UNIVERSITY OF CENTRAL PUNJAB

FALL 2025

Course Title: object oriented programming
Course Code: CP221

Project

Course Instructor: Chudhary Muhammad Owais		
Section: AC1	Program: BSAI	Date: 8/01/2026
Submission Date: 8/01/2026	Maximum Marks:	
Program Objective:	Course Objective:	Course Learning Objective:
TO BE FILLED IN BY THE STUDENT		
Student Name: Khurram Ullah Khan Muhammad Ilyas	Registration No: L1F24BSAI0038 L1F24BSAI0078	Sr. No:

Project Title: Mental Health & Mood Tracker v6.5

Table of Contents

PROJECT REPORT: Mental Health & Mood Tracker v6.5

2. Introduction

Brief Overview: The **Mental Health Tracker** is a console-based application written in C++. In today's busy student life, mental health is often ignored. This project allows users to log their daily moods, chat with a basic AI therapist, and receive random advice based on how they are feeling. It uses file handling to save data permanently so users can see their progress over time.

Problem Statement Many students feel stressed, anxious, or lonely but hesitate to talk to real people or doctors. They need a private, safe space to track their emotions and see if they are improving or getting worse over time without fear of judgment.

Purpose and Objectives

To apply Object Oriented Programming (OOP) concepts in a real-world scenario.

To create a system where users can self-reflect on their daily mood.

To provide simple, automated advice based on the user's current feelings.

To implement data persistence using File Handling.

3. Scope of the Project

What the System Can Do

User Management: Allows new users to register and existing users to login securely.

Mood Tracking: Users can rate their mood (1-10) and select keywords (e.g., Happy, Anxious).

Data Persistence: All history is saved in specific text files (e.g., khuram_history.txt).

Analysis: It can generate a report card and draw visual graphs using * # characters to show mood trends.

Social Feature: A unique feature to compare your mental health average with a friend using Operator Overloading.

What is Not Included

This is not a medical replacement for a real doctor.

It does not use a graphical user interface (GUI); it runs on the CLI (Command Line Interface).

Limitations

The AI chatbot uses pre-defined responses (if-else logic) and does not use real Machine Learning.

The "Random" advice generation relies on the provided time and math logic, not an external API.

4. Requirements

4.1 Functional Requirements

Registration: System must accept username, ID, and password and save them to users_db.txt.

Login: System must verify credentials before granting access.

Input Mood: Users must be able to input a score (1-10) and select a mood category.

Advice System: System must read advice from text files based on keywords.

Chatbot: A loop-based chat interface that responds to specific triggers (e.g., "sad", "exam", "exit").

History View: Display all past sessions in a list format.

Visual Graphs:

Graph 1: Shows mood scores *before* advice (using *).

Graph 2: Shows mood scores *after* advice (using #).

Delete history: user can delete his saved history.

Comparison: Compare the average mood score of the current user vs. a friend.

5. Object Oriented Concepts Used

Here is how we applied OOP concepts in this project:

1. Classes & Objects we used classes to organize the code logically:

User: Handles user data, menus, and history.

MychatBot: Handles the chat interface logic.

MyAdvicer: Handles file reading and random advice generation.

2. Constructors & Destructors

Constructor: User(string n, int r, int p) initializes the user's details immediately upon login.

Destructor: ~User() is used to ensure proper cleanup when the object goes out of scope.

3. Inheritance we used **Hierarchical Inheritance** to categorize tasks:

MainTask (Parent Class): Contains the virtual function.

physicalTask (Child Class): Inherits from MainTask (for physical advice like exercise).

MindTask (Child Class): Inherits from MainTask (for mental advice like meditation).

4. Polymorphism we used **Run-time Polymorphism (Function Overriding)**. The function showCategory() is defined as virtual in the parent class. Based on the user's mood, a pointer MainTask* ptr points to either a physicalTask object or a MindTask object, calling the correct version of the function at runtime.

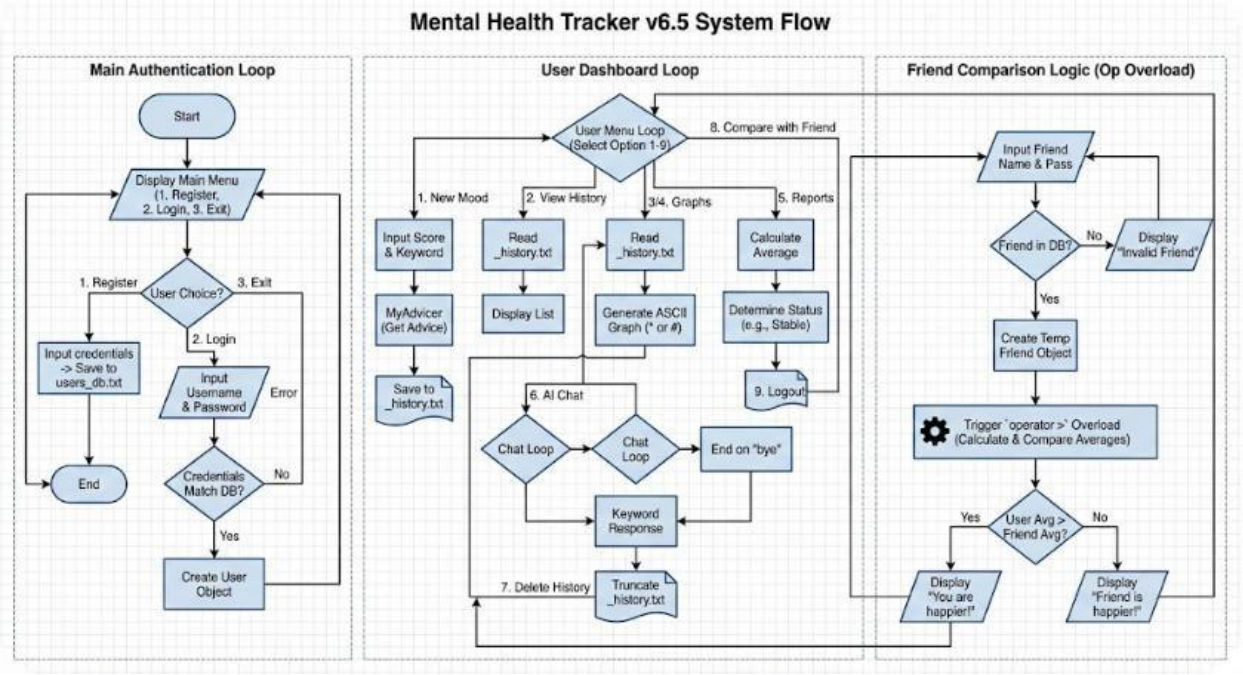
5. Association The User class creates objects of MychatBot and MyAdvicer inside its member functions. This represents a "Uses-A" relationship, as the User utilizes these components to get help.

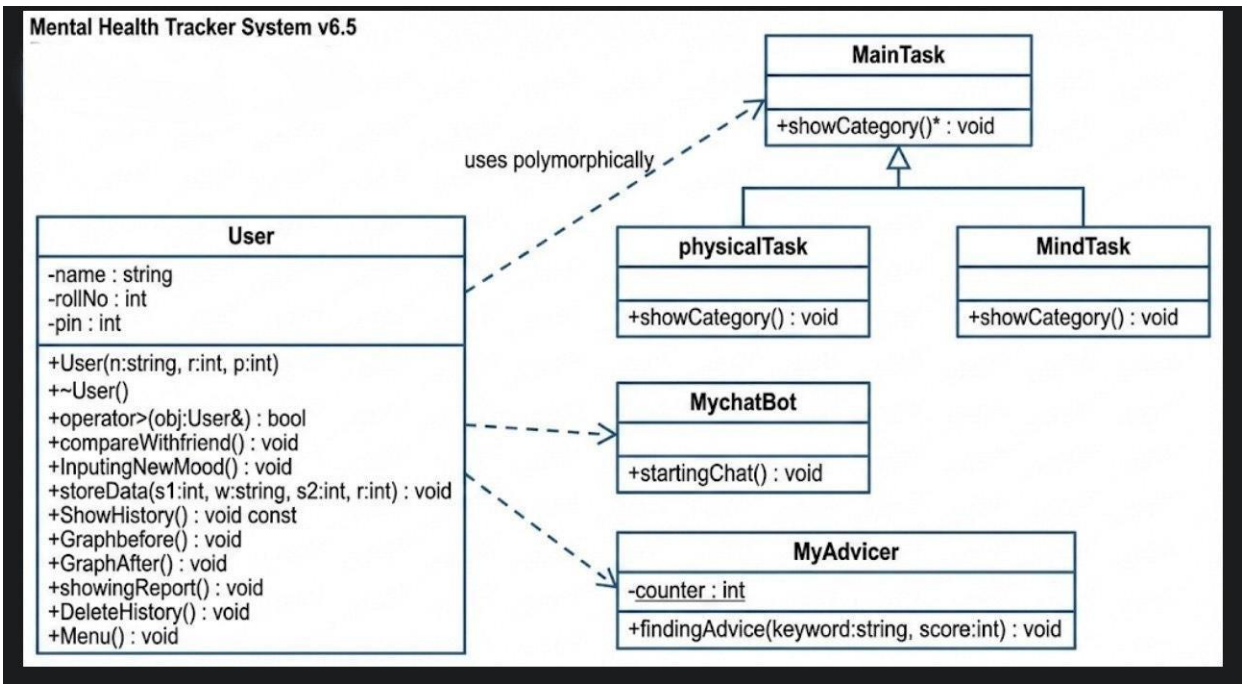
6. Static Members we used static int counter in the MyAdvicer class. This variable retains its value between function calls to ensure the random number generation (advice selection) changes every time.

