A

PROJECTREPORT

ON

**BLOOD GLUCOSE METER**

SUBMITTED BY :

**MISS. Patil Gayatri Vitthal(2124UCEF1019)**

SUBJECT :

**C++ PROGRAMMING**

Under the guidance of

**Miss. Ishwari Tirse**



Department of computer Science and Engineering

Sanjivani Rural Education Society’s

**SANJIVANI UNIVERSITY**

KOPARGOAN – 423603, DIST : AHMEDNAGAR

2024-2025

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **CONTENT** | **PAGE NO.** |
| **1.** | **INTRODUCTION** | **1** |
| **2.** | **CODE** | **2 - 5** |
| **3.** | **OUTPUT** | **6 - 7** |
| **4.** | **CONCLUSION** | **8** |

**INTRODUCTION**

Patients must measure their blood glucose 2-10 times a day using blood glucose meters. For people with diabetes, these monitors are an even more crucial part of their toolkit in monitoring blood sugar levels to control the condition properly. It helps users to make smarter choices about their diet, physical activity and use of medication in case they suffer from diabetes which eventually minimizes the risk for diabetic complications.

The miniature device that uses these test strips is a blood glucose meter. The user will prick their finger to gather the sample of blood, which is put on a test strip in the meter. It is equipped with electrochemical sensors that read the blood glucose level and deliver a consistent reading in seconds.

Improvements in technology over recent years have also given way to more advanced meters, such as continuous glucose monitors (CGMs), which can provide data in real time. These make both life easier for end users as well improve the quality of diabetes management generally.

**CODE**

#include <iostream>

#include <string>

#include <vector>

#include <algorithm>

#include <iomanip>

using namespace std;

const int MAX\_READINGS = 100;

const int MIN\_GLUCOSE = 40;

const int MAX\_GLUCOSE = 400;

struct GlucoseReading {

int value;

string date;

string time;

};

void displayMenu();

void addReading(vector<GlucoseReading>& readings);

void viewReadings(const vector<GlucoseReading>& readings);

void calculateAverage(const vector<GlucoseReading>& readings);

void deleteReading(vector<GlucoseReading>& readings);

int main() {

vector<GlucoseReading> readings;

int choice;

while (true) {

displayMenu();

cin >> choice;

switch (choice) {

case 1:

addReading(readings);

break;

case 2:

viewReadings(readings);

break;

case 3:

calculateAverage(readings);

break;

case 4:

deleteReading(readings);

break;

case 5:

return 0;

default:

cout << "Invalid choice. Please choose again." << endl;

}

}

return 0;

}

void displayMenu() {

cout << "Blood Glucose Meter Menu" << endl;

cout << "------------------------" << endl;

cout << "1. Add Glucose Reading" << endl;

cout << "2. View Glucose Readings" << endl;

cout << "3. Calculate Average Glucose" << endl;

cout << "4. Delete Glucose Reading" << endl;

cout << "5. Exit" << endl;

cout << "Enter your choice: ";

}

void addReading(vector<GlucoseReading>& readings) {

GlucoseReading reading;

cout << "Enter glucose value (" << MIN\_GLUCOSE << "-" << MAX\_GLUCOSE << "): ";

cin >> reading.value;

while (reading.value < MIN\_GLUCOSE || reading.value > MAX\_GLUCOSE) {

cout << "Invalid value. Please enter again: ";

cin >> reading.value;

}

cout << "Enter date (YYYY-MM-DD): ";

cin >> reading.date;

cout << "Enter time (HH:MM): ";

cin >> reading.time;

readings.push\_back(reading);

cout << "Reading added successfully!" << endl;

}

void viewReadings(const vector<GlucoseReading>& readings) {

cout << "Glucose Readings" << endl;

cout << "----------------" << endl;

for (const auto& reading : readings) {

cout << "Value: " << reading.value << " mg/dL" << endl;

cout << "Date: " << reading.date << endl;

cout << "Time: " << reading.time << endl;

cout << endl;

}

}

void calculateAverage(const vector<GlucoseReading>& readings) {

double sum = 0;

for (const auto& reading : readings) {

sum += reading.value;

