Hi, I’m Dan - T1A3 Terminal assignment

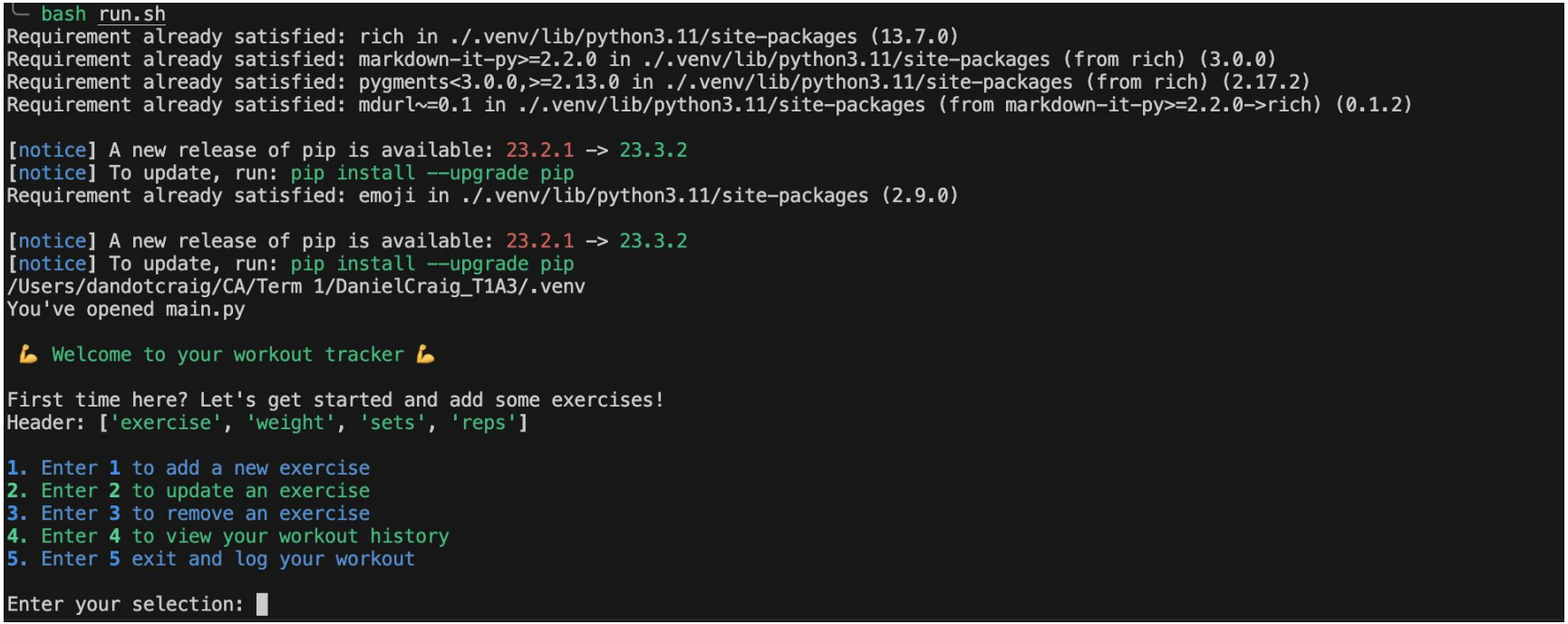
Disclaimer

I’ve made another video running through the app and the logic

View HERE

OVERVIEW

MENU


└ bash run.sh
Requirement already satisfied: rich in ./.venv/lib/python3.11/site-packages (13.7.0)
Requirement already satisfied: markdown-it-py>=2.2.0 in ./.venv/lib/python3.11/site-packages (from rich) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in ./.venv/lib/python3.11/site-packages (from rich) (2.17.2)
Requirement already satisfied: mdurl~=0.1 in ./.venv/lib/python3.11/site-packages (from markdown-it-py>=2.2.0->rich) (0.1.2)
[notice] A new release of pip is available: 23.2.1 -> 23.3.2
[notice] To update, run: pip install --upgrade pip
Requirement already satisfied: emoji in ./.venv/lib/python3.11/site-packages (2.9.0)
[notice] A new release of pip is available: 23.2.1 -> 23.3.2
[notice] To update, run: pip install --upgrade pip
/Users/dandotcraig/CA/Term 1/DanielCraig_T1A3/.venv
You've opened main.py
💪 Welcome to your workout tracker 💪
First time here? Let's get started and add some exercises!
Header: ['exercise', 'weight', 'sets', 'reps']
1. Enter 1 to add a new exercise
2. Enter 2 to update an exercise
3. Enter 3 to remove an exercise
4. Enter 4 to view your workout history
5. Enter 5 exit and log your workout
Enter your selection:

ADD

```
First time here? Let's get started and add some exercises!
Header: ['exercise', 'weight', 'sets', 'reps']

1. Enter 1 to add a new exercise
2. Enter 2 to update an exercise
3. Enter 3 to remove an exercise
4. Enter 4 to view your workout history
5. Enter 5 exit and log your workout

Enter your selection: 1
You choose add a new exercise
Enter your new exercise: pull up
Enter your weight in kgs: 12kg
Enter your sets: 12
Enter your reps: 12
```

Terminal app

UPDATE

```
Enter your selection: 2
You choose update an exercise
Enter the correct name of the exercise you want to update: row
['row', '12kg', '4', '12']

Update the exercises details:
Re-type or update row: rowarm
Re-type or update, must be in kg 12kg: 14kg
Re-type or update sets 4: 5
Re-type or update reps 12: 10
```

