TITLE: Lifestyle Stability Meter (Multi-User).

It is a multi user program. A C-based lifestyle tracking system that allows multiple users to log in. Enter daily wellness data, and view personalized analytics based on their previous entriesDeveloped a multi-user, C-based lifestyle tracking system. This program enables users to log in, input their daily wellness data, and receive personalized analytics derived from their previous entries.

SUBMITTED BY:

AVANTIKA SRIVASTAVA
590027832
BATCH 48
BTECH CSE
SCHOOL OF COMPUTER SCIENCE

ABSTRACT

The *Lifestyle Stability Meter (Multi-User Version)* is a C-based console application created to help users track their daily wellness habits and derive meaningful insights from them. The program supports multiple users and allows them to log daily parameters such as sleep, productivity, stress level, and mood. Using a custom scoring algorithm, each entry is converted into a *Life Score* between 0 and 100.

Over time, users can view analytics including:

- average lifestyle score
- highest and lowest score
- performance trends
- ASCII-based visual graphs
- personalized improvement suggestions

This project demonstrates the practical use of C programming concepts such as **file handling**, **structures**, **loops**, **decision-making**, and **modular programming**. It bridges theoretical concepts with a meaningful real-world application.

1. INTRODUCTION

Today's lifestyle heavily influences both physical and mental well-being. People often struggle to track daily habits such as sleep quality, stress levels, and productivity. Having a simple tool that collects and analyzes such data can help individuals understand patterns and make informed decisions.

The *Lifestyle Stability Meter* provides exactly that.
It is designed as a console-based application for students learning the fundamentals of C programming. It incorporates essential programming concepts while also providing practical, insightful output to users.

The project is beginner-friendly yet advanced enough to demonstrate structured coding, data organization, and file-based persistence.

This program is designed using **C language**, especially focusing on:

- File Handling
- Struct-based data management
- Multiple user support
- Personalized analytics
- User-friendly console interface

2. PROBLEM STATEMENT

Many individuals want to understand whether their lifestyle is improving or declining. However:

- Most wellness applications require internet access
- Professional tracking tools are too complex
- No simple multi-user console-based tool exists
- Students rarely find beginner-level projects with meaningful real-world usage

Thus, there is a need for a lightweight, offline, console-based tracker that:

- accepts daily wellness input
- stores multiple users
- analyzes long-term patterns

The *Lifestyle Stability Meter* addresses this gap effectively.

3. OBJECTIVES

Functional Objectives

- Allow multiple users to maintain separate lifestyle logs
- Enable users to record daily wellness parameters
- Calculate a standard Life Score for each entry
- Provide analytics based on user history
- Generate ASCII graph visualizations

Educational Objectives

- Demonstrate usage of structures
- Implement file handling techniques
- Follow modular program architecture
- Apply loops and conditional logic
- Work with arrays and strings
- Experience real-world style coding in C

SYSTEM DESIGN

The program follows a **three-level architecture**:

Level 1 – User Interface

- Login screen
- User menu
- Analytics display

Level 2 – Processing

- Score calculation
- Trend analysis
- Data loading and filtering

Level 3 – Storage

- All data stored in a text file: `users_log.txt`
- Simple, readable format
- Easy cross-platform compatibility

ALGORITHM OF THE PROGRAM:

Step 1: Start: Begin the execution of the program.

Step 2: Display Main Menu

Show the following options to the user:

1. Login
2. Exit

Step 3: Read User Choice (Main Menu)

- If choice = **2**, go to Step 16 (Exit).
- If choice = **1**, continue.

Step 4: Prompt User for Username

Ask the user to enter a username
(e.g., “avantika”, “tanisha”, “mehak”).

Step 5: Load User Entries from File

Open the file `users_log.txt`.
Read all the entries.
Store only matching entries for the given username.

Step 6: Display User Menu

Show options:

1. Enter Today’s Data

2. View Analytics
3. Logout

Step 7: Read User Choice (User Menu)

- If **choice = 3**, go back to Step 2 (Main Menu).
- If **choice = 1**, go to Step 8.
- If **choice = 2**, go to Step 11.

