rounds list
- Convert input to str to search lmpl.csv
- Try and Except block throws error for ints outside of 1-14
- Prints Round in this format

```
def print_round():
    print("\nThere are 14 rounds in the LMPL Competition for this year.\n")
    print("Each team plays each other twice, once at Home, and once Away.\n")
    round_number = 0
    rounds = [
        "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12", "13", "14"
    ]

    while str(round_number) not in rounds:
        try:
            round_number = int(input("Enter the round (1-14) you'd like to print: "))
            if str(round_number) in rounds:
                with open("lmpl.csv", "r") as f:
                    reader = csv.reader(f)
                    round_selected = rounds[round_number - 1]
                    print(f"\nThese are the games in Round {round_selected}\n")
                    for row in reader:
                        try:
                            if row[1] == round_selected:
                                print(f"Round {row[1]}, Game {row[2]}, {row[3]} vs {row[4]}, "
                                      f"{row[5]}, {row[6]}, {row[7]}")

                            print("\n")
                        except IndexError:
                            print("Invalid")
                            break
                    except ValueError:
                        print("Invalid input! Must be a number between 1 and 14")
```



Feature 5: Enter Results

Enter Results

Enter your selection: 5

Here are the results from the season so far:

Game Number	Home Team	Home Goals	Away Team	Away Goals
1	Bandits	5	Bulldogs	0
2	Cobras	2	Eagles	2
3	Guardians	0	Spartans	0
4	Silverbacks	3	Wanderers	1
5	Bulldogs	2	Cobras	3
6	Eagles	1	Guardians	1
7	Spartans	0	Silverbacks	4
8	Wanderers	3	Bandits	1
9	Bandits	2	Cobras	0
10	Bulldogs	1	Eagles	2
11	Guardians	4	Silverbacks	2
12	Spartans	1	Wanderers	3
13	Cobras	0	Guardians	1
14	Eagles	3	Spartans	3
15	Silverbacks	1	Bandits	0
16	Wanderers	1	Bulldogs	2
17	Bandits	4	Eagles	1
18	Bulldogs	1	Guardians	2
19	Cobras	2	Silverbacks	2

Let's enter the results of a game!

First, we need the Unique Game Number. It will be a number from 1 to 56 and can only be used once.

NOTE: Don't enter a game number that has already been used in the table above.

Enter the Game Number. (enter 0 to return to the main menu):

Enter Results Code

Lmpl_functions.py

- file_name = results.csv
- Prints Table
- game_number int
- Zero as escape
- Try and Except block throws error for ints outside of 1-56

```
def enter_results(file_name):
    view_results()

    print("\nLet's enter the results of a game!")
    print("\nFirst, we need the Unique Game Number. It will be a number from 1 to 56 and can only be used once.")
    print("\nNOTE: Don't enter a game number that has already been used in the table above.\n")
    with open(file_name, "a") as f:
        game_number = int
        while game_number != 0:
            try:
                game_number = int(input("Enter the Game Number. (enter 0 to return to the main menu): "))
                if game_number not in range(1, 57):
                    print("Must be a number between 1 and 56")
                else:
                    break
            except ValueError:
                print("Must be a number between 1 and 56")
```

Enter Results

Now we need to enter the MASCOT of the HOME TEAM

Type the name of the MASCOT out, just as you see it below.

NO SPACES, NUMBERS, or SPECIAL CHARACTERS

Bandits

Bulldogs

Cobras

Eagles

Guardians

Spartans

Silverbacks

Wanderers

Enter the Home Team: Wanderers

Accepted.

