Cameron Lawrence
T1A3
Terminal Application

Terminal Application - Max Hang Test Score

My application was created for rock climbers how would like to calculate and store their finger strength by performing a max hang test. This test requires the climber to hang from a 20mm hangboard with either the bodyweight or bodyweight + added weight. The sum of either would then equate to a percentage score determining the climbers finger strength. The score can then help the climber to determine further progressions and subtleties in their training/programming. The percentage is calculated where the body climbers bodyweight is equal to 100%.

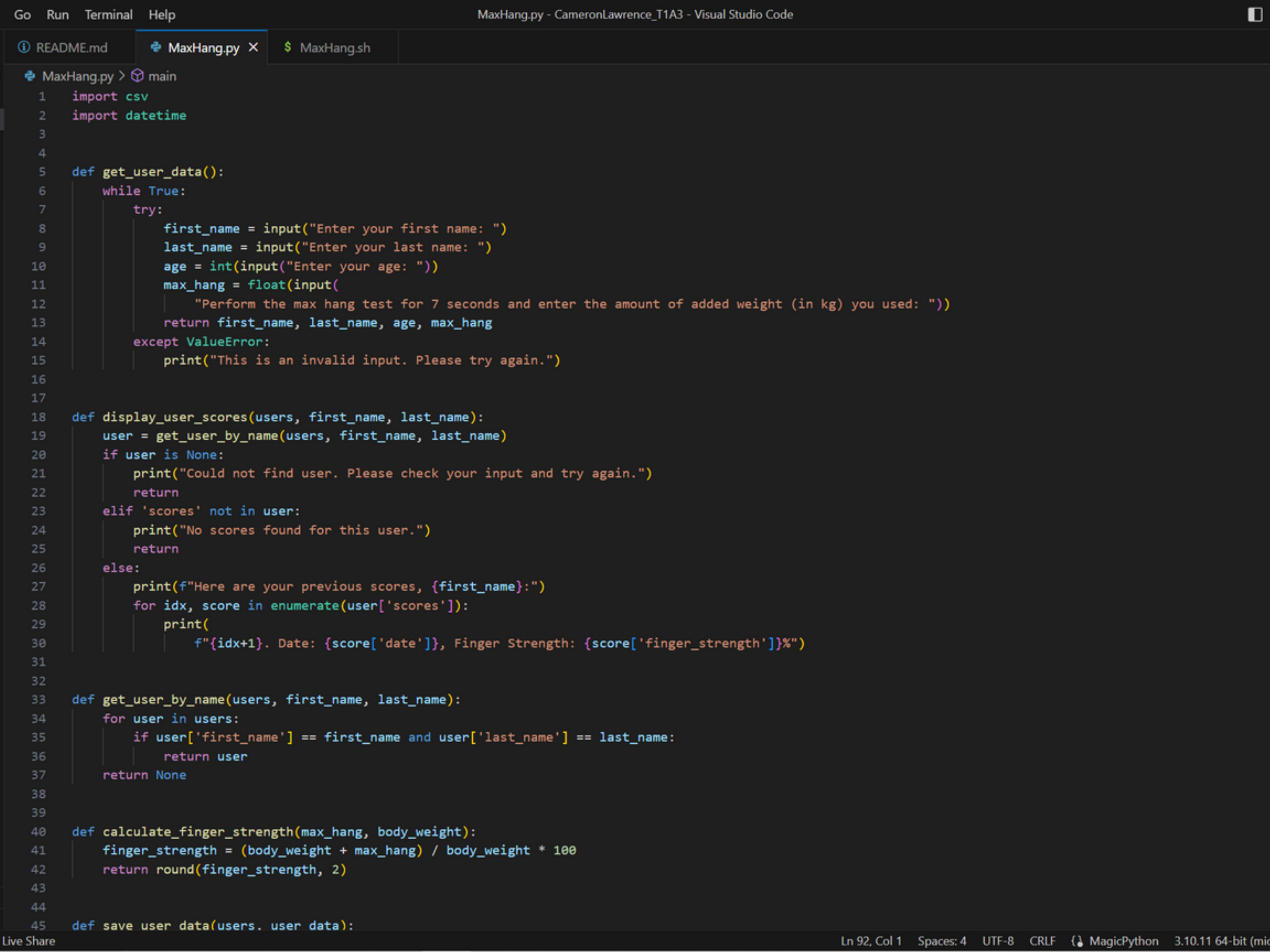
Terminal Application - Functionality

My application has several functions. They are as follows.

