HMI Skin Customization System

Objective: Implement a system to manage different themes or skins for an HMI display.

Requirements:

Create a Theme class:

Attributes: backgroundColor, fontColor, fontSize, and iconStyle.

Implement methods to apply and display the theme.

Store multiple themes using std::map:

Example themes: Classic, Sport, and Eco.

Create a simple user interface to switch between themes:

Use console input to select a theme.

Display a preview of the selected theme on the console (e.g., "Sport Theme: Red Background, White Font").

Deliverables:

A C++ program that demonstrates theme switching.

Output displaying applied themes.

General Instructions:

Ensure your code follows OOP principles (e.g., use of classes, inheritance, and encapsulation).

Comment your code to explain each step and decision.

Include a README file with instructions for running the program.

Program:

#include <iostream>

#include <string>

#include <map>

using namespace std;

class Theme

{

public:

string backgroundColor;

string fontColor;

int fontSize;

string iconStyle;

Theme() : backgroundColor("White"), fontColor("Black"), fontSize(14), iconStyle("Flat") {}

Theme(string bgColor, string fColor, int fSize, string iStyle)

: backgroundColor(bgColor), fontColor(fColor), fontSize(fSize), iconStyle(iStyle) {}

void displayPreview() const

{

cout << "Theme Preview: \n";

cout << "Background Color: " << backgroundColor << "\n";

cout << "Font Color: " << fontColor << "\n";

cout << "Font Size: " << fontSize << "px\n";

cout << "Icon Style: " << iconStyle << "\n";

}

};

void displayThemeOptions(const map<string, Theme>& themes)

{

cout << "Available Themes:\n";

for (const auto& theme : themes)

{

cout << "- " << theme.first << "\n";

}

cout << "\n";

}

int main()

{

Theme classicTheme("White", "Black", 14, "Flat");

Theme sportTheme("Red", "White", 16, "Bold");

Theme ecoTheme("Green", "Dark Green", 12, "Minimal");

map<string, Theme> themes =

{

{"Classic", classicTheme},

{"Sport", sportTheme},

{"Eco", ecoTheme}

};

string selectedTheme;

displayThemeOptions(themes);

cout << "Enter the theme you want to apply: ";

getline(cin, selectedTheme);

if (themes.find(selectedTheme) != themes.end())

{

cout << "\nYou have selected the " << selectedTheme << " theme:\n";

themes[selectedTheme].displayPreview();

}

else

{

cout << "Invalid theme selection.\n";

}

return 0;

}

Output:

Available Themes:

- Classic

- Eco

- Sport

Enter the theme you want to apply: Eco

You have selected the Eco theme:

Theme Preview:

Background Color: Green

Font Color: Dark Green

Font Size: 12px

Icon Style: Minimal