Enter the Home Goals: 0

Enter Results Code

Lmpl_functions.py

- Prints team mascots
- Invalid_name declare stops empty rows in results.csv
- Goals input. Int or throws error

```
if game_number != 0:  
    print("\nNow we need to enter the MASCOT of the HOME TEAM")  
    print("Type the name of the MASCOT out, just as you see it below.")  
    print("NO SPACES, NUMBERS, or SPECIAL CHARACTERS\n")  
    print_team_mascots()  
    home_team_name = True  
    while home_team_name:  
        home_team = input("\nEnter the Home Team: ")  
        invalid_name = re.search("[^A-Za-z]+", home_team)  
  
        if invalid_name:  
            print("No special characters will be accepted.")  
        elif not home_team:  
            print("You must type something in!")  
        else:  
            print("\nAccepted.\n")  
            home_team = home_team.lower()  
            home_team = home_team.title()  
            n = False  
            break  
  
    home_goals = 0  
    while True:  
        try:  
            home_goals = int(input("Enter the Home Goals: "))  
            break  
        except ValueError:  
            print("Invalid input. Please enter an integer.")
```

Enter Results

Great! Now onto the AWAY TEAM

Enter the MASCOT of the AWAY TEAM

Type the name of the MASCOT out, just as you see it below.

NO SPACES, NUMBERS, or SPECIAL CHARACTERS

Bandits

Bulldogs

Cobras

Eagles

Guardians

Spartans

Silverbacks

Wanderers

Enter the Away Team: Spartans

Accepted.

Enter the Away Goals: 1

Thank you! You can check your entry with option 7 on the main menu.

Enter Results Code

Lmpl_functions.py

- Prints team mascots
- Invalid_name declare stops empty rows in results.csv
- Goals input. Int or throws error

```
if game_number != 0:  
    away_team_name = True  
    print("\nGreat! Now onto the AWAY TEAM\n")  
    print("Enter the MASCOT of the AWAY TEAM")  
    print("Type the name of the MASCOT out, just as you see it below.")  
    print("NO SPACES, NUMBERS, or SPECIAL CHARACTERS\n")  
    print_team_mascots()  
    while away_team_name:  
        away_team = input("\nEnter the Away Team: ")  
        invalid_name = re.search("[^A-Za-z]+", away_team)  
  
        if invalid_name:  
            print("No special characters will be accepted.")  
        elif not away_team:  
            print("You must type something in!")  
        else:  
            print("\nAccepted.\n")  
            away_team = away_team.lower()  
            away_team = away_team.title()  
            n = False  
            break  
  
    away_goals = 0  
    while True:  
        try:  
            away_goals = int(input("Enter the Away Goals: "))  
            break  
        except ValueError:  
            print("Invalid input. Please enter an integer.")
```

Enter Results Code

Lmpl_functions.py

- Writes to results.csv

```
writer = csv.writer(f)
writer.writerow([game_number, home_team, home_goals, away_team, away_goals])
print("\nThank you! You can check your entry with option 7 on the main menu.\n")
```

- Team_mascots list

```
team_mascots = [
    f"\u001b[31m{fore('white')}{back('red')}Bandits\033[0m{attr('reset')}\u001b[0m",
    f"\u001b[31m{fore('white')}{back('blue')}Bulldogs\033[0m{attr('reset')}\u001b[0m",
    f"\u001b[31m{fore('white')}{back('yellow')}Cobras\033[0m{attr('reset')}\u001b[0m",
    f"\u001b[31m{fore('white')}{back('131')}Eagles\033[0m{attr('reset')}\u001b[0m",
    f"\u001b[31m{fore('white')}{back('56')}Guardians\033[0m{attr('reset')}\u001b[0m",
    f"\u001b[31m{fore('white')}{back('172')}Spartans\033[0m{attr('reset')}\u001b[0m",
    f"\u001b[31m{fore('white')}{back('22')}Silverbacks\033[0m{attr('reset')}\u001b[0m",
    f"\u001b[31m{fore('white')}{back('13')}Wanderers\033[0m{attr('reset')}\u001b[0m"
]
```

- Print_team_mascots function

```
def print_team_mascots():
    for mascot in team_mascots:
        print(mascot)
```



Feature 6: Remove Results

Enter Results

Enter your selection: 6

Here are the results from the season so far:

Game Number	Home Team	Home Goals	Away Team	Away Goals
1	Bandits	5	Bulldogs	0
2	Cobras	2	Eagles	2
3	Guardians	0	Spartans	0
4	Silverbacks	3	Wanderers	1
5	Bulldogs	2	Cobras	3
6	Eagles	1	Guardians	1
7	Spartans	0	Silverbacks	4
8	Wanderers	3	Bandits	1
9	Bandits	2	Cobras	0
10	Bulldogs	1	Eagles	2
11	Guardians	4	Silverbacks	2
12	Spartans	1	Wanderers	3
13	Cobras	0	Guardians	1
14	Eagles	3	Spartans	3
15	Silverbacks	1	Bandits	0
16	Wanderers	1	Bulldogs	2
17	Bandits	4	Eagles	1
18	Bulldogs	1	Guardians	2
19	Cobras	2	Silverbacks	2

Enter the GAME NUMBER for the RESULTS you want to REMOVE (enter 0 to return to the main menu):

Enter Results Code

Lmpl_functions.py

- Game_results string
- Zero as esc key
- Games_list stores entries we're keeping
- Input throws error for anything but a number in CSV

```
def edit_results():
    game_results = ""
    removed_result = ""

    while game_results != "0":
        view_results()
        print("\n")

        try:
            games_list = []

            game_results = input("Enter the GAME NUMBER for the RESULTS you want to REMOVE "
                                "(enter 0 to return to the main menu): ")

            if not game_results.isdigit():
                raise ValueError
        except ValueError:
            print("Must be a number")
```

Enter Results Code

Lmpl_functions.py

- Reads csv for game number
- Adds results to Current_game_numbers
- Writes files that made it to games_list to a new results.csv. Unwanted results gone!
- Throws error when game_number entered isn't in row[0] of results.csv

```
try:  
    current_game_numbers = []  
    with open(file_name, "r") as f:  
        reader = csv.reader(f)  
        for row in reader:  
            current_game_numbers.append(row[0])  
            if game_results != row[0]:  
                games_list.append(row)  
            else:  
                removed_result = row[0]  
  
    with open(file_name, "w") as f:  
        writer = csv.writer(f)  
        writer.writerows(games_list)  
  
    if game_results not in current_game_numbers and game_results != "0":  
        raise UnboundLocalError  
  
except UnboundLocalError:  
    print("Must be a number that corresponds with the Game Number "  
          "for the Game you want to delete.")
```

Feature 7: View Results



View Results

Enter your selection: 7

Here are the results from the season so far:

Game Number	Home Team	Home Goals	Away Team	Away Goals
1	Bandits	5	Bulldogs	0
2	Cobras	2	Eagles	2
3	Guardians	0	Spartans	0
4	Silverbacks	3	Wanderers	1
5	Bulldogs	2	Cobras	3
6	Eagles	1	Guardians	1
7	Spartans	0	Silverbacks	4
8	Wanderers	3	Bandits	1
9	Bandits	2	Cobras	0
10	Bulldogs	1	Eagles	2
11	Guardians	4	Silverbacks	2
12	Spartans	1	Wanderers	3
13	Cobras	0	Guardians	1
14	Eagles	3	Spartans	3
15	Silverbacks	1	Bandits	0
16	Wanderers	1	Bulldogs	2
17	Bandits	4	Eagles	1
18	Bulldogs	1	Guardians	2
19	Cobras	2	Silverbacks	2

View Results Code

Lmpl_functions.py

- Uses prettytable

```
import re
import csv
import datetime
from colored import fore, back, attr
from prettytable import from_csv
```

- Prints results.csv in a table

```
def view_results():
    print("\nHere are the results from the season so far: \n")
    with open("results.csv") as f:
        table = from_csv(f)
    print(table)
```