7. Operator Overloading we overloaded the Greater Than operator (>).

Logic: It reads the history files of two users, calculates their average mood scores, and returns true if the current user has a higher average than the friend.

6. UML Diagrams & Flow chart





7. Conclusion

Summary of Learning Outcomes Working on "Mental Health Tracker v6.5" clarified our concepts of C++. we learned how to connect data (File Handling) with logic (Classes). we specifically struggled with the "Compare with Friend" logic and graph specially in tabular but learned how to open multiple files simultaneously to calculate averages.

Key Achievements

Successfully implemented **Operator Overloading** for a practical feature.

Created a visual representation of data (Graphs) using simple loops and ASCII characters, making the data easy to read.

8. Future Enhancements

Password Encryption: Currently, passwords are saved in plain text. we would like to encrypt them for security.

Date & Time: we want to automatically save the exact date of the session instead of just the session count.

GUI: Converting this from a black console screen to a proper windowed application using C# or Python in the future.

9. References

Course Lectures: Slides on Polymorphism and File Handling.

Book: *Object-Oriented Programming in C++* by Robert Lafore.

Online: cplusplus.com for syntax reference regarding fstream.

10. Appendix A: Source Code

```
#include <iostream>
#include <fstream>
#include <string>
#include <iomanip>
using namespace std;

// array of main keywordds
string moodkeyWords[10] =  {
   "Depressed", "Anxious", "Stressed", "Angry", "Tired",
   "Neutral", "Calm", "Focused", "Happy", "Motivated"
};

//mychatbot classs
class MychatBot {
public:
   void startingChat()
   {
      string userLine;
      cout << endl;
      cout <<
"=========================================================
=======" << endl;
      cout << "          AI THERAPIST SESSION (Type 'bye' to exit)       "
<< endl;
      cout <<
"=========================================================
=======" << endl;
      cout << "AI : Hello. I am here to listen. How are you feeling today?"
<< endl;

      cin.ignore(); // clearing buffer
```

```
        while(true) {
            cout << endl;
            cout << "You: ";
            getline(cin, userLine);
            if(userLine == "bye" || userLine == "exit" || userLine == "goodbye"
|| userLine == "stop") {
                cout << "AI : Take care of yourself. Remember, you matter." <<
endl;
                cout <<
"=============================================================
=======" << endl;
                break;
            }else if(userLine.find("die") != string::npos ||
userLine.find("suicide") != string::npos || userLine.find("kill") !=
string::npos) {
                cout << "AI : I am concerned. Your life has value." << endl;
                cout << "     Please talk to a real person or doctor immediately."
<< endl;
            }else if(userLine.find("exam") != string::npos ||
userLine.find("study") != string::npos) {
                cout << "AI : Exams check memory, not intelligence. Do your
best and leave the rest." << endl;
            }else if(userLine.find("sad") != string::npos ||
userLine.find("cry") != string::npos) {
                cout << "AI : It's okay to cry. Tears clean the soul. Let it out." <<
endl;
            }else if(userLine.find("tired") != string::npos ||
userLine.find("sleep") != string::npos) {
                cout << "AI : You can't pour from an empty cup. Go get some
rest." << endl;
            }else if(userLine.find("happy") != string::npos ||
userLine.find("good") != string::npos) {
                cout << "AI : That is great! Hold onto this feeling." << endl;
            }else if(userLine.find("alone") != string::npos) {
```

```
            cout << "AI : You are never truly alone. Use this time to know
yourself." << endl;
        }else {
            cout << "AI : I hear you. Tell me more about that." << endl;
        }
    }
  }
};

// polymorpisom & inheritance
class MainTask {
public:
  virtual void showCategory() {
    cout << "   [ Category: General Purposes ]" << endl;
  }
};

class physicalTask : public MainTask {
public:
  void showCategory() override {
    cout << "   [ Category: Physical Activation ]" << endl;
  }
};

class MindTask : public MainTask {
public:
  void showCategory() override {
    cout << "   [ Category: Mental Focus & Relaxation ]" << endl;
  }
};

//my advicer class...
```

```
class MyAdvicer {
private:
   static int counter;
public:
   void findingAdvice(string keyword, int score) {
      string filename = keyword + ".txt";
      // opening the file to count the liness
      ifstream fileIn (filename);
      if(!fileIn.is_open()) {
         cout << "   >> Error: Cannot find advice file for: " << filename <<
endl;
         return;
      }
      int countLine = 0;
      string templine;
      while(getline(fileIn, templine)) {
         // to avoid counting empty lines
         if(templine != "") {
            countLine++;
         }
      }
      fileIn.close();
      if(countLine == 0) {
         cout << "   >> File is empty." << endl;
         return;
      }
      // asking the user for time to show random
      int h, m;
      cout << endl;
      cout << "   > To generate advice, enter current time (Hour Minute): ";
      cin >> h >> m;
```

```
    // logic for random line findibg
    long long unique = (h * 60) + m + (score * 13) + (counter * 7);
    int linetoshow = (unique % countLine) + 1;

    // this check is good but only works if countLine is correct
    if(linetoshow > countLine) {
       linetoshow = score;
    }

    // opening file again to read data
    ifstream fileRead(filename);
    int curr = 0;
    string data;
    bool found = false;
    while(getline(fileRead, data)) {
       // to avoid showing empty lines
       if(data != "") {
          curr++;
          if(curr == linetoshow) {
             cout << endl;
             cout << "   ------- YOUR PERSONAL ADVICE -------" <<
endl;
             cout << "   \"" << data << "\"" << endl;
             cout << "   ----------------------------------" << endl;
             found = true;
             break;
          }
       }
    }
    fileRead.close();
    if(!found) {
       cout << "   >> Error occurred while reading this file.\n";
```

```
        }
        counter++;
    }
};

int MyAdvicer::counter = 1;

// user class
class User {
    string name; int rollNo;  int pin;
public:
    //constructor  ..
    User(string n, int r, int p) {
        name = n;    rollNo = r;
        pin = p;
    }

    // destructor..
    ~User() { }
    // operator overloading... > for comparing avgerges
    bool operator > (User& obj) {
        // clculating my average
        string myFile = name + "_history.txt";
        ifstream f1(myFile);
        int s1, s2, r, sum1=0, c1=0;
        string w;
        if(f1.is_open()){
            while(f1 >> s1 >> w >> s2 >> r) {
                sum1+=s2;
                c1++;
            }
            f1.close();
```

```
        }
        float myAvg;
        if(c1 == 0) {
            myAvg = 0;
        } else {
            myAvg = sum1 / c1;
        }

        // aclculateing freind average
        string otherfile = obj.name +"_history.txt";
        ifstream f2(otherfile);
        int sum2=0, c2=0;
        if(f2.is_open()){
            while(f2 >> s1 >> w >> s2 >> r) {
                sum2+=s2;
                c2++;
            }
            f2.close();
        }
        float otherAvg;
        if(c2 == 0) {
            otherAvg = 0;
        } else {
            otherAvg = sum2 / c2;
        }
        cout << "\n   STATS: You (" << myAvg << ") vs " << obj.name << "
(" << otherAvg << ")" << endl;
        return myAvg > otherAvg;
    }

    // function for comparison
    void compareWithfriend() {
```

```
        string fname; int fpass;
        cout << endl;
        cout << "============= COMPARE WITH FRIEND
=============" << endl;
        cout << "   Enter friend username: ";
        cin >> fname;
        cout << "   Enter friend password: ";
        cin >> fpass;

        // verifying friend exists or not
        ifstream db("users_db.txt");
        string n_db; int id_db, p_db;
        bool found = false;

        while(db >> n_db >> id_db >> p_db) {
            if(n_db == fname && p_db == fpass) {
                found = true;
                // createing a temp object for the friend
                User friendUser(n_db, id_db, p_db);
                // using operatar overlod..to comparee
                if(*this > friendUser) {
                    cout << "   RESULT: You have a higher score than " << fname
