FitTrackCLI User Guide

Introduction

FitTrackCLI is command-line-based chatbot to help users manage their NAPFA related exercises and goals! The user can:

1. Record and track training sessions.
2. Calculate NAPFA scores from training sessions.
3. Set fitness goals.
4. Visualise their training progress.

The NAPFA score sheet used for this chatbot can be found here. This guide will bring you through the various features of FitTrackCLI, and how to utilise them!

FitTrackCLI's Features

Before diving into the details of the features, it's essential to understand how the exercises are stored in this CLI chatbot. The table below provides a quick reference for the different exercises, their acronyms, valid input formats, units, and examples of invalid entries:

Exercise	Acronym	Valid Input	Units/Format	Invalid Input Examples
Pull Ups	PU	15	Repetitions (integer)	-5, "ten"
Shuttle Run	SR	11.2	Seconds (1 decimal)	11.25, 11., "11:20"
Sit And Reach	SAR	30	Centimeters (integer)	-10, "thirty"
Sit Ups	SU	20	Repetitions (integer)	-3, "twenty"
Standing Broad Jump	SBJ	200	Centimeters (integer)	-50, "two hundred"
Walk And Run	WAR	12:45	Minutes:Seconds (MM:SS)	13:60, "12:7", "12.45", "twelve:45"

1. User Configuration: `set`

Purpose: Set the age and gender of the user.

Format: `set [gender] [age]`

- `gender` and `age` field must be non-empty. Gender must be "male" or "female". Age must be an integer.

Example: `male 12`

Expected output:

```
____________________________________________________________________________________________________
You are a 12 year old male.
____________________________________________________________________________________________________
```

> **Note**: The user will be prompted to set their age and gender at each program start. This is because individuals will age, and hence the calculations of their NAPFA exercise points will be different.

2. Help Function: `help`

Purpose: Prints a complete list of valid commands.

Format: `help`

Expected output:

```
____________________________________________________________________________________________________
COMMAND                                                           EXAMPLE
help                                                              help
set (gender) (age)                                                set male 12
add (session name)                                                add session1
modify (session index) (new datetime)                             modify 1 04/08/1986 12:30
list                                                              list
view (session index)                                              view 1
edit (session index) (exercise acronym) (repetitions/time)        edit 1 PU 1
delete (session index)                                            delete 1
edit-mood (session index) (mood description)                      edit-mood 1 Sad
remind (Event / Task) (deadline)                                  remind NAPFA DD/MM/YYYY
list-remind                                                       list-remind
delete-remind (reminder index)                                    delete-remind1
upcoming-remind                                                   upcoming-remind
add-goal (goal name) (deadline)                                   add-goal run 12/12/2024 14:00:00
delete-goal (goal index)                                          delete-goal 1
list-goal                                                         list-goal
gpoints                                                           gpoints PU
gperformance                                                      gperformance WAR
add-water (amount)                                                add-water 500
delete-water (index)                                              delete-water 1
list-water                                                        list-water
add-food (food item) (calories)                                   add-food apple100
delete-food (index)                                               delete-food 1
list-food                                                         list-food
list-intake                                                       list-intake
exit                                                              exit
____________________________________________________________________________________________________
```

3. Add a Training Session: `add`

Purpose: Adds a Training Session with the specified name.

Format: `add [session name]`

- `session name` field must be non-empty.

Example: `add session1`

Expected output:

```
____________________________________________________________________________________________
Got it. I've added a new training session:
1. session1 | 29/10/2024 12:40
There are 1 sessions in the list.
____________________________________________________________________________________________
```

> **Note 1**: The session datetime of the training added training session will be the current system datetime of the system.
>
> **Note 2**: The user's age and gender, provided at the start, determine the point calculation criteria for each training session. These attributes are tied to each session and directly influence the points awarded. Note that training sessions added in different program runs may have varying criteria, as the user's age and gender may change between runs.

4. Modify the DateTime of a Training Session: `modify`

Purpose: Modifies the recorded date and time of an existing Training Session.

Format: `modify [session index] [new datetime]`

- `session index` and `new datetime` fields must be non-empty.

Example: `modify 1 10/11/2024 12:56`

Example Output:

```
____________________________________________________________________________________________
Session 1 has been modified:
New Date/Time: 10/11/2024 12:56
____________________________________________________________________________________________
```

> **Note**: This could cause the modified session to have a different index in the training session list, as as the list is sorted in a chronological order.

5. List all Training Sessions: `list`

Purpose: Displays all Training Sessions the user has added.

Format: `list`

Example Output:

```
____________________________________________________________________________________________
Here are your training sessions:
1. session1 | 29/10/2024 12:40
2. session2 | 29/10/2024 12:41
There are 2 sessions in the list.
____________________________________________________________________________________________
```

> **Note**: The training sessions are listed in their chronological datetime order.

