

# T1A3 Workout Tracker

Terminal Application Assignment

Rahal Abeyrathna



# Key Functions

- There are 3 key features;
  - Workout Entries
  - Workout Templates
  - Exercise Database
- All 3 of these features utilize 3 key functions
  - Create
    - Create custom record or entry
  - View
    - View table of existing records
  - Delete
    - Delete a record from the database
- There are many other functions implemented to get those functions to work properly



# Exercise Database

```
-- Exercise Database Menu --

Please select from the following options:

[ 1 ] Create a new exercise
[ 2 ] View all Exercises
[ 3 ] Delete an Exercise
[ 4 ] Return to Main Menu

Enter the function you would like to enter:
█
```

```
Would you like to save the following exercise?

+-----+-----+
| Exercise | PB Weight (kg) |
+-----+-----+
| Test_Exercise | 20.0 |
+-----+-----+

Please enter 'YES' or 'NO':
█
```

```
Confirmation: Are you sure you would like to delete Squats? This action cannot be undone.
(Type 'YES' to confirm or 'NO' to abort:)
█
```

Displaying all exercises in Exercise Database:

Exercise	PB Weight (kg)
Squats	0.0
Push-ups	0.0
Deadlifts	0.0
Bench Press	20.0
Overhead Press	20.0
Rows	20.0
Plank	0.0
Lunges	0.0
Bicep Curls	20.0
Tricep Dips	0.0
Leg Press	0.0
Chest Flyes	0.0
Lat Pulldowns	0.0
Shoulder Press	0.0
Leg Curls	0.0
Hammer Curls	0.0
Tricep Extensions	20.0
Calf Raises	0.0
Plank Rows	0.0
Mountain Climbers	0.0
Russian Twists	0.0
Box Jumps	0.0
Seated Rows	0.0
Lateral Raises	20.0
Hamstring Curls	0.0
Incline Bench Press	0.0
Front Squats	0.0
Chin-ups	20.0
Dumbbell Rows	0.0
Leg Extensions	0.0
Side Planks	0.0
Step-ups	0.0
Reverse Flyes	0.0
Skull Crushers	0.0
Jumping Jacks	0.0
Seated Leg Press	0.0
Forearm Plank	0.0
Barbell Curls	0.0
Cable Crunches	0.0
Plank to Push-up	0.0
Wide Grip Pull-ups	0.0
Seated Shoulder Press	0.0
Kettlebell Swings	0.0
Leg Raises	0.0
Cable Rows	0.0
Front Raises	0.0
Decline Bench Press	0.0
Medicine Ball Slams	0.0
Reverse Crunches	0.0
Bicycle Crunches	0.0
Incline Curls	0.0

Press enter when you would like to return to the Exercise List menu.



# Workout Templates

```
-- Workout Templates Menu --
```

```
Please select from the following options:
```

- [ 1 ] Create a New Workout Template
- [ 2 ] View all Workout Templates
- [ 3 ] Delete a Workout Template
- [ 4 ] Return to Main Menu

```
Enter the function you would like to enter:
```

```
█
```

```
Would you like to save the following template?
```

```
Template Name: Test Template
```

Exercise	Last Working Weight (kg)
Push-Ups	0
Bench Press	0
Squats	0

```
Please enter 'YES' or 'NO':
```

```
█
```

```
-- Template Name: Test Template --
```

```
Please enter the name of the exercise you would like to add:
```

```
Type 'done' when you've added all of your exercises:
```

```
fake_exercise
```

```
Error: Fake_Exercise does not exist in database. Did you want to add it?
```

```
Type 'YES' or 'NO':
```

```
█
```