<< "!\n   Keep it up." << endl;
                } else {
                    cout << "   RESULT: " << fname << " has a better mood
average.\n   Ask them for tips!" << endl;
                }
                break;
            }
        }
        db.close();
```

```
        if(!found) {
            cout << "   >> Error: Invalid friend username or password." << endl;
        }
        cout <<
"================================================" <<
endl;
    }

    void InputingNewMood()  {
        int startingScore; int endingScore;
        int cIndexno;
        int ratingno;
        cout << endl;
        cout << "============== NEW MOOD ENTRY
==============" << endl;
        cout << "   Rate your current mood (1-10): ";
        cin >> startingScore;
        if(startingScore < 1) {
            startingScore = 1;
        }
        if(startingScore > 10)  {
            startingScore = 10;
        }

        cout << "\n   Select your feeling:" << endl;
        cout << "   ------------------------------" << endl;
        for(int i=0; i<10; i++) {
            cout << "   " << i << ". " << left << setw(15) << moodkeyWords[i];
            if ((i+1) % 2 == 0) cout << endl; // split into 2 columns for better UI
        }
        cout << "   ------------------------------" << endl;
        cout << "   Choice (0-9): ";
```

```
        cin >> cIndexno;
        if(cIndexno<0||cIndexno>9){
            cIndexno = 5;
        }
        string selectedmood = moodkeyWords[cIndexno];
        // calling the adviser
        MyAdvicer a;
        a.findingAdvice(selectedmood, startingScore);
        // using polymorphism for category identryfing
        MainTask* ptr;
        if(selectedmood == "Motivated" || selectedmood == "Happy" ||
selectedmood == "Angry") {
            ptr = new physicalTask();
        } else {
            ptr = new MindTask();
        }
        ptr->showCategory();
        delete ptr;

        // feedback system
        char ans;
        cout << endl;
        cout << "   Did you read the advice and do the task? (y/n): ";
        cin >> ans;
        if(ans =='y' ||ans =='Y')  {
            cout << endl;
            cout << "   --- POST SESSION CHECK ---" << endl;
            cout << "   How are you feeling now? (1-10): ";
            cin >> endingScore;
            cout << "   Rate our advice system (1-5): ";
            cin >> ratingno;
```

```
            storeData(startingScore, selectedmood ,endingScore, ratingno);
            cout << endl;
            cout << "   >> Result: Mood changed from " << startingScore << " -
> " << endingScore << endl;
        } else {
            storeData(startingScore, selectedmood, startingScore, 0);
            cout << "   >> Data saved without post-session update." << endl;
        }
        cout << "=========================================="
<< endl;
    }

    void storeData(int s1, string w, int s2, int r)  {
        string filename = name +"_history.txt";
        ofstream myfile(filename,ios::app);
        if(myfile.is_open()) {
            myfile<<s1 << " "<<w <<" " << s2 <<" " << r<< endl;
            myfile.close();
        }
    }

    void ShowHistory() const {
        string filename = name +"_history.txt";
        ifstream myfile(filename);

        if(!myfile.is_open()) {
            cout << endl;
            cout << "   >> No history found."<< endl;
            return;
        }
        cout << endl;
        cout << "============= " << name << "'s SESSION HISTORY
```

```
==========" << endl;
        int sc1, sc2, rt;
        int c = 0;
        string kw;

        // UI Using setw for perfect alignment
        cout << left << setw(5) << "No" << setw(15) << "Mood" << setw(15)
<< "Score Change" << setw(10) << "Rating" << endl;
        cout << "-------------------------------------------------" << endl;
        while(myfile >> sc1 >> kw >> sc2 >> rt) {
            c++;
            string scoreStr = to_string(sc1) + "->" + to_string(sc2);
            string rateStr = to_string(rt) + "/5";
            cout << left << setw(5) << c << setw(15) << kw << setw(15) <<
scoreStr << setw(10) << rateStr << endl;
        }
        cout <<
"============================================================"
<< endl;
        myfile.close();
    }

    void Graphbefore()  {
        string filename = name +"_history.txt";
        ifstream myfile(filename);
        if(!myfile.is_open()) {
            cout << endl;
            cout << "\n   >> No history found!" << endl;
            return;
        }

        int arr[15] = {0};
```

```
        int total = 0;
        int sc1, sc2, rt;
        string kw;
        while(myfile >> sc1 >> kw >> sc2 >> rt)
        {
           if(total < 15) {
              arr[total] = sc1;
              total++;
           } else {
              for(int i=0; i<14; i++) {
                 arr[i] = arr[i+1];
              }
              arr[14] = sc1;
           }
        }
        myfile.close();

        cout << endl;
        cout << "========== GRAPH: MOOD BEFORE SESSIONS
==========" << endl;
        cout << " (High bars = Better mood)" << endl << endl;
        for(int h = 10; h >= 1; h--) {
           if(h < 10) {
              cout << " " << h << " | ";
           } else {
              cout << h << " | ";
           }
           for(int i = 0; i < total; i++) {
              if(arr[i] >= h) {
                 cout << " * ";
              }else {
                 cout << "   ";
```

```
            }
        }
        cout << endl;
    }
    cout << "    ";
    for(int i=0; i<total; i++) {
        cout << "---" ;
    }
    cout << endl;
}

void GraphAfter(){
    string filename = name +"_history.txt";
    ifstream myfile(filename);
    if(!myfile.is_open()) {
        cout << endl;
        cout << "   >> No history found" << endl;
        return;
    }

    int arr[15] = {0};
    int total = 0;
    int sc1, sc2, rt;
    string kw;
    while(myfile >> sc1 >> kw >> sc2 >> rt) {
        if(total < 15) {
            arr[total] = sc2;
            total++;
        } else {
            for(int i=0; i<14; i++) {
                arr[i] = arr[i+1];
            }
```

```
            arr[14] = sc2;
        }
    }
    myfile.close();

    cout << endl;
    cout << "========== GRAPH: MOOD AFTER SESSIONS
==========" << endl;
    cout << " (High bars = Better mood)" << endl << endl;
    for(int h = 10; h >= 1; h--) {
        if(h < 10) {
            cout << " " <<h<< " | ";
        }else {
            cout <<h<< " | ";
        }
        for(int i = 0; i < total; i++) {
            if(arr[i] >= h){
                cout << " # ";
            }else {
                cout << "   ";
            }
        }
        cout << endl;
    }
    cout << "    ";
    for(int i=0; i<total; i++) {
        cout << "---";
    }
    cout << endl;
}

void showingReport(){
```

```
    string filename = name +"_history.txt";
    ifstream myfile(filename);
    if(!myfile.is_open()) {
       cout << endl;
       cout << "   >> No data found!" << endl;
       return;
    }
    int sc1, sc2, rt;
    int c = 0;
    int sum = 0;
    string kw;

    while(myfile >> sc1 >> kw >> sc2 >> rt) {
       c++;
       sum += sc1;
    }
    myfile.close();
    if(c==0) return;
    float avg = (float)sum / c; // casting  to float
    cout << endl;
    cout << "========== PERFORMANCE REPORT
==========" << endl;
    cout << "   User       : " << name << endl;
    cout << "   Sessions   : " << c << endl;
    cout << "   Avg Mood   : " << avg << " / 10" << endl;
    cout << "   ------------------------------------" << endl;
    if(avg >= 7.0) {
       cout << "   Status     : Excellent!" << endl;
    }else if(avg >= 4.0) {
       cout << "   Status     : Stable." << endl;
    }else {
       cout << "   Status     : Needs Care." << endl;
```

```
        }
        cout << "=========================================="
<< endl;
    }

    void DeleteHistory(){
        char yes;
        cout << endl;
        cout << "   >> WARNING: Are you sure you want to delete ALL
history? (y/n): ";
        cin >> yes;

        if(yes=='y'||yes=='Y') {
            string filename = name + "_history.txt";
            ofstream myfile(filename, ios::out|ios::trunc);
            myfile.close();
            cout << "   >> Success: History deleted." << endl;
        }
    }

    void Menu() {
        int option = 0;
        MychatBot bot;

        while(option!=9) {
            cout << endl;
            cout << "=========================================="
<< endl;
            cout << "       MAIN MENU (" << name << ")       " << endl;
            cout << "=========================================="
<< endl;
            cout << "  1. New Mood Login" << endl;
```