6. View a Training Session: `view`

Purpose: View the details of a Training Session, including session name, datetime, exercise data, points and awards.

Format: `view [session index]`

Example: `view 1`

Expected Output:

```
____________________________________________________________________________________________
Training Session: session1
Training Datetime: 29/10/2024 12:40
Pull Up Station | Reps: 0 | 0 points
Shuttle Run Station | Time: NA | 0 points
Sit and Reach Station | Distance: 0cm | 0 points
Sit Up Station | Reps: 0 | 0 points
Standing Broad Jump Station | Distance: 0cm | 0 points
Walk and Run Station | Time: NA | 0 points
Total points: 0
Overall Award: No Award
____________________________________________________________________________________________
```

7. Edit a Training Session: `edit`

Purpose: Edit the details of a training session, namely exercise and reps/time.

Format: `edit [session index] [exercise acronym] [repetitions/time]`

> **Note**: The format for exercise acronyms and the corresponding repetitions/timing strictly adheres to the table provided at the beginning of the features section.

Example: `edit 1 PU 45`

Expected Output:

```
1
Exercise edited! Here's your new input: Reps: 45 | 5 points
____________________________________________________________________________________________________
Training Session: session1
Training Datetime: 29/10/2024 12:40
Mood: No mood recorded
Pull Up Station | Reps: 45 | 5 points
Shuttle Run Station | Time: NA | 0 points
Sit and Reach Station | Distance: 0cm | 0 points
Sit Up Station | Reps: 0 | 0 points
Standing Broad Jump Station | Distance: 0cm | 0 points
Walk and Run Station | Time: NA | 0 points
Total points: 5
Overall Award: No Award
____________________________________________________________________________________________________
```

> **Note**: Refer to the beginning of the features section for the table of valid and invalid inputs.

8. Edit your post-Training Session mood: `edit-mood`

Purpose: Edit the post-training mood of a training session.

Format: `edit-mood [session index] [mood description]`

Example: `edit-mood 1 Happy`

Expected Output:

```
____________________________________________________________________________________________________

"Mood for Training Session 1 updated: Happy"
____________________________________________________________________________________________________
```

9. Deleting a Training Session: `delete`

Purpose: Removes a Training Session from the list.

Format: `delete [session index]`

Example: `delete 1`

Expected Output:

```
____________________________________________________________________________________________________
Got it. I've deleted this training session:session1 | 29/10/2024 12:40
There are 1 sessions in the list.
____________________________________________________________________________________________________
```

10. Exiting the program: `exit`

Purpose: Ends FitTrack CLI task and exits.

Format: `exit`

Expected Output:

```
____________________________________________________________________________________________
Bye! Hope to see you again soon!
____________________________________________________________________________________________
```

11. Add a Reminder: `remind`

Purpose: Adds a Reminder with the specified description and due date.

Format: `remind [description] // [deadline]`

- `description` and `deadline` fields must be non-empty.
- `deadline` field must be formatted `dd/MM/yyyy` or `dd/MM/yyyy HH:mm` .
- If `deadline` field is given as `dd/MM/yyyy` , `HH:mm` information will default to `00:00` on that date.

Example: `remind NAPFA // 31/12/2024`

Expected output:

```
____________________________________________________________________________________________
Got it. I've added a new reminder
1. NAPFA | 31/12/2024 00:00
There are 1 reminders in your list.
____________________________________________________________________________________________
```

12. List all Reminders: `list-remind`

Purpose: Displays all active Reminders the user has added.

Format: `list-remind`

Example Output:

```
____________________________________________________________________________________________
Here are your reminders:
1. TEST1 | 31/12/2024 00:00
2. TEST2 | 30/12/2024 00:00
3. TEST3 | 29/12/2024 00:00
There are 3 reminders in your list.
____________________________________________________________________________________________
```

13. List soon-due Reminders: `upcoming-remind`

Purpose: Displays all Reminders the user has added that are due in the next week (7 days).

Format: `upcoming-remind`

Example Output:

```
____________________________________________________________________________________
There are 1 reminders due in the next week:
1. UPCOMINGTEST | 06/11/2024 00:00
You have 2 reminders in total. View them with 'list-remind'.
____________________________________________________________________________________
```

14. Delete a Reminder: `delete-remind`

Purpose: Removes a reminder from your list.

Format: `delete-remind [reminder index]`

- `reminder index` can be found using the `list-remind` command.