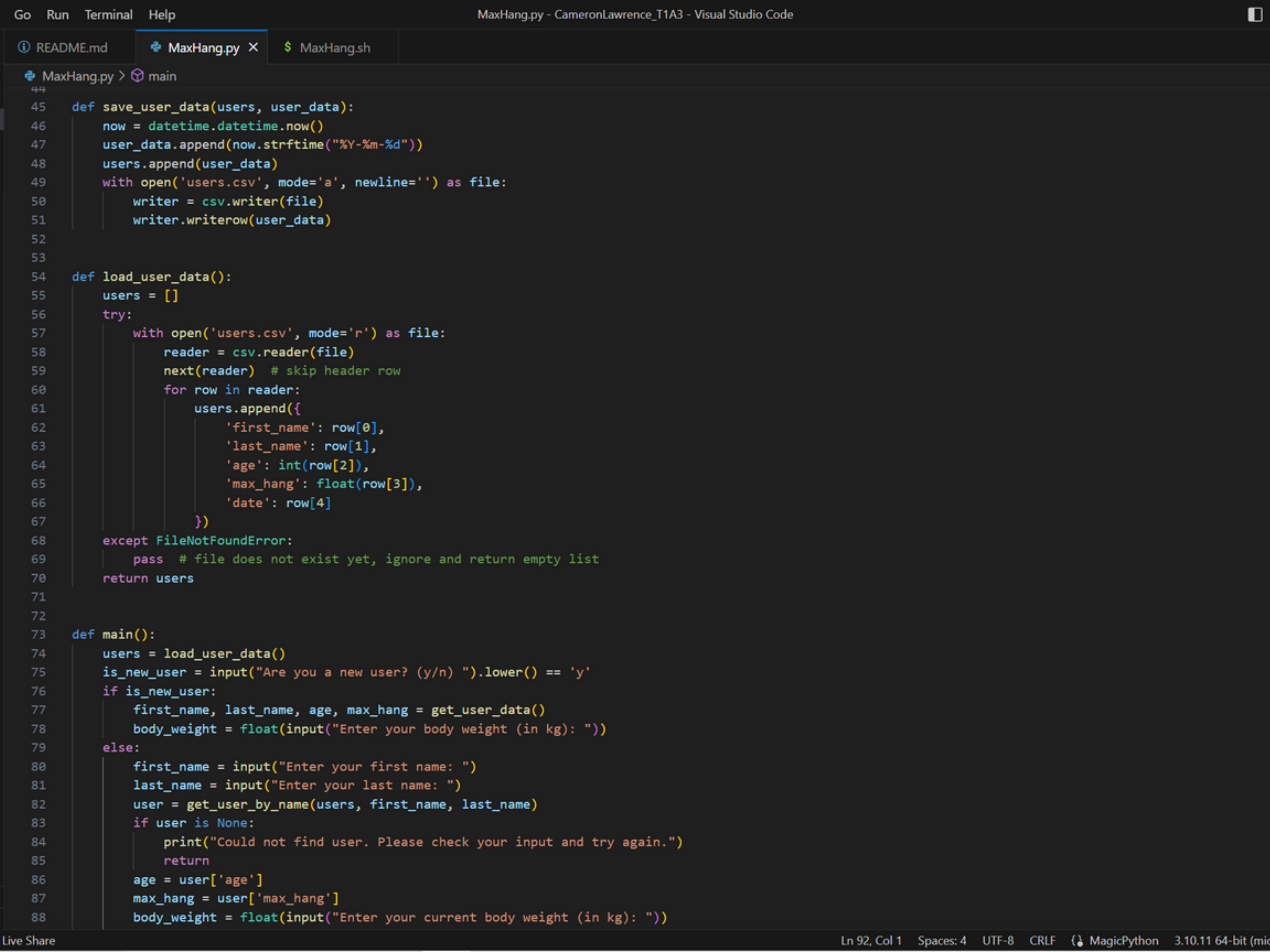
- Data collection - The application requires the user whether new or existing to input certain amounts of data in order to calculate the users score. It requires first & last name, age, bodyweight and added weight when testing.
- Data storing and retrieval - The application can save all the recorded data for each user in an associated csv file. This can then be called upon to retrieve user information and previous scores.
- Score calculation - This calculation is the core function of the terminal application. It uses the data collected to calculate the users finger strength as a percentage. Where finger strength (%) = (bodyweight + added weight) / bodyweight * 100.