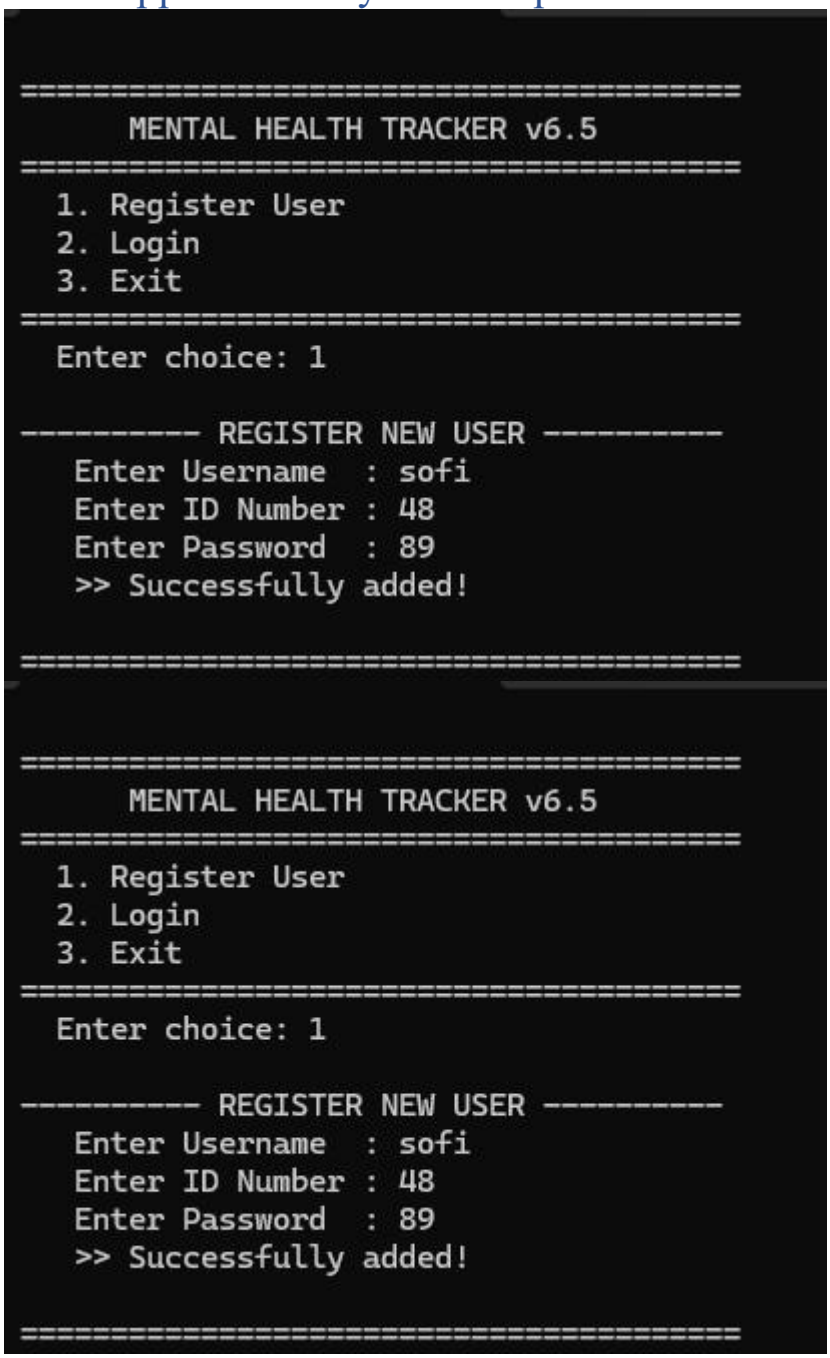
```
        cout << "  2. View History (List)" << endl;
        cout << "  3. View Graph (Before Sessions *)" << endl;
        cout << "  4. View Graph (After Sessions #)" << endl;
        cout << "  5. View Reports" << endl;
        cout << "  6. Chat with AI Therapist" << endl;
        cout << "  7. Delete History" << endl;
        cout << "  8. Compare with Friend" << endl;
        cout << "  9. Logout" << endl;
        cout << "=========================================="
<< endl;
        cout << "  Enter your choice: ";
        cin >> option;
        switch(option){
            case 1:
                InputingNewMood();
                break;
            case 2:
                ShowHistory();
                break;
            case 3:
                Graphbefore();
                break;
            case 4:
                GraphAfter();
                break;
            case 5:
                showingReport();
                break;
            case 6:
                bot.startingChat();
                break;
            case 7:
```

```
                DeleteHistory();
                break;
            case 8:
                compareWithfriend();
                break;
            case 9:
                cout << "   Logging out..." << endl;
                break;
            default:
                cout << "   >> Invalid choice entered!" << endl;
        }
    }
  }
};

// global functions...
void registeringUser() {
   string n; int r, p;
   cout << endl;
   cout << "---------- REGISTER NEW USER ----------" << endl;
   cout << "   Enter Username  : ";
   cin >> n;
   cout << "   Enter ID Number : ";
   cin >> r;
   cout << "   Enter Password  : ";
   cin >> p;
   ofstream database("users_db.txt", ios::app);
   if(database.is_open()) {
      database<<n<<" "<<r<<" "<<p<<endl;
      database.close();
      cout << "   >> Successfully added!" << endl;
   }
```

```
}

bool loginSystem() {
    string n; int p;
    cout << endl;
    cout << "------------- USER LOGIN -------------" << endl;
    cout << "   Enter Username : ";
    cin >> n;
    cout << "   Enter Password : ";
    cin >> p;
    ifstream database("users_db.txt");
    if(!database.is_open()) {
        cout << "   >> Error: No user database found." << endl;
        return false;
    }
    string namedb;
    int iddb, passdb;
    bool flag = false;
    while(database>> namedb>>iddb>>passdb) {
        if(namedb ==n && passdb== p) {
            flag = true;
            User u1(namedb, iddb, passdb);
            cout << endl;
            cout << "   >> Welcome, " << namedb << "!" << endl;
            u1.Menu();
            break;
        }
    }
    database.close();

    if(!flag) {
        cout << "   >> Error: Invalid password or username." << endl;
```