Review of Development/Build Process

Development/Build Process

- Figured out WAY too late that a CSV for the whole season probably wasn't a great idea.
- Would've been better to use a dictionary, list, list of lists, or set. Would allow colours all through app and make it updateable if teams changed.
- Went through MANY iterations trying to get my idea to work. Realised I should be printing to a separate results.csv rather than trying to append to the lmpl.csv
- Excel is a better tool for this, but this would work great WITH excel, as it's a very quick way to populate a csv.

```
1 - Game Number,Round,Home Team,Away Team
2 - 1,1,Belmont Bandits,Boolaroo Bulldogs
3 - 2,1,Charlestown Cobras,Eleebana Eagles
4 - 3,1,Glendale Guardians,Speers Point Spartans
5 - 4,1,Swansea Silverbacks,Warners Bay Wanderers
6 - 5,2,Boolaroo Bulldogs,Charlestown Cobras
7 - 6,2,Eleebana Eagles,Glendale Guardians
8 - 7,2,Speers Point Spartans,Swansea Silverbacks
9 - 8,2,Warners Bay Wanderers,Belmont Bandits
10 - 9,3,Belmont Bandits,Charlestown Cobras
11 - 10,3,Boolaroo Bulldogs,Eleebana Eagles
12 - 11,3,Glendale Guardians,Swansea Silverbacks
13 - 12,3,Speers Point Spartans,Warners Bay Wanderers
14 - 13,4,Charlestown Cobras,Glendale Guardians
15 - 14,4,Eleebana Eagles,Speers Point Spartans
16 - 15,4,Swansea Silverbacks,Belmont Bandits
17 - 16,4,Warners Bay Wanderers,Boolaroo Bulldogs
18 - 17,5,Belmont Bandits,Eleebana Eagles
19 - 18,5,Boolaroo Bulldogs,Glendale Guardians
20 - 19,5,Charlestown Cobras,Swansea Silverbacks
21 - 20,5,Warners Bay Wanderers,Speers Point Spartans
22 - 21,6,Belmont Bandits,Glendale Guardians
23 - 22,6,Boolaroo Bulldogs,Speers Point Spartans
```

```
1 + Round,Game Number,Home Team,Away Team,Day,Date,Time,Home Team Goals,Away Team Goals
2 + 1,1,Belmont Bandits,Boolaroo Bulldogs,Friday,3/11/23,18:00,5,0
3 + 1,2,Charlestown Cobras,Eleebana Eagles,Saturday,4/11/23,12:00,2,2
4 + 1,3,Glendale Guardians,Speers Point Spartans,Saturday,4/11/23,15:00,0,0
5 + 1,4,Swansea Silverbacks,Warners Bay Wanderers,Sunday,5/11/23,14:00,3,1
6 + 2,5,Boolaroo Bulldogs,Charlestown Cobras,Friday,10/11/23,18:00,2,3
7 + 2,6,Eleebana Eagles,Glendale Guardians,Saturday,11/11/23,12:00,1,1
8 + 2,7,Speers Point Spartans,Swansea Silverbacks,Saturday,11/11/23,15:00,0,4
9 + 2,8,Warners Bay Wanderers,Belmont Bandits,Sunday,12/11/23,14:00,3,1
10 + 3,9,Belmont Bandits,Charlestown Cobras,Friday,17/11/23,18:00,2,0
11 + 3,10,Boolaroo Bulldogs,Eleebana Eagles,Saturday,18/11/23,12:00,1,2
12 + 3,11,Glendale Guardians,Swansea Silverbacks,Sunday,19/11/23,14:00,4,2
13 + 3,12,Speers Point Spartans,Warners Bay Wanderers,Friday,24/11/23,18:00,1,3
14 + 4,13,Charlestown Cobras,Glendale Guardians,Friday,1/12/23,18:00,0,1
15 + 4,14,Eleebana Eagles,Speers Point Spartans,Saturday,2/12/23,12:00,3,3
16 + 4,15,Swansea Silverbacks,Belmont Bandits,Sunday,3/12/23,14:00,1,0
17 + 4,16,Warners Bay Wanderers,Boolaroo Bulldogs,Friday,8/12/23,18:00,1,2
18 + 5,17,Belmont Bandits,Eleebana Eagles,Friday,15/12/23,18:00,4,1
19 + 5,18,Boolaroo Bulldogs,Glendale Guardians,Saturday,16/12/23,12:00,1,2
20 + 5,19,Charlestown Cobras,Swansea Silverbacks,Sunday,17/12/23,14:00,2,2
21 + 5,20,Warners Bay Wanderers,Speers Point Spartans,Friday,22/12/23,18:00,0,1
22 + 6,21,Belmont Bandits,Glendale Guardians,Friday,29/12/23,18:00,3,0
23 + 6,22,Boolaroo Bulldogs,Speers Point Spartans,Saturday,30/12/23,12:00,2,3
```