Step 8: Accept Daily Input reads

- Sleep hours
- Productivity rating
- Stress rating
- Mood rating

Step 9: Calculate Life Score

Use formula:

```
Sleep Score       = (sleep / 8) × 25

Productivity Score = (productivity / 10) × 25

Mood Score         = (mood / 10) × 25

Stress Score       = ((10 - stress) / 10) × 25

Life Score = sum of all 4
```

Step 10: Append Entry to File

Open `users_log.txt` in append mode.
Write:

```
username sleep productivity stress mood score
```

Return to **Step 6** (User Menu).

Step 11: Load All User Entries

Reload file.
Filter entries belonging to the current user.

Step 12: Compute Analytics

From all user entries:

- Calculate **average score**
- Find **highest score**
- Find **lowest score**
- Determine **trend**
- Prepare **last 7 scores**

Step 13: Display ASCII Graph

```
Print "*" × (score / 5)
```

Show formatted graph lines.

Step 14: Display Suggestions

Based on average:

- If avg < 50 → Needs improvement
- If 50–75 → Moderate stability
- If > 75 → Great stability

Return to Step 6.

Step 15: Logout

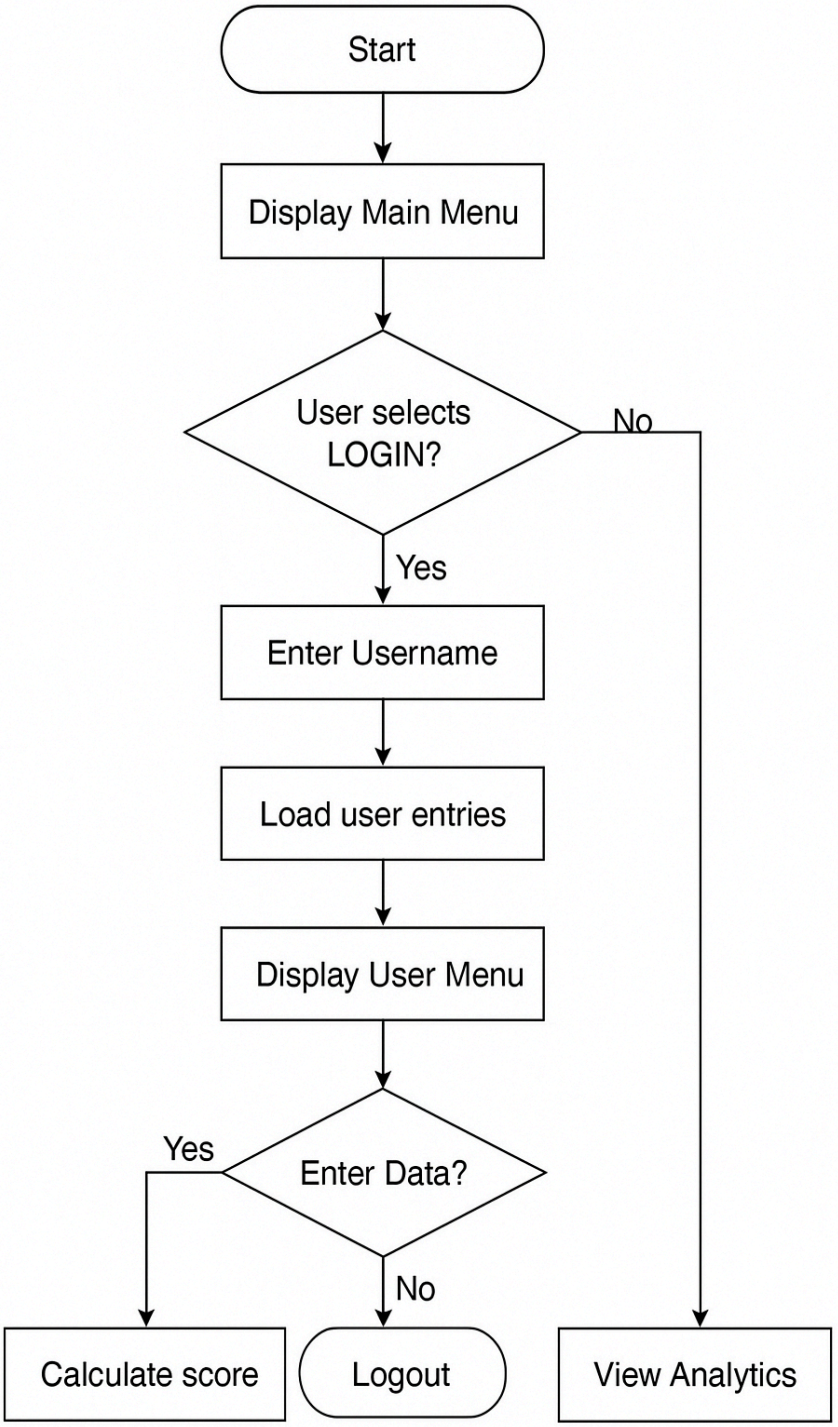
Return the user to the main menu.
Go to Step 2.