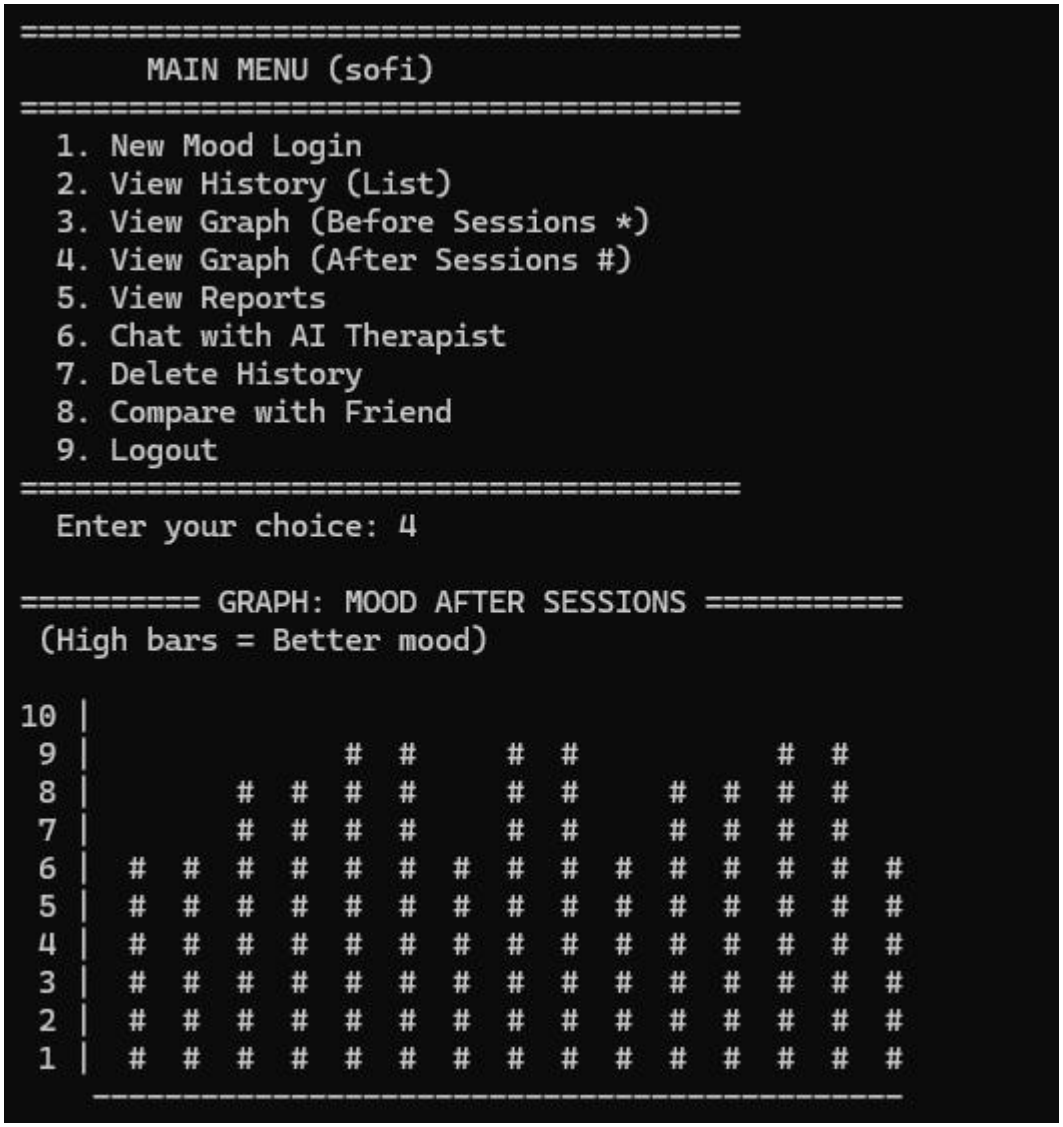
```
    }
    return flag;
}

int main() {
    int choice = 0;
    while(choice != 3)  {
        cout << endl;
        cout << "==========================================" <<
endl;
        cout << "      MENTAL HEALTH TRACKER v6.5        " << endl;
        cout << "==========================================" <<
endl;
        cout << "  1. Register User" << endl;
        cout << "  2. Login" << endl;
        cout << "  3. Exit" << endl;
        cout << "==========================================" <<
endl;
        cout << "  Enter choice: ";
        cin >> choice;
        if(choice == 1) {
            registeringUser();
        } else if(choice == 2) {
            loginSystem();
        }else if(choice == 3) {
            cout << "   Goodbye. Stay blessed." << endl;
        }
    }
    int n;
    cout<<"enter:";cin>>n;
    return 0;
}
```

11. Appendix B: System Output Screenshots



```
=====================================
      MENTAL HEALTH TRACKER v6.5
=====================================
  1. Register User
  2. Login
  3. Exit
=====================================
  Enter choice: 2

------------ USER LOGIN ------------
   Enter Username : sofi
   Enter Password : 88
   >> Error: Invalid password or username.

=====================================
```

```
=====================================
      MENTAL HEALTH TRACKER v6.5
=====================================
  1. Register User
  2. Login
  3. Exit
=====================================
  Enter choice: 2

------------ USER LOGIN ------------
   Enter Username : sofi
   Enter Password : 89

   >> Welcome, sofi!

=====================================
       MAIN MENU (sofi)
=====================================
  1. New Mood Login
  2. View History (List)
  3. View Graph (Before Sessions *)
  4. View Graph (After Sessions #)
  5. View Reports
  6. Chat with AI Therapist
  7. Delete History
  8. Compare with Friend
  9. Logout
=====================================
  Enter your choice:
```

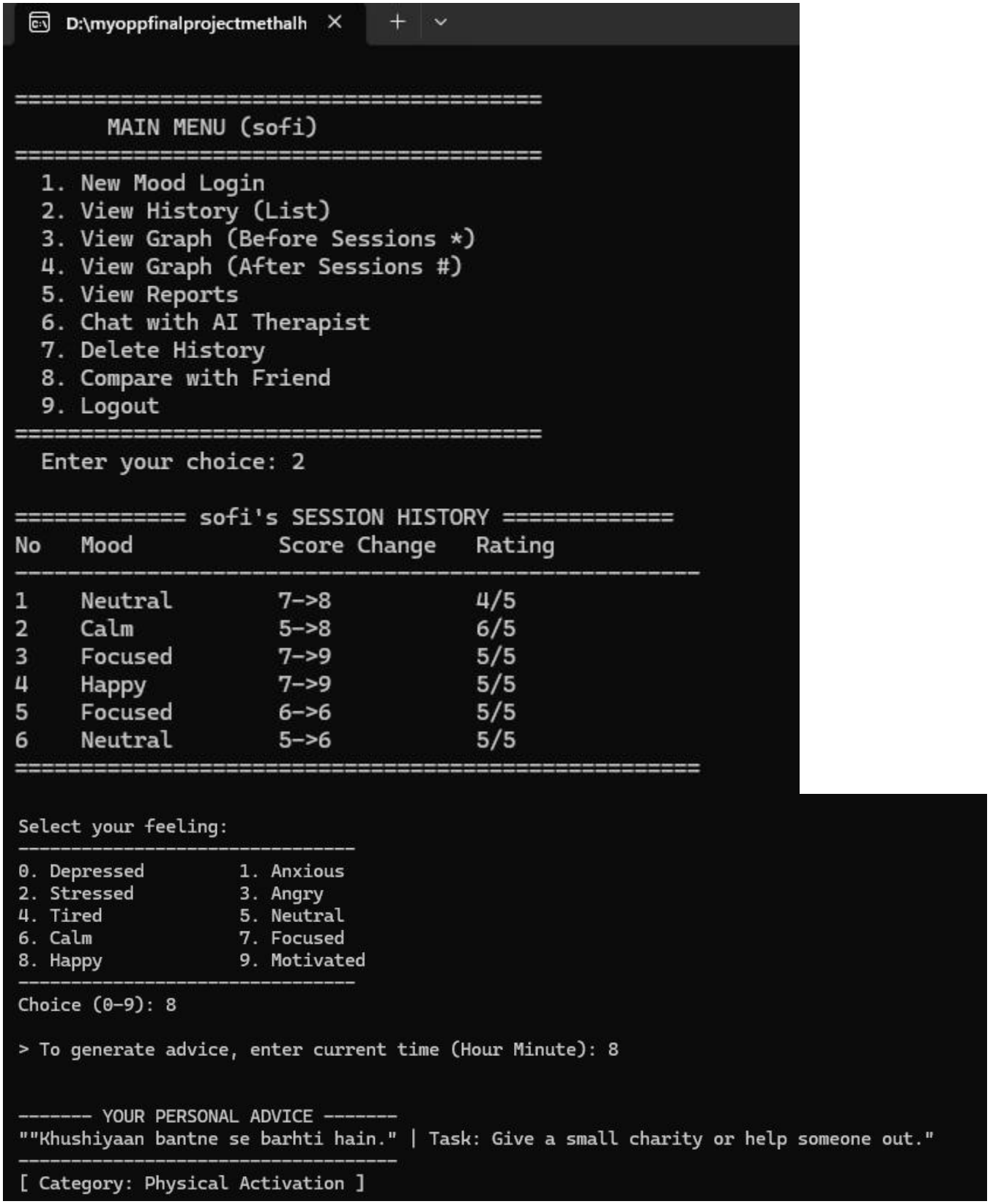

MAIN MENU (sofi)
1. New Mood Login
2. View History (List)
3. View Graph (Before Sessions *)
4. View Graph (After Sessions #)
5. View Reports
6. Chat with AI Therapist
7. Delete History
8. Compare with Friend
9. Logout
Enter your choice: 4
GRAPH: MOOD AFTER SESSIONS
(High bars = Better mood)
10
9
8
7
6
5
4
3
2
1

```
========================================
       MAIN MENU (sofi)
========================================
  1. New Mood Login
  2. View History (List)
  3. View Graph (Before Sessions *)
  4. View Graph (After Sessions #)
  5. View Reports
  6. Chat with AI Therapist
  7. Delete History
  8. Compare with Friend
  9. Logout
========================================
  Enter your choice: 3

========== GRAPH: MOOD BEFORE SESSIONS ==========
 (High bars = Better mood)

10 |
 9 |
 8 |
 7 |           *  *  *     *  *     *     *  *
 6 |  *        *  *  *  *  *  *  *  *     *  *  *
 5 |  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
 4 |  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
 3 |  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
 2 |  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
 1 |  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
    ----------------------------------------------
```

