#include "imgui.h"

#include "backends/imgui\_impl\_sdl2.h"

#include "backends/imgui\_impl\_opengl3.h"

#include <SDL.h>

#include <SDL\_opengl.h>

#include <iostream>

#include <vector>

#include <string>

#include <cstdlib>

#include <ctime>

using namespace std;

struct Smartphone {

int id;

string name;

int budgetCategory;

vector<string> tags;

int stock;

int matchScore(const vector<string>& prefs) const {

int score = 0;

for (const auto& tag : prefs)

for (const auto& phoneTag : tags)

if (tag == phoneTag)

score++;

return score;

}

};

class SmartphoneBuilder {

public:

vector<Smartphone>& allPhones;

vector<string> softPrefs;

int budget;

SmartphoneBuilder(vector<Smartphone>& phones) : allPhones(phones), budget(0) {}

void setBudget(int b) { budget = b; }

void addSoftPref(const string& tag) { softPrefs.push\_back(tag); }

vector<Smartphone> build() {

vector<Smartphone> results;

int bestScore = 0;

for (const auto& phone : allPhones) {

if (phone.budgetCategory != budget) continue;

int score = phone.matchScore(softPrefs);

if (score > bestScore) bestScore = score;

}

for (const auto& phone : allPhones) {

if (phone.budgetCategory != budget) continue;

int score = phone.matchScore(softPrefs);

if (score == bestScore)

results.push\_back(phone);

}

return results;

}

};

int main() {

srand(time(0));

vector<Smartphone> phones = {

{1, "Apple iPhone 17 Air", 4, {"Apple", "Slim"}, rand()%10+1},

{2, "Samsung S25 Edge", 4, {"Snapdragon", "Slim"}, rand()%10+1},

{3, "Samsung S25+", 4, {"Snapdragon", "Camera"}, rand()%10+1},

{4, "Nothing Phone 2a", 2, {"Snapdragon", "Slim"}, rand()%10+1},

{5, "Samsung A56", 1, {"Budget", "Camera"}, rand()%10+1},

{6, "Redmi Note 13+", 1, {"Budget", "Camera"}, rand()%10+1},

{7, "Samsung M53", 2, {"Mediatek", "LongBattery"}, rand()%10+1},

{8, "Apple iPhone 15", 4, {"Apple", "Camera"}, rand()%10+1},

{9, "Apple iPhone 15+", 4, {"Apple", "Camera"}, rand()%10+1}

};

// SDL + OpenGL context

if (SDL\_Init(SDL\_INIT\_VIDEO | SDL\_INIT\_TIMER) != 0) return -1;

SDL\_Window\* window = SDL\_CreateWindow("Smartphone Recommender", SDL\_WINDOWPOS\_CENTERED, SDL\_WINDOWPOS\_CENTERED, 1000, 720, SDL\_WINDOW\_OPENGL | SDL\_WINDOW\_RESIZABLE);

SDL\_GLContext gl\_context = SDL\_GL\_CreateContext(window);

SDL\_GL\_MakeCurrent(window, gl\_context);

SDL\_GL\_SetSwapInterval(1);

IMGUI\_CHECKVERSION();

ImGui::CreateContext();

ImGuiIO& io = ImGui::GetIO(); (void)io;

ImGui::StyleColorsDark();

ImGui\_ImplSDL2\_InitForOpenGL(window, gl\_context);

ImGui\_ImplOpenGL3\_Init("#version 130");

// GUI State

int role = 0;

char adminPwd[32] = "";

bool loggedIn = false;

bool showCustomer = false;

bool showAdmin = false;

vector<Smartphone> recommendations;

int budget = 0, ss = 0, soc = 0, battery = 0, extras = 0;

int selectedPhoneId = -1;

char purchaseStatus[64] = "";

bool running = true;

SDL\_Event event;

while (running) {

while (SDL\_PollEvent(&event))

ImGui\_ImplSDL2\_ProcessEvent(&event);

if (event.type == SDL\_QUIT)

running = false;

ImGui\_ImplOpenGL3\_NewFrame();

ImGui\_ImplSDL2\_NewFrame();

ImGui::NewFrame();