Step 16: Exit

Stop program execution.

Step 17: End.

FLOWCHART:

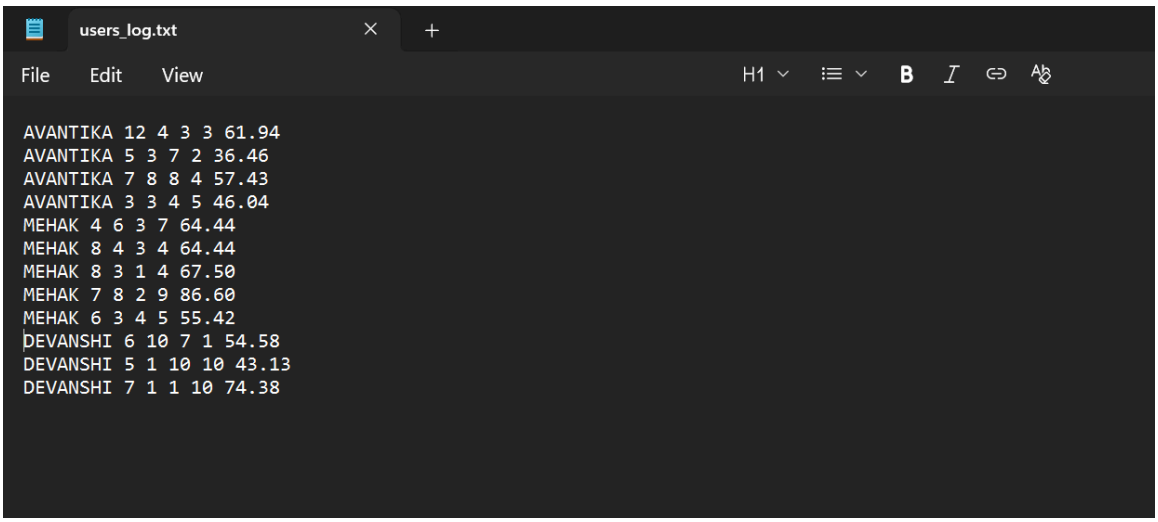

Start
Display Main Menu
User selects LOGIN?
No
Yes
Enter Username
Load user entries
Display User Menu
Yes
Enter Data?
No
Calculate score
Logout
View Analytics

IMPLEMENTATION

The project is implemented in the **C programming language** using a modular and menu-driven approach. A structure (`struct`)

```c
typedef struct
 {
    char username[USERNAME_LEN];
    int sleep;
    int productivity;
    int stress;
    int mood;
    double score;
} UserEntry;
```

is used to store each day's wellness data such as sleep, productivity, stress, mood, and the computed Life Score. All user entries are saved in a single file (`users_log.txt`)



using file handling operations like reading, writing, and appending.

The system supports **multiple users** by storing the username with every entry and filtering data based on the user who logs in. The

program includes two menus: a main menu for login and a user menu for entering data, viewing analytics, or logging out.

Life Score is calculated using a weighted formula where sleep, productivity, mood, and stress each contribute 25%. Stress is reverse-scaled so lower stress gives a higher score.