========== sofi's SESSION HISTORY ==========

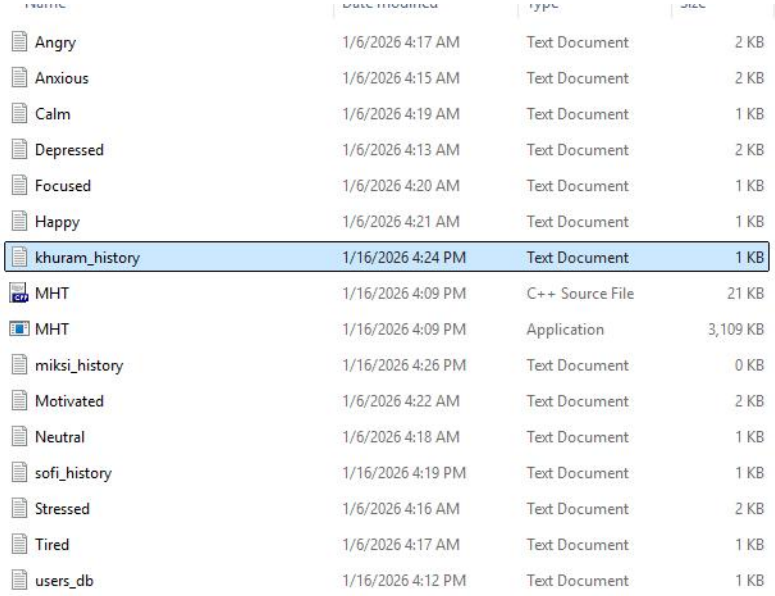
No	Mood	Score Change	Rating
1	Neutral	7->8	4/5
2	Calm	5->8	6/5
3	Focused	7->9	5/5
4	Happy	7->9	5/5
5	Focused	6->6	5/5
6	Neutral	5->6	5/5
7	Neutral	5->8	5/5
8	Focused	7->8	9/5
9	Focused	7->9	5/5
10	Happy	7->9	5/5
11	Focused	6->6	5/5
12	Focused	7->9	5/5
13	Happy	7->9	5/5
14	Focused	6->6	5/5
15	Neutral	7->8	4/5
16	Calm	5->8	6/5
17	Focused	7->9	5/5
18	Happy	7->9	5/5
19	Focused	6->6	5/5

==========


D:\myoppfinalprojectmethalh
MAIN MENU (sofi)
1. New Mood Login
2. View History (List)
3. View Graph (Before Sessions *)
4. View Graph (After Sessions #)
5. View Reports
6. Chat with AI Therapist
7. Delete History
8. Compare with Friend
9. Logout
Enter your choice: 2
sofi's SESSION HISTORY
No Mood Score Change Rating
1 Neutral 7->8 4/5
2 Calm 5->8 6/5
3 Focused 7->9 5/5
4 Happy 7->9 5/5
5 Focused 6->6 5/5
6 Neutral 5->6 5/5
Select your feeling:
0. Depressed 1. Anxious
2. Stressed 3. Angry
4. Tired 5. Neutral
6. Calm 7. Focused
8. Happy 9. Motivated
Choice (0-9): 8
> To generate advice, enter current time (Hour Minute): 8
YOUR PERSONAL ADVICE
""Khushiyaan bantne se barhti hain." | Task: Give a small charity or help someone out."
[Category: Physical Activation]

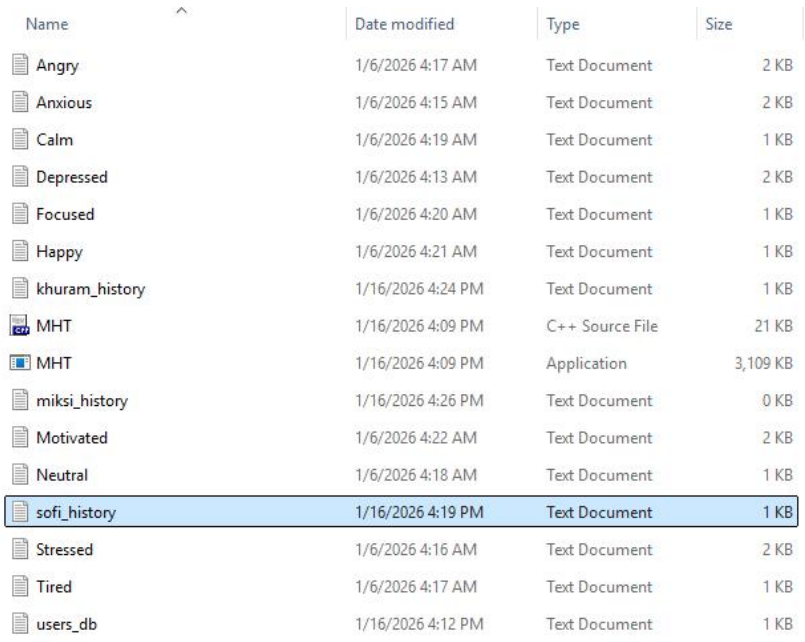
```
------- YOUR PERSONAL ADVICE -------
""Normal is boring." | Task: Try to learn one new word in a foreign language."
-----------------------------------
[ Category: Mental Focus & Relaxation ]

Did you read the advice and do the task? (y/n): y

--- POST SESSION CHECK ---
How are you feeling now? (1-10): 8
Rate our advice system (1-5): 4

>> Result: Mood changed from 7 -> 8
```

```
====================================
  1. New Mood Login
  2. View History (List)
  3. View Graph (Before Sessions *)
  4. View Graph (After Sessions #)
  5. View Reports
  6. Chat with AI Therapist
  7. Delete History
  8. Compare with Friend
  9. Logout
====================================
  Enter your choice: 1

============= NEW MOOD ENTRY =============
   Rate your current mood (1-10): 7

   Select your feeling:
   --------------------------------
   0. Depressed         1. Anxious
   2. Stressed          3. Angry
   4. Tired             5. Neutral
   6. Calm              7. Focused
   8. Happy             9. Motivated
   --------------------------------
   Choice (0-9): 5

   > To generate advice, enter current time (Hour Minute): 6
7

   ------- YOUR PERSONAL ADVICE -------
   ""Normal is boring." | Task: Try to learn one new word in a foreign language."
   -----------------------------------
   [ Category: Mental Focus & Relaxation ]

   Did you read the advice and do the task? (y/n):
```



Name	Date modified	Type	Size
Angry	1/6/2026 4:17 AM	Text Document	2 KB
Anxious	1/6/2026 4:15 AM	Text Document	2 KB
Calm	1/6/2026 4:19 AM	Text Document	1 KB
Depressed	1/6/2026 4:13 AM	Text Document	2 KB
Focused	1/6/2026 4:20 AM	Text Document	1 KB
Happy	1/6/2026 4:21 AM	Text Document	1 KB
khuram_history	1/16/2026 4:24 PM	Text Document	1 KB
MHT	1/16/2026 4:09 PM	C++ Source File	21 KB
MHT	1/16/2026 4:09 PM	Application	3,109 KB
miksi_history	1/16/2026 4:26 PM	Text Document	0 KB
Motivated	1/6/2026 4:22 AM	Text Document	2 KB
Neutral	1/6/2026 4:18 AM	Text Document	1 KB
sofi_history	1/16/2026 4:19 PM	Text Document	1 KB
Stressed	1/6/2026 4:16 AM	Text Document	2 KB
Tired	1/6/2026 4:17 AM	Text Document	1 KB
users_db	1/16/2026 4:12 PM	Text Document	1 KB

```
7 Neutral 8 4
5 Calm 8 6
7 Focused 9 5
7 Happy 9 5
6 Focused 6 5
5 Neutral 6 5
5 Neutral 8 5
7 Focused 8 9
7 Focused 9 5
7 Happy 9 5
6 Focused 6 5
7 Focused 9 5
7 Happy 9 5
6 Focused 6 5
7 Neutral 8 4
5 Calm 8 6
7 Focused 9 5
7 Happy 9 5
6 Focused 6 5
```



