#include <SFML/Graphics.hpp>

#include <iostream>

#include <vector>

#include <string>

#include <algorithm>

using namespace std;

struct BuyerRecord {

string name;

string contact;

string address;

string product;

int quantitySoldThisMonth;

int quantitySoldLastMonth;

int totalStock;

float pricePerUnit;

float costPerUnit;

};

int main() {

vector<BuyerRecord> records = {

{"Ali", "0321", "Lahore", "Rice", 30, 20, 100, 120.0, 90.0},

{"Sara", "0300", "Karachi", "Oil", 50, 40, 120, 200.0, 170.0},

{"John", "0333", "Islamabad", "Flour", 20, 30, 80, 100.0, 85.0}

};

const int width = 1000, height = 700;

sf::RenderWindow window(sf::VideoMode(width, height), "Warehouse Sales & Profit Report");

sf::Font font;

if (!font.loadFromFile("arial.ttf")) {

cerr << "Error loading font! Make sure 'arial.ttf' is in the working directory.\n";

return 1;

}

int maxSold = 0;

for (const auto& r : records)

maxSold = max(maxSold, r.quantitySoldThisMonth + r.quantitySoldLastMonth);

const int barWidth = 60, spacing = 90, maxBarHeight = 250;

while (window.isOpen()) {

sf::Event event;

while (window.pollEvent(event)) {

if (event.type == sf::Event::Closed)

window.close();

}

window.clear(sf::Color::White);

for (size\_t i = 0; i < records.size(); ++i) {

const auto& r = records[i];

int totalSold = r.quantitySoldThisMonth + r.quantitySoldLastMonth;

int remaining = r.totalStock - totalSold;

float x = 70 + i \* (barWidth + spacing);

float barHeight = static\_cast<float>(totalSold) / maxSold \* maxBarHeight;

float y = height - barHeight - 180;

// Draw sales bar

sf::RectangleShape bar(sf::Vector2f(barWidth, barHeight));

bar.setPosition(x, y);

bar.setFillColor(sf::Color(100, 200, 250));

window.draw(bar);

// Info text below

vector<string> info = {

"Name: " + r.name,

"Contact: " + r.contact,

"Address: " + r.address,

"Product: " + r.product,

"This Month: " + to\_string(r.quantitySoldThisMonth),

"Last Month: " + to\_string(r.quantitySoldLastMonth),

"Remaining: " + to\_string(remaining)

};

for (int j = 0; j < info.size(); ++j) {

sf::Text txt(info[j], font, 12);

txt.setPosition(x - 10, height - 160 + j \* 18);

txt.setFillColor(sf::Color::Black);

window.draw(txt);

}

// Profit/Loss calculation

float netProfit = (r.pricePerUnit - r.costPerUnit) \* totalSold;

if (netProfit >= 0) {

sf::Text profitText("Profit: Rs " + to\_string(static\_cast<int>(netProfit)), font, 14);

profitText.setPosition(x, y - 30);

profitText.setFillColor(sf::Color::Green);

window.draw(profitText);

} else {

sf::Text lossText("Loss: Rs " + to\_string(static\_cast<int>(-netProfit)), font, 14);

lossText.setPosition(x, y - 30);

lossText.setFillColor(sf::Color::Red);

window.draw(lossText);

}

}

window.display();

}

return 0;

}