Challenges,
Ethical
Issues,
Favourite
parts

Challenges



ALL OF IT!!!



ALL OF IT!!!

LIKE...
EVERYTHING!!!!



ALL OF IT!!!

LIKE...
EVERYTHING!!!

...Literally!

Challenges

- Indentations: will be the Death of me!
- Solution: LOTS of trial and error

```
def enter_results(file_name):
    print("Let's enter the results of a game!")
    with open(file_name, "a") as f:

        game_number = 0
        game_number = int(input("Enter the Game Number: "))
        if game_number >= 57:
            print("Must be a number between 1 and 56")
            return
    except ValueError:
        print("Must be a number between 1 and 56")
        return

    print(f"\n{teams[0]}")
    print(f"\n{teams[1]}")
    print(f"\n{teams[2]}")
    print(f"\n{teams[3]}")
    print(f"\n{teams[4]}")
    print(f"\n{teams[5]}")
    print(f"\n{teams[6]}")
    print(f"\n{teams[7]}")

    try:
        home_team = input("\nEnter the Home Team: ")
        home_goals = 0
        home_goals = int(input("Enter the Home Team Goals: "))
    except ValueError:
        print("Must be a Number")

    print(f"\n{teams[0]}")
    print(f"\n{teams[1]}")
    print(f"\n{teams[2]}")
    print(f"\n{teams[3]}")
    print(f"\n{teams[4]}")
    print(f"\n{teams[5]}")
    print(f"\n{teams[6]}")
    print(f"\n{teams[7]}")
    try:
        away_team = input("Enter the Away Team: ")
        away_goals = 0
        away_goals = int(input("Enter the Away Team Goals: "))
    except ValueError:
        print("Must be a Number")
    writer = csv.writer(f)
    writer.writerow([game_number, home_team, home_goals, away_team, away_goals])
```

```
def enter_results(file_name):
    print("Let's enter the results of a game!")
    with open(file_name, "a") as f:

        try:
            game_number = 0
            game_number = int(input("Enter the Game Number: "))
            if game_number >= 57:
                print("Must be a number between 1 and 56")
                return
        except ValueError:
            print("Must be a number between 1 and 56")
            return

        print("Type the name of the Home Team as you see it below.")
        for team in teams:
            print(f"\n{team}")
        home_team = input("\nEnter the Home Team: ")

        home_goals = 0 # Initializing with an invalid value to enter the loop
        while True:
            try:
                home_goals = int(input("Enter the Home Goals: "))
            except ValueError:
                print("Invalid input. Please enter an integer.")

        print("Type the name of the Away Team as you see it below.")
        for team in teams:
            print(f"\n{team}")
        away_team = input("\nEnter the Away Team: ")

        away_goals = 0 # Initializing with an invalid value to enter the loop
        while True:
            try:
                away_goals = int(input("Enter the Away Goals: "))
            except ValueError:
                print("Invalid input. Please enter an integer.")

        writer = csv.writer(f)
        writer.writerow([game_number, home_team, home_goals, away_team, away_goals])
        print("\nThank you! You can check your entry with option 7 on the main menu.\n")
```