Terminal Application - Code Overview

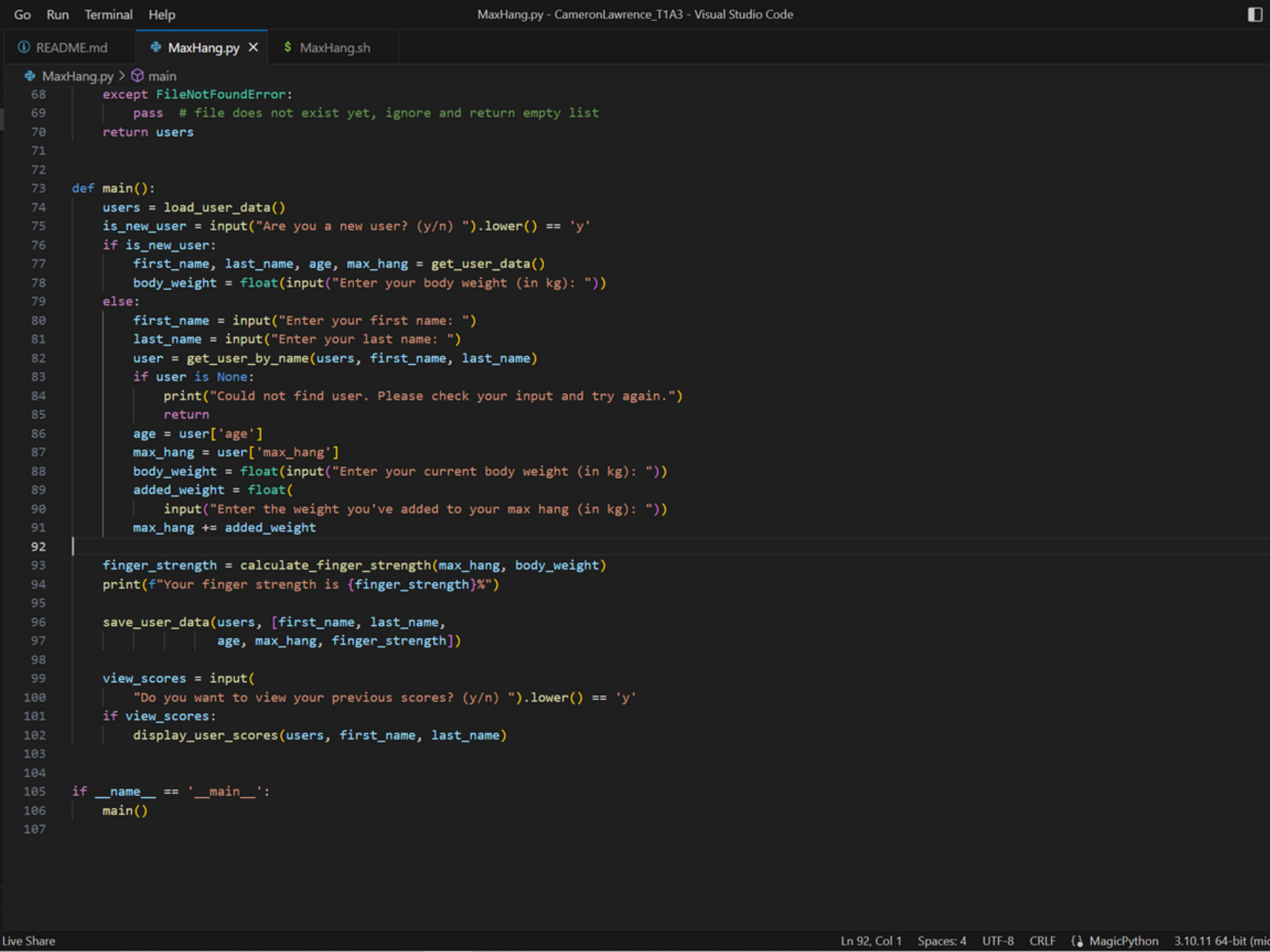
The coding for this application for me was quite tricky as I have struggled with python. It required lots of trial and error and lots of constant back and forth between lesson recordings. Please see the following overview of my code.



```python
import csv
import datetime


def get_user_data():
    while True:
        try:
            first_name = input("Enter your first name: ")
            last_name = input("Enter your last name: ")
            age = int(input("Enter your age: "))
            max_hang = float(input(
                "Perform the max hang test for 7 seconds and enter the amount of added weight (in kg) you used: "))
            return first_name, last_name, age, max_hang
        except ValueError:
            print("This is an invalid input. Please try again.")


def display_user_scores(users, first_name, last_name):
    user = get_user_by_name(users, first_name, last_name)
    if user is None:
        print("Could not find user. Please check your input and try again.")
        return
    elif 'scores' not in user:
        print("No scores found for this user.")
        return
    else:
        print(f"Here are your previous scores, {first_name}:")
        for idx, score in enumerate(user['scores']):
            print(
                f"{idx+1}. Date: {score['date']}, Finger Strength: {score['finger_strength']}%")


def get_user_by_name(users, first_name, last_name):
    for user in users:
        if user['first_name'] == first_name and user['last_name'] == last_name:
            return user
    return None


def calculate_finger_strength(max_hang, body_weight):
    finger_strength = (body_weight + max_hang) / body_weight * 100
    return round(finger_strength, 2)


def save_user_data(users, user_data):
```



```python
def save_user_data(users, user_data):
    now = datetime.datetime.now()
    user_data.append(now.strftime("%Y-%m-%d"))
    users.append(user_data)
    with open('users.csv', mode='a', newline='') as file:
        writer = csv.writer(file)
        writer.writerow(user_data)


def load_user_data():
    users = []
    try:
        with open('users.csv', mode='r') as file:
            reader = csv.reader(file)
            next(reader)  # skip header row
            for row in reader:
                users.append({
                    'first_name': row[0],
                    'last_name': row[1],