ImGui::Begin("Smartphone Recommendation System");

if (!loggedIn) {

ImGui::Text("Login as:");

if (ImGui::RadioButton("Admin", role == 1)) role = 1;

if (ImGui::RadioButton("Customer", role == 2)) role = 2;

if (role == 1) {

ImGui::InputText("Admin Password", adminPwd, 32, ImGuiInputTextFlags\_Password);

if (ImGui::Button("Login")) {

if (string(adminPwd) == "admin123") {

loggedIn = true;

showAdmin = true;

}

}

} else if (role == 2) {

if (ImGui::Button("Proceed")) {

loggedIn = true;

showCustomer = true;

}

}

}

// Admin GUI

if (showAdmin) {

ImGui::Text("Admin Dashboard");

for (auto& phone : phones) {

ImGui::Text("%d. %s (Stock: %d)%s", phone.id, phone.name.c\_str(), phone.stock,

(phone.stock < 5 ? " [LOW STOCK]" : ""));

}

static int idToAdd = 0;

static int qtyToAdd = 0;

ImGui::InputInt("Phone ID", &idToAdd);

ImGui::InputInt("Quantity", &qtyToAdd);

if (ImGui::Button("Add Stock")) {

for (auto& phone : phones) {

if (phone.id == idToAdd) {

phone.stock += qtyToAdd;

break;

}

}

}

}

// Customer GUI

if (showCustomer) {

ImGui::Text("Customer Preferences:");

const char\* screenOptions[] = {"<6.5", "6.5–6.7", ">6.7", "Slim"};

const char\* socOptions[] = {"Apple", "Snapdragon", "Mediatek", "No preference"};

const char\* batteryOptions[] = {"LongBattery", "FastCharging", "EfficientUI", "No preference"};

const char\* budgetOptions[] = {"<25k", "25–30k", "30–50k", "50k+"};

const char\* extraOptions[] = {"Slim", "Camera", "Performance", "None"};

ImGui::Combo("Screen Size", &ss, screenOptions, IM\_ARRAYSIZE(screenOptions));

ImGui::Combo("SoC", &soc, socOptions, IM\_ARRAYSIZE(socOptions));

ImGui::Combo("Battery", &battery, batteryOptions, IM\_ARRAYSIZE(batteryOptions));

ImGui::Combo("Budget", &budget, budgetOptions, IM\_ARRAYSIZE(budgetOptions));

ImGui::Combo("Extras", &extras, extraOptions, IM\_ARRAYSIZE(extraOptions));

if (ImGui::Button("Get Recommendations")) {

SmartphoneBuilder builder(phones);

builder.setBudget(budget + 1);

if (soc == 0) builder.addSoftPref("Apple");

else if (soc == 1) builder.addSoftPref("Snapdragon");

else if (soc == 2) builder.addSoftPref("Mediatek");

if (battery == 0) builder.addSoftPref("LongBattery");

else if (battery == 1) builder.addSoftPref("FastCharging");

else if (battery == 2) builder.addSoftPref("EfficientUI");

if (extras == 0) builder.addSoftPref("Slim");

else if (extras == 1) builder.addSoftPref("Camera");

else if (extras == 2) builder.addSoftPref("Performance");

recommendations = builder.build();

strcpy(purchaseStatus, "");

}

if (!recommendations.empty()) {

ImGui::Separator();

ImGui::Text("Recommendations:");

for (const auto& phone : recommendations) {

ImGui::RadioButton((to\_string(phone.id) + ". " + phone.name + " (Stock: " + to\_string(phone.stock) + ")").c\_str(), &selectedPhoneId, phone.id);

}

if (ImGui::Button("Buy Selected")) {

bool found = false;

for (auto& phone : recommendations) {

if (phone.id == selectedPhoneId && phone.stock > 0) {

phone.stock--;

snprintf(purchaseStatus, 64, "Order placed for: %s", phone.name.c\_str());

found = true;

break;

}

}

if (!found) {

strcpy(purchaseStatus, "Invalid ID selected or out of stock.");

}

}

ImGui::Text("%s", purchaseStatus);

}

}

ImGui::End();

ImGui::Render();

glClearColor(0.1f, 0.1f, 0.15f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

ImGui\_ImplOpenGL3\_RenderDrawData(ImGui::GetDrawData());

SDL\_GL\_SwapWindow(window);

}

// Cleanup

ImGui\_ImplOpenGL3\_Shutdown();

ImGui\_ImplSDL2\_Shutdown();

ImGui::DestroyContext();

SDL\_GL\_DeleteContext(gl\_context);

SDL\_DestroyWindow(window);

SDL\_Quit();

return 0;

}