# Fitness Solution Report

Group 110
Professor Bills
INST326

# Fitness Solution Report (Brief Overview)

---

**Description:** The goal of Group 110's final project is to provide the user with a complex fitness report that is tailored to their unique characteristics, environment, and overall workout objectives. The script takes in multiple arguments in the form of various biometric identifiers, then personalizes the output based on these characteristics. The report was structured and stylized using Pandas, and provides the user with various recommendations to begin their fitness journey. This information has been coded to be easy to read and interpret.

---

**Repository Files:**

- fsr.py
    - Script for our project
    - Takes in user's biometric identifiers as arguments and outputs fitness report
- fsr_test.py
    - Test script for our project
    - Makes sure that functions in the fitness report are handling arguments properly

---

**Instructions:**

1. Install the following modules: Re, Requests, Sys, MatPlotLib, Pandas, Argparse, and BeautifulSoup
2. If operating script on Mac, go to Applications>Python 3.9 Folder>Install Certificates.command. Click this file and let the certificate install.
3. Download 'fitness_resources.txt' and 'tips.csv' from the repository and add them to the same directory you have placed 'fsr.py'
4. Open file labeled 'fsr.py' in your source code editor interface of choice
5. Run the file in your terminal by following the example below:
    a. Windows:
        i. python3 fsr.py 18 70 male 150 gain 4
    b. Mac:

        i. python fsr.py 18 70 male 150 gain 4
    c. Explanation
        i. 18 is the age
        ii. 70 is the weight in kg.
        iii. ‘male’ is the gender
        iv. 150 is the height in cm.
        v. ‘gain’ is the user’s objective
        vi. 4 is the amount of days the user wants to workout
6. Receive and interpret output

---

**Interpreting the Output:** After receiving output from our project, the user is shown a display of workouts tailored to their fitness objectives and desired workout frequency. Additionally, they are provided with biological information (such as BMR/BMI), along with dietary suggestions that correspond with this number. To help motivate the user, we have included a fitness magazine resources dataframe and a dataframe that contains daily tips. Finally, a graph will be displayed that shows the user’s weight changes after weeks into the program based on their desired objective (outputting nothing if the user is maintaining weight).

---

**Works Cited:**


- *Concept: Web Scraping*
    - No Specific Line
        - URL: **Click here**
        - Explanation: General web scraping advice lecture, where we
    - Lines 52-54
        - URL: **Click here**
        - Explanation: Taught us more about the ‘find_all’ function in BeautifulSoup and how we can use it to scrape HTML tables
    - Lines 52-54
        - URL: **Click here**
        - Explanation: Learning more about table tags and their specific meanings in order to scrape the proper tables
    - Lines 57-75
        - URL: **Click here**

        - Explanation: Learning about limited retry loops in order to scrape table titles and organize/align data
    - Line 296
        - URL: **Click here**
        - Explanation: Extracting ‘strong’ elements using BeautifulSoup
    - Lines 59-75, 291, 298
        - URL: **Click here**
        - Explanation: Using ‘.text’ and ‘.get_text’ in BeautifulSoup
- *Concept: Calculation Formulas*
    - Lines 102-119
        - URL: **Click here**
        - Explanation: Formulas used in BioCalculator function
- *Concept: Regular Expressions*
    - Lines 327-239
        - URL: **Click here**
        - Explanation: Learning more about the ‘.replace’ function in the Regex module
- *Concept: Pandas*
    - Line 333
        - URL: **Click here**
        - Explanation: Use case of ‘reset_index’ function in Pandas
    - Line 352
        - URL: **Click here**
        - Explanation: Use case of ‘read_csv’ function in Pandas
- *Concept: MatPlotLib*
    - Line 352
        - URL: **Click here**
        - Explanation: How to create a PyPlot and use ‘py.xxx’
- *Concept: Parsing:*
    - Lines: 393-414
        - URL: **Click here**
        - Explanation: Writing a ‘parse_args’ function with required arguments and a __main__ statement that prints out class objects