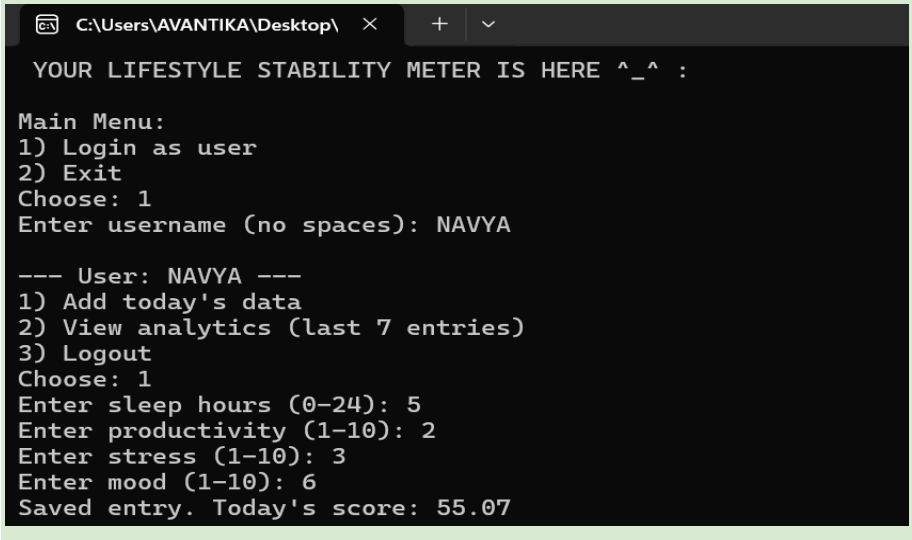
```
double calculateScore(int sleep, int prod, int stress, int mood) //calculates the score of the user
{
    double s = (sleep / 8.0) * 25.0; //each element has 25% of contribution
    if (s > 25.0) s = 25.0;

    double p = (prod / 10.0) * 25.0;
    double m = (mood / 10.0) * 25.0;
    double st = (1.0 - ((stress - 1.0) / 9.0)) * 25.0; //score works opposite for stress

    double score = s + p + m + st;
    if (score < 0)
    score = 0;  //to avoid data leaks
    if (score > 100)
     score = 100;
    return score;
}
```

Analytics are generated by loading all entries of the logged-in user and calculating average score, highest and lowest score, trend (improving, declining, or stable), and a simple ASCII graph to visualize progress. Suggestions are shown based on the user's average score.

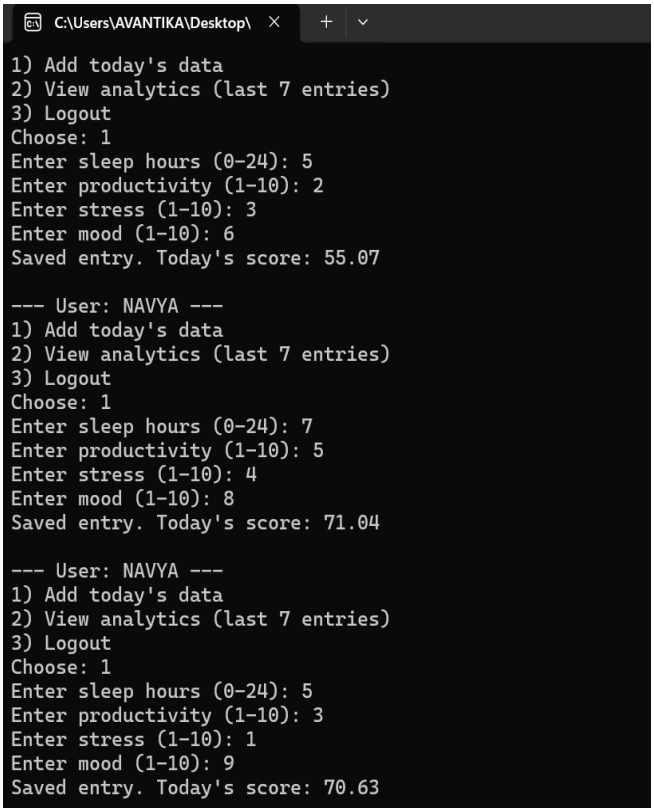
```
Last 3 entries for DEVANSHI:
Day  1 -> Sleep: 6 Prod:10 Stress: 7 Mood: 1 Score: 54.58
Day  2 -> Sleep: 5 Prod: 1 Stress:10 Mood:10 Score: 43.13
Day  3 -> Sleep: 7 Prod: 1 Stress: 1 Mood:10 Score: 74.38

Average (last 3): 57.36
Highest (last 3): 74.38
Lowest  (last 3): 43.13
Trend: Improving! KEEP IT UP BLUD :P (43.13 -> 74.38)


 YOUR SCORE GRAPH IS HERE ^_^ :
Day  1: ************** (54.58)
Day  2: *********** (43.13)
Day  3: ******************* (74.38)

Suggestions: Room for improvement.
- Strengthen hydration and consistent sleep.
```