Challenges

- Broken Table: prettytable throws this error whenever there's an empty row in a csv.

```
Here are the results from the season so far
Traceback (most recent call last):
  File "/Users/reecedoyle/CA/term-1/T1A3/main.py", line 56, in <module>
    enter_results(file_name)
  File "/Users/reecedoyle/CA/term-1/T1A3/lmpl_functions.py", line 116, in enter_results
    view_results()
  File "/Users/reecedoyle/CA/term-1/T1A3/lmpl_functions.py", line 250, in view_results
    table = from_csv(f)
           ^^^^^^^^^^
  File "/usr/local/lib/python3.11/site-packages/prettytable/prettytable.py", line 2439, in from_csv
    table.add_row([x.strip() for x in row])
  File "/usr/local/lib/python3.11/site-packages/prettytable/prettytable.py", line 1407, in add_row
    raise ValueError(
ValueError: Row has incorrect number of values, (actual) 0!=5 (expected)
```

- Solution: The re.search special characters loop.

```
invalid_name = re.search("[^A-Za-z]+", home_team)

if invalid_name:
    print("No special characters will be accepted.")
elif not home_team:
    print("You must type something in!")
```

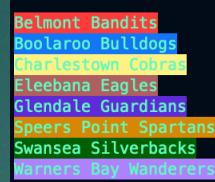
Ethical Issues

- All data used by this app is fictional and generated by AI, so I'm not using anyone else's data.
- While the places are real, the teams are not. I'm not infringing on anyone else's intellectual property.
- I've cited and credited all packages that the app uses.

Favourite Parts

- I did it! (I'm not from a tech background). I learned so much!

- Love the colours and logo!



- A I persevered and found a solution to get the app done, even if it wasn't my favourite option.

Now we need to enter the MASCOT of the HOME TEAM
Type the name of the MASCOT out, just as you see it below.
NO SPACES, NUMBERS, or SPECIAL CHARACTERS

Bandits
Bulldogs
Cobras
Eagles
Guardians
Spartans
Silverbacks
Wanderers

Enter the Home Team: Wanderers

Accepted.

Enter the Home Goals: 0

- It's about Football and my hometown! What could be better?



prettytable highly effective

Enter your selection: 7				
Here are the results from the season so far:				
Game Number	Home Team	Home Goals	Away Team	Away Goals
1	Bandits	5	Bulldogs	0
2	Eagles	2	Eagles	2
3	Guardians	0	Spartans	0
4	Silverbacks	3	Wanderers	1
5	Bulldogs	2	Cobras	3
6	Eagles	1	Guardians	1
7	Spartans	0	Silverbacks	4
8	Wanderers	3	Bandits	1
9	Bandits	2	Cobras	0
10	Bulldogs	1	Eagles	2
11	Guardians	4	Silverbacks	2
12	Spartans	1	Wanderers	3
13	Cobras	0	Guardians	1
14	Eagles	3	Spartans	3
15	Silverbacks	1	Bandits	0
16	Wanderers	1	Bulldogs	2
17	Bandits	4	Eagles	1
18	Bulldogs	1	Guardians	2
19	Cobras	2	Silverbacks	2

Datetime was a great solution for real time use.

Enter your selection: 2				
Today's date is Sunday 24 December 2023				
It is week #51 for the year				
This week's games are:				
Round 8, Game 29, Belmont Bandits vs Speers Point Spartans Friday, 19/1/24, 18:00				
Round 8, Game 30, Boolaroo Bulldogs vs Swansea Silverbacks Saturday, 20/1/24, 12:00				
Round 8, Game 31, Eleebana Eagles vs Warners Bay Wanderers Saturday, 20/1/24, 15:00				
Round 8, Game 32, Glendale Guardians vs Charlestown Cobras Sunday, 21/1/24, 14:00				



Thank you
for your time!