Example: `delete-remind 1`

Expected Output:

```
____________________________________________________________________________________
Got it. I've deleted this reminder:NAPFA | 31/12/2024 00:00
There are 0 reminders in your list.
____________________________________________________________________________________
```

15. Add Goal: `add-goal`

Purpose: User can add a fitness goal to the the list of goals and attach a deadline to it in order to have clear targets to prepare for the NAPFA test.

Format: `add-goal (goal name) (deadline)`

- `(goal name)` is the description of the goal (e.g., "run", "swim").
- `[deadline]` is an optional argument. If provided, it should follow the format DD/MM/YYYY HH:MM:SS.
- If no deadline is provided, the time will default to 00:00:00 on the specified date.

Example: `add-goal run 12/12/2024 14:00:00`

Expected Output:

If a deadline is provided:

```
____________________________________________________________________________________________________
Goal added: run
Deadline: 12/12/2024 14:00:00
____________________________________________________________________________________________________
```

If a deadline is not specified:

```
____________________________________________________________________________________________________
Goal added: run
No deadline set.
____________________________________________________________________________________________________
```

16. List of Goals: `list-goal`

View a list of all fitness goals and deadlines to keep track of progress in preparation for the NAPFA test

Input Command: list-goal

Format: `list-goal`

Example: `list-goal`

Expected Output:

```
____________________________________________________________________________________________________
Goals:
1. Goal: run, Deadline: 2024-12-12T14:00
____________________________________________________________________________________________________
```

17. Delete Goal: `delete-goal`

User can delete a fitness goal to the the list of goals to moderate a fitness goal.

Format: `delete-goal (goal index)`

Example: `delete-goal 1`

Expected Output:

```
____________________________________________________________________________________________________
Goal at index 1 has been removed.
____________________________________________________________________________________________________
```

18. Display Points Graph: `gpoints`

Purpose: Display the points the user has accumulated across different training sessions. Points can either reflect the user's total overall points or be specific to a chosen exercise.

Format:
- Overall Points: Use `gpoints` to view total points for the training sessions.
- Exercise specific points: Use `gpoints [EXERCISE_ACRONYM]` to view points for a specific exercise in the training session.

Example 1: `gpoints`

Expected Output 1:

```
Here's your point progression over the various training sessions:
Session Description | Date             | Points
--------------------|------------------|
session1            | 07/11/2024 16:57 | ********* (9)
session2            | 07/11/2024 16:57 | ************** (14)
session3            | 07/11/2024 16:57 | *** (3)
session4            | 07/11/2024 16:57 | ******* (7)
```

Example 2: `gpoints PU`

Expected Output 2:

```
Here's your point progression for PULL_UP over your training sessions:
Session Description | Date             | Points
--------------------|------------------|
session1            | 07/11/2024 16:57 | * (1)
session2            | 07/11/2024 16:57 | ** (2)
session3            | 07/11/2024 16:57 | *** (3)
session4            | 07/11/2024 16:57 | ***** (5)
```

19. Display Performance Graph: `gperformance`

Purpose:

1. For non-time based station, command displays a bar graph of the raw performance metric (i.e. distance/length/rep) against session index.
2. For time based station, command displays a scatter graph of the normalised performance metric (i.e. time) against session index.

Format: `gperformance [EXERCISE_ACRONYM]`

Example 1: `gperformance PU`

```
Here's your progression for PULL_UP over your training sessions:

16                                                                             2 points
15                                                                                 *
14                                                           2 points              *
13                                                               *                 *
12                      1 points                                 *                 *
11                          *                                    *                 *
10                          *                                    *                 *
```

```
9                      *                                   *                 *
8                      *                                   *                 *
7    0 points          *                                   *                 *
6        *             *                                   *                 *
5        *             *                                   *                 *
4        *             *                                   *                 *
3        *             *                                   *                 *
2        *             *                                   *                 *
1        *             *            0 points               *                 *
     Session 1         Session 2         Session 3         Session 4         Session 5
  07/11/2024 17:45  08/11/2024 17:45  09/11/2024 17:45  10/11/2024 17:45  11/11/2024 17:45
```

Example 2: `gperformance WAR`

Expected Output:

```
             09:11             08:05             NIL             10:15             09:55
1.00                                                               *
0.95
0.90
0.85                                                                                 *
0.80
0.75
0.70
0.65
0.60
0.55
0.50           *
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05                             *
          Session 1         Session 2         Session 3         Session 4         Session 5
       07/11/2024 17:46  08/11/2024 17:46  09/11/2024 17:46  10/11/2024 17:46  11/11/2024 17:46
```

20. Add Food Intake: `add-food`

View a list of daily food intake to have a more comprehensive understanding of factors affecting my fitness.