}

double average = sum / readings.size();

cout << "Average Glucose: " << fixed << setprecision(2) << average << " mg/dL" << endl;

}

void deleteReading(vector<GlucoseReading>& readings) {

int index;

cout << "Enter reading number to delete (1-" << readings.size() << "): ";

cin >> index;

while (index < 1 || index > readings.size()) {

cout << "Invalid index. Please enter again: ";

cin >> index;

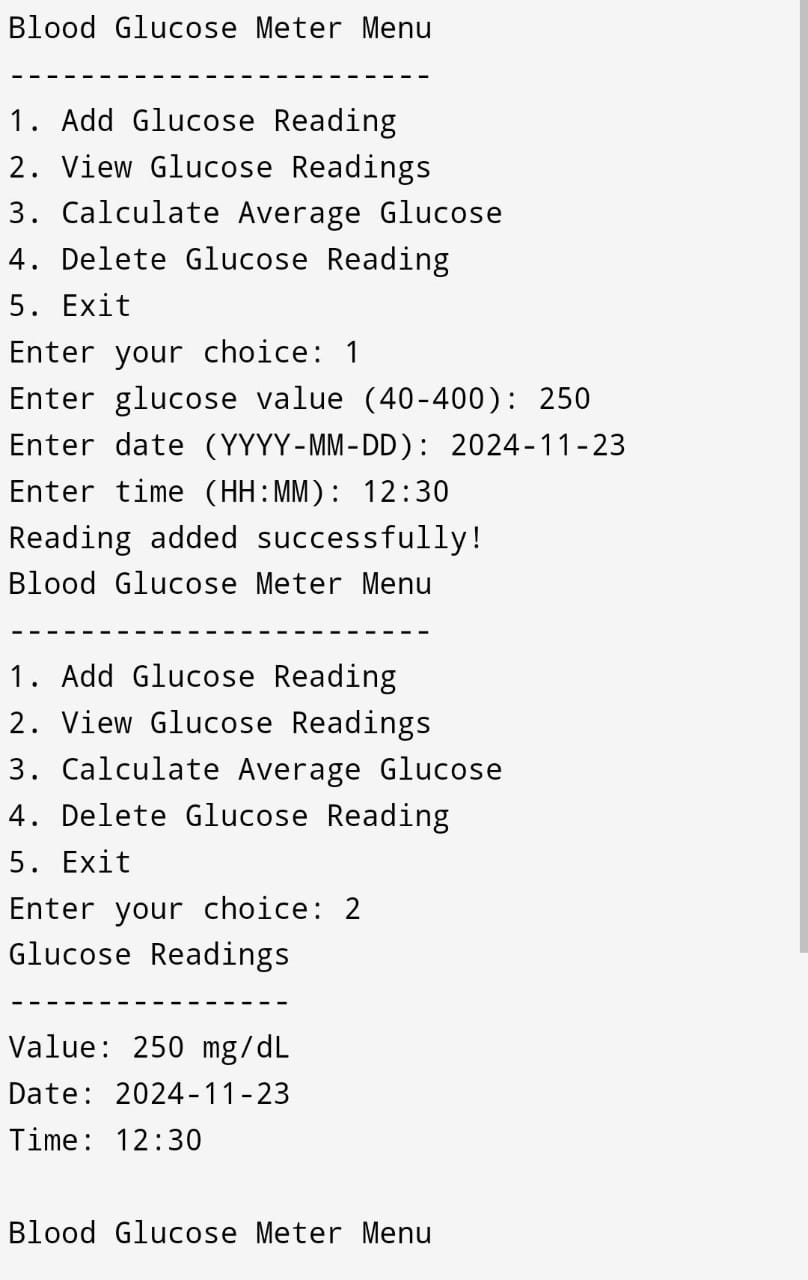
}

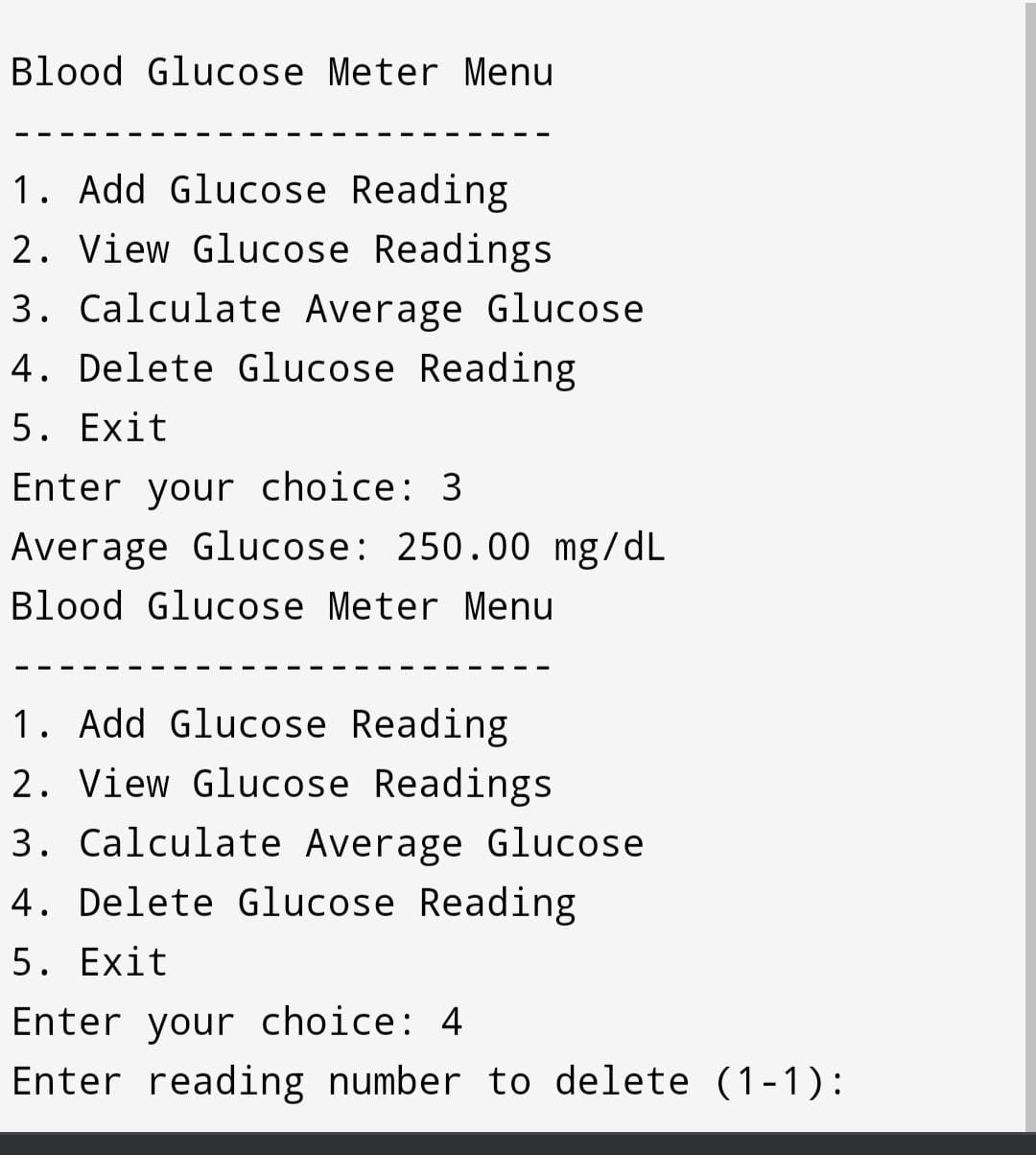
readings.erase(readings.begin() + (index - 1));

cout << "Reading deleted successfully!" << endl;

}

**OUTPUT**





**CONCLUSION**

Blood glucose meters are indispensable for those living with diabetes, facilitating timely and accurate monitoring of blood sugar levels. The ability to track glucose fluctuations empowers users to take proactive steps in managing their health, reducing the risk of complications associated with diabetes. As technology continues to advance, these devices are becoming increasingly sophisticated, improving usability and accuracy. Ultimately, blood glucose meters play a crucial role in enhancing the quality of life for individuals with diabetes, supporting them in their journey toward better health and well-being.