# Workout Entries

```
-- Workout Entry Menu --  
  
Please select from the following options:  
  
[ 1 ] Create a new Workout Entry  
[ 2 ] View all Workout Logs  
[ 3 ] Delete a Workout Log  
[ 4 ] Return to Main Menu  
  
Enter the function you would like to enter:  
█
```

```
Would you like to save the following Workout Entry?  
  
Entry Date: 2023-12-19  
  
Template Used: Full Body  
  
+-----+-----+  
| Exercise | Working Weight (kg) |  
+-----+-----+  
| Squats   | 20.0                 |  
| Deadlifts| 20.0                 |  
| Bench Press| 20.0                |  
| Rows     | 20.0                 |  
| Overhead Press| 20.0              |  
| Bicep Curls| 20.0                |  
| Tricep Dips| 20.0                |  
| Leg Curls | 20.0                 |  
+-----+-----+  
  
Please enter 'YES' or 'NO':  
█
```

```
-- Enter Working Weight for Squats --  
  
Your last recorded weight for squats in this template was: 0.0kg  
  
What was your working weight for this exercise today?20  
  
-- Enter Working Weight for Deadlifts --  
  
Your last recorded weight for deadlifts in this template was: 0.0kg  
  
What was your working weight for this exercise today?20  
  
-- Enter Working Weight for Bench Press --  
  
Your last recorded weight for bench press in this template was: 0.0kg  
  
What was your working weight for this exercise today?20  
  
-- Enter Working Weight for Rows --  
  
Your last recorded weight for rows in this template was: 0.0kg  
  
What was your working weight for this exercise today?20  
  
-- Enter Working Weight for Overhead Press --  
  
Your last recorded weight for overhead press in this template was: 0.0kg  
  
What was your working weight for this exercise today?20  
  
-- Enter Working Weight for Bicep Curls --  
  
Your last recorded weight for bicep curls in this template was: 0.0kg  
  
What was your working weight for this exercise today?20  
  
-- Enter Working Weight for Tricep Dips --  
  
Your last recorded weight for tricep dips in this template was: 0.0kg  
  
What was your working weight for this exercise today?20█
```



## run.sh

```
#!/bin/bash

source setup.sh

if check_python; then
    if start_venv; then
        echo "Checks have all passed. Starting application.."
        python3 ./modules/main.py
    else
        echo "Error: Failed to set up virtual environment and install dependencies."
        exit 2
    fi
else
    echo "Error: Python 3 is required to run this application. Please install Python 3 and try again."
    exit 1
fi
```

# Starting the application

## setup.sh

```
#!/bin/bash

check_python() {
    python_version=$(python3 --version 2>&1)
    if [[ $python_version == *"3"* ]]; then
        return 0
    else
        return 1
    fi
}

start_venv() {
    echo "Creating virtual environment.."
    if python3 -m venv .venv; then
        echo "Virtual environment created successfully."
        source .venv/bin/activate

        echo "Installing required dependencies.."
        if pip3 install -r ./requirements.txt; then
            echo "Dependencies installed successfully."
            return 0
        else
            echo "Error: Failed to install dependencies."
            return 1
        fi
    else
        echo "Error: Failed to create virtual environment."
        return 1
    fi
}
```



# General Menu

```
general_menu("Main", main_options_list)
```

```
# Main Menu Options
main_options_list = {
    " Enter Workout Entry Menu": "general_menu('Workout Entry', pw_options_list)",
    " Enter Workout Templates Menu": "general_menu('Workout Templates', wt_options_list)",
    " View Exercise List Database": "general_menu('Exercise Database', el_options_list)",
}
```

```
Welcome the the Workout Tracker application!
```

```
-- Main Menu --
```

```
Please select from the following options:
```

```
[ 1 ] Enter Workout Entry Menu
[ 2 ] Enter Workout Templates Menu
[ 3 ] View Exercise List Database
[ 4 ] Exit Application
```

```
Enter the function you would like to enter:
```



# General Menu

```
# General menu function used to navigate through main menus of the application
def general_menu(menu_name, options_list):
    total_options = len(options_list) + 1
    user_selection = ""
    while user_selection != total_options:
        print(f"{Style.BOLD}{Fore.CYAN} -- {menu_name} Menu --\n")
        print(
            f"{Style.BOLD}{Fore.YELLOW}Please select from the following options:{Style.reset}\n"
        )
        for index, option in enumerate(options_list):
            print(f"{Fore.blue}[ {index + 1} ]{option}")
        if menu_name == "Main":
            print(
                f"{Fore.RED}[ {total_options} ] Exit Application{Style.reset}"
            )
        else:
            print(
                f"{Fore.YELLOW}[ {total_options} ] Return to Main Menu{Style.reset}"
            )
        user_selection = input(
            f"{Fore.green}\nEnter the function you would like to enter: {Style.reset}\n"
        )
        clear_console()
        try:
            user_selection = int(user_selection)
            if user_selection < 1 or user_selection > total_options:
                raise ValueError
            if user_selection == total_options:
                break
            else:
                selected_option = list(options_list.keys())[user_selection - 1]
                try:
                    eval(options_list[selected_option])
                except Exception as e:
                    general_exception(
                        "An error occurred while trying to evaluate the option as a command",
                        e,
                    )
        except ValueError:
            clear_console()
            print(
                f"{Fore.red}Error: Please enter a valid number between 1 and {total_options}:{Style.reset}\n"
            )
        except Exception as e:
            general_exception("An error occurred in the general menu", e)
```