Format: `add-food (food name) (calories)`

Example: `add-food apple 100`

Expected Output:

```
____________________________________________________________________________________________
Got it. I've added food item: apple (100 calories, 06/11/2024 17:32:07).
____________________________________________________________________________________________
```

21. Delete Food Intake: delete-food

Purpose: Remove a food item from the daily food intake list.

Format: `delete-food (food index)`

Example: `delete-food 1`

Expected Output:

```
____________________________________________________________________________________________________
Got it. I've deleted food item: apple - 219 calories, added on 11/11/2024 07:57
____________________________________________________________________________________________________
```

22. List Food Intake: list-food

Purpose: Display the list of all food items that have been added for the day.

Format: `list-food`

Example: `list-food`

Expected Output:

```
____________________________________________________________________________________________________
Here is your food intake list:
1. apple (100 calories) at 06/11/2024 17:32:07
____________________________________________________________________________________________________
```

23. Add Water Intake: add-water

Purpose: Add water intake in milliliters to track hydration levels.

Format: `add-water (water ml)`

Example: `add-water 43`

Expected Output:

```
____________________________________________________________________________________________________
Got it. I've added 43ml of water at 11/11/2024 07:58.
____________________________________________________________________________________________________
```

24. Delete Water Intake: delete-water

Purpose: Remove a specified amount of water from the daily water intake record.

Format: `delete-water (water ml)`

Example: `delete-water 1`

Expected Output:

```
____________________________________________________________________________________________________
Got it. I've deleted water item: 100 ml, added on 07/11/2024 22:34
____________________________________________________________________________________________________
```

25. List Water Intake: list-water

Purpose: Display the total water intake recorded for the day.

Format: `list-water`

Example: `list-water`

Expected Output:

```
____________________________________________________________________________________________________
Water Entries for 2024-11-11:
1. 43 ml, added on 11/11/2024 07:58
2. 3443 ml, added on 11/11/2024 07:58
Total daily water: 3486 ml
____________________________________________________________________________________________________
```

26. List Daily Intake: list-intake

Purpose: Display the list of all food items and water items that have been added for the day.

Format: `list-intake`

Example: `list-intake`

Expected Output:

```
Here is your daily intake summary:
____________________________________________________________________________________________________
Food Entries for 2024-11-11:
Total daily Calories: 0

Water Entries for 2024-11-11:
Total daily water: 0 ml
____________________________________________________________________________________________________
```

FitTrackCLI's Command Summary

Command	Format	Example
set	`set GENDER AGE`	`set male 12`
help	`help`	`help`
add	`add SESSION_NAME`	`add session1`
modify	`modify SESSION_INDEX DATETIME`	`modify 1 10/11/2024 12:30`
list	`list`	`list`
view	`view SESSION_INDEX`	`view 1`
edit	`edit SESSION_INDEX EXERCISE_ACRONYM REPETITION/TIME_DURATION`	`edit 1 PU 45`
modify	`modify SESSION_INDEX NEW_DATETIME`	`modify 1 12/12/2024 14:00:00`
delete	`delete SESSION_INDEX`	`delete 1`
exit	`exit`	`exit`
remind	`remind REMINDER_NAME // DEADLINE`	`remind run // 12/12/2024`
list-remind	`list-goal`	`list-remind`
upcoming-remind	`upcoming-remind`	`upcoming-remind`
delete-remind	`delete-remind REMINDER_INDEX`	`delete-remind 1`
add-goal	`add-goal GOAL_NAME DEADLINE`	`add-goal run 12/12/2024 14:00:00`
delete-goal	`delete-goal GOAL_INDEX`	`delete-goal 1`
list-goal	`list-goal`	`list-goal`
gpoints	`gpoints / gpoints EXERCISE_ACRONYM`	`gpoints / gpoints PU`
gperformance	`gpeformance EXERCISE_ACRONYM`	`gperformance PU`
add-water	`add-water`	`add-water 500`
delete-water	`delete-water`	`delete-water 1`
list-water	`list-water`	`list-water`
add-food	`add-food`	`add-food apple 100`
delete-food	`delete-food`	`delete-food 1`
list-food	`list-food`	`list-food`

Command	Format	Example
list-intake	`list-intake`	`list-intake`

FAQ

Q: I was an 18-year-old male up until yesterday and have been using FitTrackCLI to track my results over the past few months. Now that I’ve just turned 19, if I re-enter the program with my updated age and attempt to edit past sessions created when I was 18, will my points still be calculated correctly?

- Each session you create is linked to your age and gender at the time of creation. If you edit past session data that was created when you were 18 years old and male, the points will be calculated based on that age and gender. Therefore, editing past session data from when you were younger will reflect calculations based on your age at that time.
- Now that you are 19, any new session will have points calculated based on your current age of 19 and male.