DELETE

```
Here is your updated workout
Exercise 0: ['exercise', 'weight', 'sets', 'reps']

This exercise ['pull up', '12kg', '12', '12'] has been removed

1. Enter 1 to add a new exercise
2. Enter 2 to update an exercise
3. Enter 3 to remove an exercise
4. Enter 4 to view your workout history
5. Enter 5 exit and log your workout

Enter your selection:
```

EXIT AND SAVE

```
Update the exercises details:
Re-type or update row: rowarm
Re-type or update, must be in kg 12kg: 14kg
Re-type or update sets 4: 5
Re-type or update reps 12: 10

Here is your updated workout
Exercise 0: ['exercise', 'weight', 'sets', 'reps']
Exercise 1: ['rowarm', '14kg', '5', '10']

This exercise has been updated to ['rowarm', '14kg', 5, 10]

1. Enter 1 to add a new exercise
2. Enter 2 to update an exercise
3. Enter 3 to remove an exercise
4. Enter 4 to view your workout history
5. Enter 5 exit and log your workout

Enter your selection: 5
You selected exit and log your workout
Exercise logged, well-done, cya next time!
🧐Thanks you for using workout tracker 🧐

~/CA/Term 1/DanielCraig_T1A3 on main !3 ?2
```

APP LOGIC

MENU

```
def home_menu():
    print("[blue]1. Enter 1 to add a new exercise")
    print("[green]2. Enter 2 to update an exercise")
    print("[blue]3. Enter 3 to remove an exercise")
    print("[green]4. Enter 4 to view your workout history")
    print("[blue]5. Enter 5 exit and log your workout")
    choice = input("\nEnter your selection: ")
    return choice

user_choice = ""
# User input links upto these external functions
while user_choice != "5":
    user_choice = home_menu()
    if (user_choice == "1"):
        add_exercise(file_name)
    elif (user_choice == "2"):
        update_exercise(file_name)
    elif (user_choice == "3"):
        remove_exercise(file_name)
    elif (user_choice == "4"):
        view_history(history)
    elif (user_choice == "5"):
        save_exit(file_name, history)
        print("[green]Exercise logged, well-done, cya next time!")
        continue
    else:
        print("Invalid Input - input needs to be a number between 1 - 5 ")
# Thanks the user for using the program!
print(":woozy_face:[blue]Thanks you for using workout tracker :woozy_face:")
```

Terminal app

ADD

```python
def add_exercise(file_name):
    print("[green]You choose add a new exercise")

    # We check for to make sure the user inputs the right format
    while True:
        try:
            exercise_name = input("Enter your new exercise: ")
            if any(i.isdigit() for i in exercise_name):
                print("[red] :cross_mark: Exercise must not contain numbers, try again but with words this time. :cross_mark:")
                raise ValueError
        except ValueError as e:
            print(e)
        else:
            break

    # We check for to make sure the user inputs the right format
    while True:
        try:
            kilo = "kg"
            weight_number = str(input("Enter your weight in kgs: "))
            if kilo not in weight_number:
                print("[red] :cross_mark: Weight must contain a number in kgs :cross_mark: ")
                raise ValueError
        except Exception:
            print("[red] :cross_mark: Try again :cross_mark: ")
        else:
            break
```

UPDATE

```
def update_exercise(file_name):
    print("[green]You choose update an exercise")
    # user input

    while True:
        exercise_name_update = input("Enter the correct name of the exercise you want to update: ")
        # list variable
        exercise_list = []
        # open file to read contents
        replaced_row = None
        with open(file_name, "r") as f:
            # new copy of the file
            reader = csv.reader(f)
            # loop through each row
            for row in reader:
                if (exercise_name_update != row[0]):
                    # we want it in the update cvs
                    exercise_list.append(row)
                else:
                    # replace row grabs the data to be used later
                    replaced_row = row
        if (replaced_row is None):
            # If it doesn't work... print this
            print("[red] :cross_mark: Input does not match, try again. :cross_mark: ")
            continue
```

DELETE

```
# Removes exercise
def remove_exercise(file_name):

    while True:
        print("[green]You selected to delete an exercise")
        # user input
        exercise_name_remove = input("Enter the name of the exercise you want to remove: ")
        # list variable
        exercise_list = []
        # open file to read contents
        replaced_row = None
        with open(file_name, "r") as f:
            # new copy of the file
            reader = csv.reader(f)
            # loop through each row
            for row in reader:
                # if its not the input
                if (exercise_name_remove != row[0]):
                    # we want it in the update cvs
                    exercise_list.append(row)
                else:
                    replaced_row = row

        if (replaced_row is None):
            print("[red] :cross_mark: Input does not match, try again.  :cross_mark: ")
            continue
        else:
            break
```

EXIT AND SAVE

```python
# Logs the workouts along with the date for next time.
def save_exit(file_name, history):
    print("[blue]You selected exit and log your workout")
    copied_file = []
    with open(file_name, 'r') as f:
        reader = csv.reader(f)

        headers = next(reader)

        for row in reader:
            copied_file.append(row)

    # Writes the new row into the history log to be returned in the future
    with open(history, 'a', newline='') as f:
        writer = csv.writer(f)
        writer.writerow([])
        writer.writerow([f'{datetime.datetime.now():%d-%B-%Y %H:%M:%S}'])
        writer.writerow([])

    with open(history, "a") as f:
        writer = csv.writer(f)
        writer.writerows(copied_file)

    return ""
```