# Create Function

```
# Function used to create different records for different features
def create_submenu(record_type, file_path):
    try:
        clear_console()
        record_name = get_record_name(record_type)
        record_exists = check_record_exists(file_path, record_name)
        while record_exists:
            print(
                f"{Fore.RED}Error: Looks like that {record_type} already exists.{Style.RESET}\n"
            )
            record_name = get_record_name(record_type)
            record_exists = check_record_exists(file_path, record_name)
        append_data = get_append_data(record_name, file_path)
        record_empty = check_record_empty(file_path, append_data)
        if not record_empty:
            create_record = confirm_record(file_path, record_name, append_data)
            if create_record:
                append_csv(file_path, append_data, record_type)
            else:
                print(
                    f"{Fore.RED}User cancelled: Aborting function...{Style.RESET}"
                )
        else:
            print(
                f"{Fore.RED}Error: That {record_type} is empty. Aborting creation..{Style.RESET}\n"
            )
    except Exception as e:
        general_exception(
            "An error occurred while trying to create a record", e
        )
```



# Display Function

```
# Function to display all exercises in Exercise Database
def el_display(csv_path):
    try:
        clear_console()
        with open(csv_path, "r") as file:
            csv_reader = csv.reader(file)
            header = next(csv_reader)
            exercise_table = PrettyTable()
            exercise_table.field_names = ["Exercise", "PB Weight (kg)"]
            for row in csv_reader:
                (
                    exercise_name,
                    pb_weight,
                ) = row
                exercise_table.add_row(row)
            print(
                f"{Fore.blue}Displaying all exercises in Exercise Database:{Style.reset}\n"
            )
            print(f"{exercise_table}\n")
            input(
                f"{Fore.green}Press enter when you would like to return to the Exercise List menu:\n{Style.reset}"
            )
            clear_console()
    except FileNotFoundError as fe:
        file_not_found(csv_path, fe)
    except Exception as e:
        general_exception(
            "An error occurred while displaying the exercise database", e
        )
```



# Delete Function

```
# Function used for features which have a delete sub-menu
def delete_submenu(menu_name, csv_path, header):
    try:
        clear_console()
        delete_loop = False
        update_menu_list(csv_path)
        return_value = len(menu_list) + 1
        user_selection = display_menu_list(menu_name, return_value, "Delete")
        if user_selection != return_value:
            selected_record = menu_list[user_selection - 1]
            delete_loop = True
        while delete_loop:
            user_confirmation = input(
                f"{Style.bold}{Fore.red}Confirmation: Are you sure you would like to delete {Fore.YELLOW}{selected_record}{Style.reset}{Style.
                bold}{Fore.red}? This action cannot be undone.{Style.reset}\n{Fore.red}(Type {Fore.YELLOW}'YES'{Style.reset}{Fore.red} to
                confirm or {Fore.YELLOW}'NO'{Style.reset}{Fore.red} to abort:){Style.reset}\n"
            )
            if user_confirmation == "YES":
                clear_console()
                delete_csv_row(csv_path, selected_record)
                print(f"{Fore.red}Deleted {selected_record}{Style.reset}")
                update_menu_list(csv_path)
                return_value = len(menu_list) + 1
                delete_loop = False
            elif user_confirmation == "NO":
                clear_console()
                print(
                    f"{Fore.red}User cancelled. Aborting deletion..{Style.reset}"
                )
                delete_loop = False
            else:
                clear_console()
                print(f"{Fore.RED}Error: Invalid input.{Style.reset}\n")
        except Exception as e:
            general_exception(
                "An error occurred while confirming deletion of record", e
            )
    )
```



# Delete CSV Row

```
# Function to delete row from CSV
def delete_csv_row(csv_path, selected_record):
    try:
        transfer_rows = []
        with open(csv_path, "r") as file:
            csv_reader = csv.reader(file)
            header = next(csv_reader, None)
            for row in csv_reader:
                if row[0] != selected_record:
                    transfer_rows.append(row)
        with open(csv_path, "w", newline="") as file:
            # Convert header into a string instead of list
            file.write(",".join(header) + "\n")
            csvwriter = csv.writer(file)
            csvwriter.writerows(transfer_rows)
    except FileNotFoundError as fe:
        file_not_found(csv_path, fe)
    except Exception as e:
        general_exception(
            "An error occurred while trying to delete a row from a CSV", e
        )
```



# Assignment Review

- Was a lot of fun to work on, hope to go about it differently next time
- There was a lot to manage and it got very confusing at times
- Learnt a lot about project management
  - Trello, Notion
- Didn't really run into any ethical issues during development
  - The CSV files will likely need to be kept locally if it was used in real-life scenarios for data privacy
- Favourite part was the testing and having everything work as planned