                    'age': int(row[2]),
                    'max_hang': float(row[3]),
                    'date': row[4]
                })
    except FileNotFoundError:
        pass  # file does not exist yet, ignore and return empty list
    return users


def main():
    users = load_user_data()
    is_new_user = input("Are you a new user? (y/n) ").lower() == 'y'
    if is_new_user:
        first_name, last_name, age, max_hang = get_user_data()
        body_weight = float(input("Enter your body weight (in kg): "))
    else:
        first_name = input("Enter your first name: ")
        last_name = input("Enter your last name: ")
        user = get_user_by_name(users, first_name, last_name)
        if user is None:
            print("Could not find user. Please check your input and try again.")
            return
        age = user['age']
        max_hang = user['max_hang']
        body_weight = float(input("Enter your current body weight (in kg): "))
```



```python
    except FileNotFoundError:
        pass  # file does not exist yet, ignore and return empty list
    return users


def main():
    users = load_user_data()
    is_new_user = input("Are you a new user? (y/n) ").lower() == 'y'
    if is_new_user:
        first_name, last_name, age, max_hang = get_user_data()
        body_weight = float(input("Enter your body weight (in kg): "))
    else:
        first_name = input("Enter your first name: ")
        last_name = input("Enter your last name: ")
        user = get_user_by_name(users, first_name, last_name)
        if user is None:
            print("Could not find user. Please check your input and try again.")
            return
        age = user['age']
        max_hang = user['max_hang']
        body_weight = float(input("Enter your current body weight (in kg): "))
        added_weight = float(
            input("Enter the weight you've added to your max hang (in kg): "))
        max_hang += added_weight

    finger_strength = calculate_finger_strength(max_hang, body_weight)
    print(f"Your finger strength is {finger_strength}%")

    save_user_data(users, [first_name, last_name,
                   age, max_hang, finger_strength])

    view_scores = input(
        "Do you want to view your previous scores? (y/n) ").lower() == 'y'
    if view_scores:
        display_user_scores(users, first_name, last_name)


if __name__ == '__main__':
    main()
```

Thank you