TESTING AND VALIDIATION

Logged in as a user (navya)



User entering their daily data



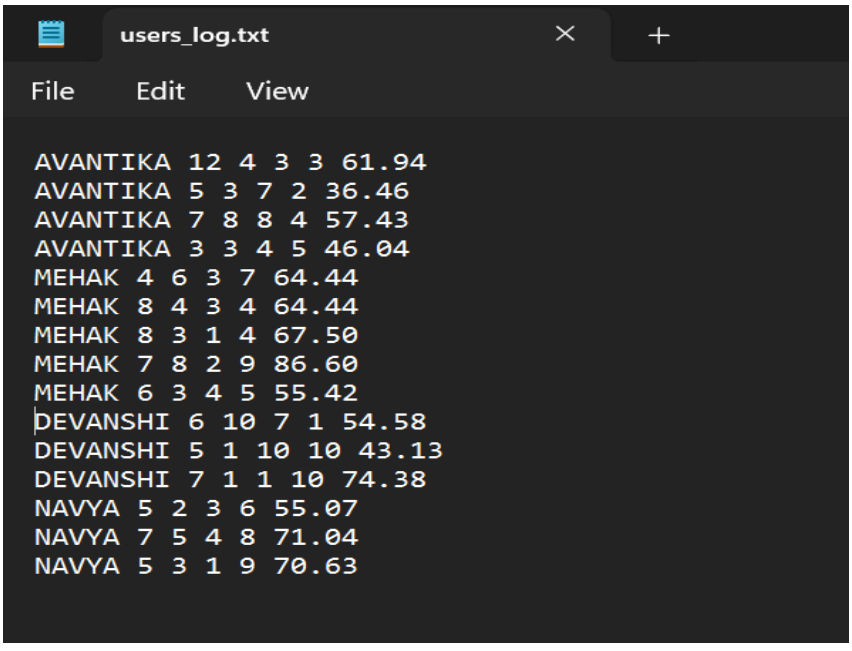
User entered an invalid choice

```
--- User: NAVYA ---
1) Add today's data
2) View analytics (last 7 entries)
3) Logout
Choose: 6
Invalid choice.
```

Analytics:

```
--- User: DEVANSHI ---
1) Add today's data
2) View analytics (last 7 entries)
3) Logout
Choose: 2

Last 3 entries for DEVANSHI:
Day  1 -> Sleep: 6 Prod:10 Stress: 7 Mood: 1 Score: 54.58
Day  2 -> Sleep: 5 Prod: 1 Stress:10 Mood:10 Score: 43.13
Day  3 -> Sleep: 7 Prod: 1 Stress: 1 Mood:10 Score: 74.38

Average (last 3): 57.36
Highest (last 3): 74.38
Lowest  (last 3): 43.13
Trend: Improving! KEEP IT UP BLUD :P (43.13 -> 74.38)


 YOUR SCORE GRAPH IS HERE ^_^ :
Day  1: ************** (54.58)
Day  2: *********** (43.13)
Day  3: ******************* (74.38)

Suggestions: Room for improvement.
- Strengthen hydration and consistent sleep.
```

Multiple user data saved in log:



Conclusion

The Lifestyle Stability Meter helps users track their daily well-being using four simple inputs: sleep, productivity, stress, and mood. By calculating a daily score and showing trends, the program allows users to understand their lifestyle patterns and make healthier choices. The project successfully demonstrates practical use of C programming concepts like structures, functions, and file handling to create a useful real-world application.

Future Work

In the future, the system can be improved by adding a graphical interface, storing data in the cloud, including more health parameters, and using advanced analytics for deeper insights. Features like reminders, mobile app support, and data security can also enhance the user experience.

REFERENCES

1. Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language. Prentice Hall.

2. Lafore, R. (1997). Object-Oriented Programming in C++. Sams Pub-lishing.

3. Online resources and documentation.

THANKYOU