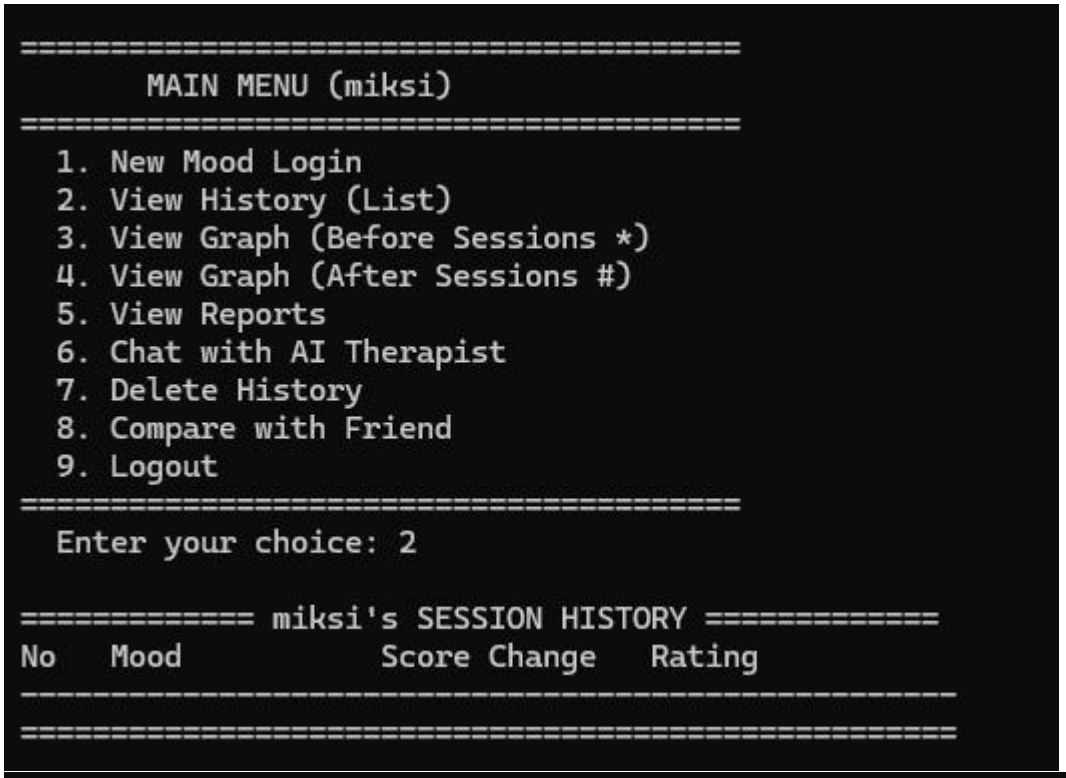
Name	Date modified	Type	Size
Angry	1/6/2026 4:17 AM	Text Document	2 KB
Anxious	1/6/2026 4:15 AM	Text Document	2 KB
Calm	1/6/2026 4:19 AM	Text Document	1 KB
Depressed	1/6/2026 4:13 AM	Text Document	2 KB
Focused	1/6/2026 4:20 AM	Text Document	1 KB
Happy	1/6/2026 4:21 AM	Text Document	1 KB
khuram_history	1/16/2026 4:24 PM	Text Document	1 KB
MHT	1/16/2026 4:09 PM	C++ Source File	21 KB
MHT	1/16/2026 4:09 PM	Application	3,109 KB
miksi_history	1/16/2026 4:26 PM	Text Document	0 KB
Motivated	1/6/2026 4:22 AM	Text Document	2 KB
Neutral	1/6/2026 4:18 AM	Text Document	1 KB
sofi_history	1/16/2026 4:19 PM	Text Document	1 KB
Stressed	1/6/2026 4:16 AM	Text Document	2 KB
Tired	1/6/2026 4:17 AM	Text Document	1 KB
users_db	1/16/2026 4:12 PM	Text Document	1 KB

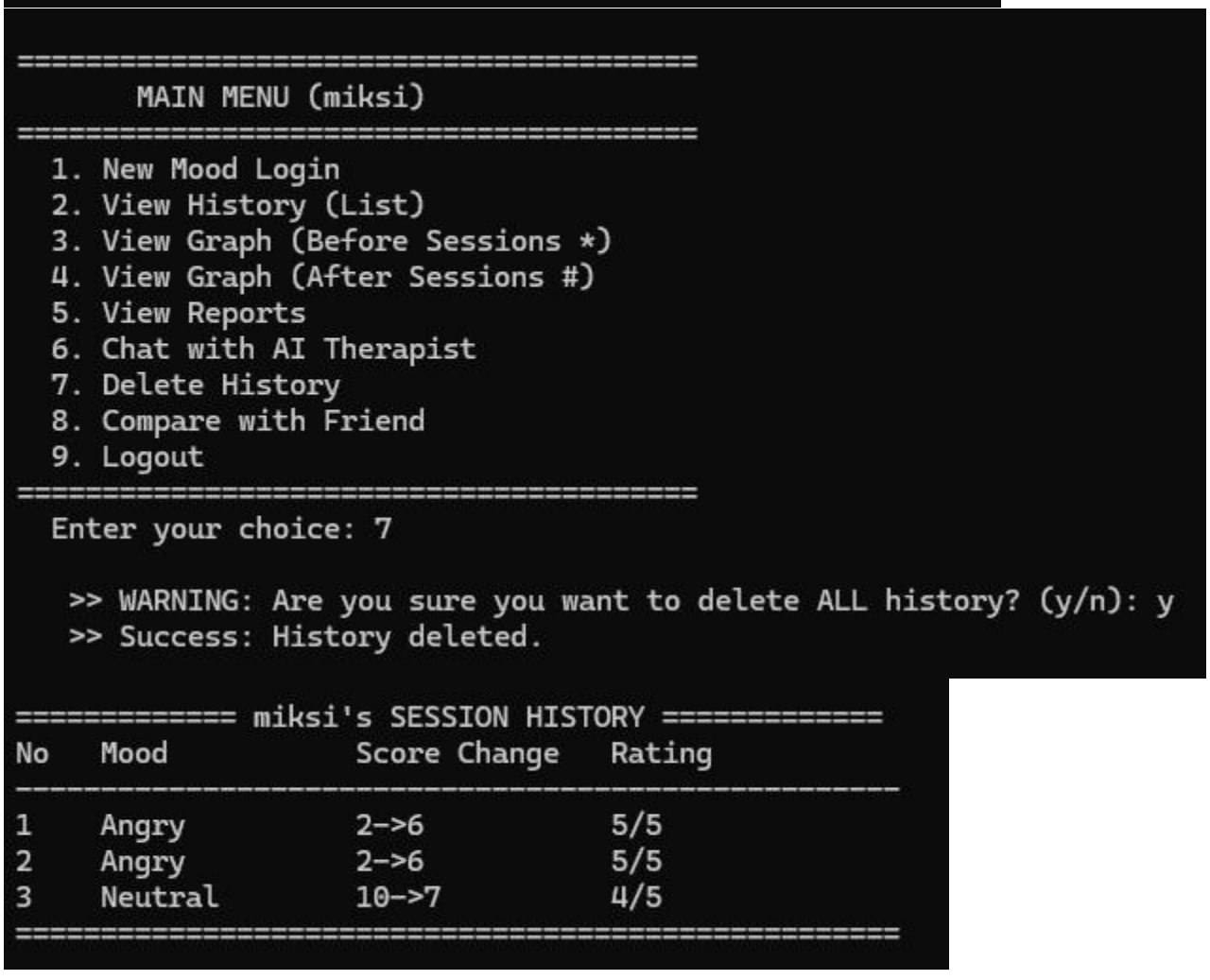
```
7 Neutral 8 4
5 Calm 8 6
7 Focused 9 5
7 Happy 9 5
6 Focused 6 5
5 Neutral 6 5
5 Neutral 8 5
7 Focused 8 9
7 Focused 9 5
7 Happy 9 5
6 Focused 6 5
7 Focused 9 5
7 Happy 9 5
6 Focused 6 5
7 Neutral 8 4
5 Calm 8 6
7 Focused 9 5
7 Happy 9 5
6 Focused 6 5
```

Name	Date modified	Type	Size
Angry	1/6/2026 4:17 AM	Text Document	2 KB
Anxious	1/6/2026 4:15 AM	Text Document	2 KB
Calm	1/6/2026 4:19 AM	Text Document	1 KB
Depressed	1/6/2026 4:13 AM	Text Document	2 KB
Focused	1/6/2026 4:20 AM	Text Document	1 KB
Happy	1/6/2026 4:21 AM	Text Document	1 KB
khuram_history	1/16/2026 4:24 PM	Text Document	1 KB
MHT	1/16/2026 4:09 PM	C++ Source File	21 KB
MHT	1/16/2026 4:09 PM	Application	3,109 KB
miksi_history	1/16/2026 4:26 PM	Text Document	0 KB
Motivated	1/6/2026 4:22 AM	Text Document	2 KB
Neutral	1/6/2026 4:18 AM	Text Document	1 KB
sofi_history	1/16/2026 4:19 PM	Text Document	1 KB
Stressed	1/6/2026 4:16 AM	Text Document	2 KB
Tired	1/6/2026 4:17 AM	Text Document	1 KB
users_db	1/16/2026 4:12 PM	Text Document	1 KB

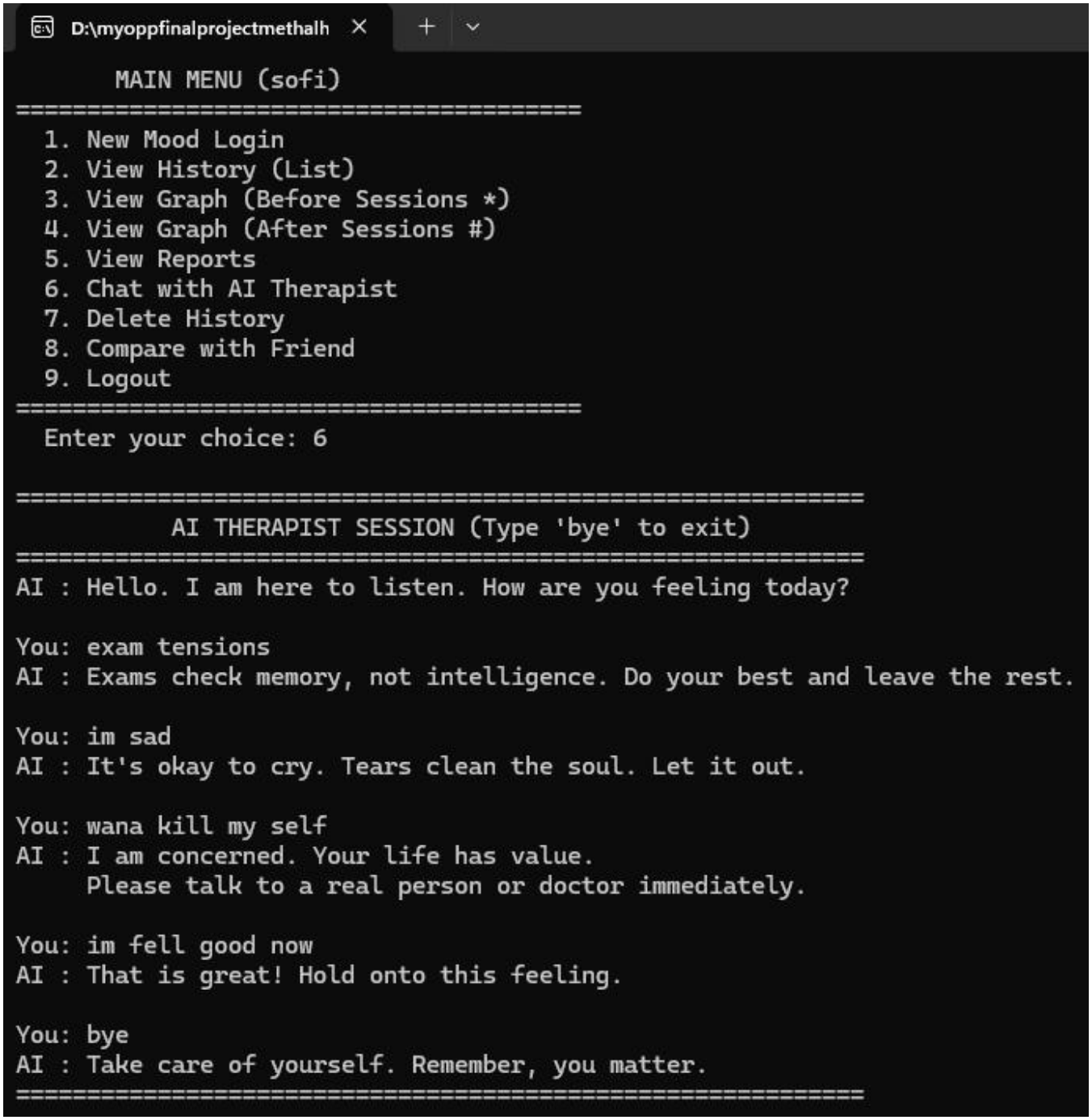
```
sofi 48 89
khuram 22 38
miksi 33 78
irtiza 33 33
```

```
---------- USER LOGIN ----------
  Enter Username : sofi
  Enter Password : 89

  >> Welcome, sofi!

========================================
     MAIN MENU (sofi)
========================================
 1. New Mood Login
 2. View History (List)
 3. View Graph (Before Sessions *)
 4. View Graph (After Sessions #)
 5. View Reports
 6. Chat with AI Therapist
 7. Delete History
 8. Compare with Friend
 9. Logout
========================================
 Enter your choice: 8

========== COMPARE WITH FRIEND ==========
  Enter friend username: khuram
  Enter friend password: 38

  STATS: You (7) vs khuram (7)
  RESULT: khuram has a better mood average.
  Ask them for tips!
==========================================
```


MAIN MENU (miksi)
1. New Mood Login
2. View History (List)
3. View Graph (Before Sessions *)
4. View Graph (After Sessions #)
5. View Reports
6. Chat with AI Therapist
7. Delete History
8. Compare with Friend
9. Logout
Enter your choice: 2
miksi's SESSION HISTORY
No Mood Score Change Rating


MAIN MENU (miksi)
1. New Mood Login
2. View History (List)
3. View Graph (Before Sessions *)
4. View Graph (After Sessions #)
5. View Reports
6. Chat with AI Therapist
7. Delete History
8. Compare with Friend
9. Logout
Enter your choice: 7
>> WARNING: Are you sure you want to delete ALL history? (y/n): y
>> Success: History deleted.
miksi's SESSION HISTORY
No Mood Score Change Rating
1 Angry 2->6 5/5
2 Angry 2->6 5/5
3 Neutral 10->7 4/5

```
=======================================
       MAIN MENU (khuram)
=======================================
  1. New Mood Login
  2. View History (List)
  3. View Graph (Before Sessions *)
  4. View Graph (After Sessions #)
  5. View Reports
  6. Chat with AI Therapist
  7. Delete History
  8. Compare with Friend
  9. Logout
=======================================
  Enter your choice: 5

========== PERFORMANCE REPORT ==========
   User       : khuram
   Sessions   : 20
   Avg Mood   : 6.5 / 10
   ---------------------------------
   Status     : Stable.
=======================================
```


D:\myoppfinalprojectmethalh
MAIN MENU (sofi)
1. New Mood Login
2. View History (List)
3. View Graph (Before Sessions *)
4. View Graph (After Sessions #)
5. View Reports
6. Chat with AI Therapist
7. Delete History
8. Compare with Friend
9. Logout
Enter your choice: 6
AI THERAPIST SESSION (Type 'bye' to exit)
AI : Hello. I am here to listen. How are you feeling today?
You: exam tensions
AI : Exams check memory, not intelligence. Do your best and leave the rest.
You: im sad
AI : It's okay to cry. Tears clean the soul. Let it out.
You: wana kill my self
AI : I am concerned. Your life has value.
Please talk to a real person or doctor immediately.
You: im fell good now
AI : That is great! Hold onto this feeling.
You: bye
AI : Take care of yourself. Remember, you matter.

```
 >> Welcome, miksi!

=====================================
     MAIN MENU (miksi)
=====================================
1. New Mood Login
2. View History (List)
3. View Graph (Before Sessions *)
4. View Graph (After Sessions #)
5. View Reports
6. Chat with AI Therapist
7. Delete History
8. Compare with Friend
9. Logout
=====================================
Enter your choice: 9
 Logging out...

=====================================
    MENTAL HEALTH TRACKER v6.5
=====================================
1. Register User
2. Login
3. Exit
=====================================
Enter choice: 3
 Goodbye. Stay blessed.
```