



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



CONSOLE- BASED REAL ESTATATE MANAGEMENT SYSTEM

A Final Output in Programming

Presented to the

Subject Instructor of Electronics Engineering

Institute of Engineering

Marikina Polytechnic College Shoe

Avenue, Marikina City

In Final Requirement of the Subject Programming 1 for the Degree of Bachelor
of Science in Electronics Engineering

By:

Alulod, Denmark Warrene P.

Perillo. Alvin F.

December 14, 2023



I. INTRODUCTION OF THE SYSTEM

The Console-based Real Estate Management System embodies a sophisticated yet user-accessible solution meticulously crafted to revolutionize the landscape of real estate management. Crafted by the collaborative efforts of Alulod, Denmark Warrene P., and Perillo, Alvin F., this innovative system represents the pinnacle of their academic pursuit, aligning with the rigorous standards of the Bachelor of Science in Electronics Engineering at the esteemed Marikina Polytechnic College.

System Architecture and Core Functionalities

At its core, this system is a testament to technological advancement, employing a command-line interface to orchestrate the intricate symphony of real estate operations. Its architecture is a fusion of intuitive design and robust functionalities catering to two principal user roles: the discerning administrator and the keen user.

Key Functionalities:

1. Admin Account:

- **Mastering System Dynamics:** Through command-line inputs, administrators wield control over the system's configurations and functionalities, ensuring a well-oiled machinery that operates seamlessly.
- **Guardians of Property Realms:** The administrator's dominion extends to the realm of property listings, with granular control over the intricate details that make each listing unique.

2. User Account:

- **Gateway to Property Exploration:** Users are welcomed into a realm of property exploration, where a treasure trove of listings awaits their perusal, each a potential dream realized.

User-Centric Ingenuity and Technological Marvel

This system transcends the conventional paradigms of real estate management. It's an eloquent fusion of technological marvel and user-centric ingenuity. The Console-based interface, while elegant in its simplicity, belies the sophistication underlying its functionality.



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



Synthesizing Real Estate Operations

This system is more than a mere amalgamation of code; it's an orchestration of functionalities designed to synthesize the myriad complexities of real estate operations. It harmonizes the multifaceted facets of property management into a symphony conducted by the user's keystrokes.

Bridge between Academia and Innovation

As a final output in the esteemed Electronics Engineering curriculum, this project serves as a bridge between academia and innovation. It's the embodiment of theoretical knowledge culminating in a practical manifestation, a testament to the prowess and creativity nurtured within the hallowed halls of Marikina Polytechnic College.

Conclusion: A Revolution Unfolds

In summation, the Console-based Real Estate Management System transcends the realm of a mere academic requirement. It's a testament to innovation, efficiency, and sophistication—a technological marvel poised to revolutionize the way real estate is managed. With an intuitive interface, powerful functionalities, and the promise of enhanced user experiences, this system heralds a new era in the domain of real estate management.



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



II. SIGNIFICANCE OF THE SYSTEM

The Console-based Real Estate Management System represents a paradigm shift in the realm of real estate operations, transcending mere functionality to embody a transformative force within the industry. Its significance resonates across multiple dimensions, each contributing to a tapestry of innovation, efficiency, and user-centricity.

Revolutionizing Operational Efficiency

At its core, this system serves as the vanguard of operational efficiency within the real estate domain. By amalgamating sophisticated command-line functionalities with a streamlined user interface, it introduces a frictionless workflow for administrators and users alike.

The Administrative Interface stands as a citadel of control, allowing administrators to orchestrate system configurations seamlessly. From managing property listings to minutely adjusting property details, this interface empowers administrators to navigate the complex labyrinth of real estate management effortlessly.

Empowering User Experience

For users, the system presents an oasis of possibilities, encapsulated within the User Interface. Here, the art of property exploration takes center stage. Users traverse a digital landscape teeming with available properties, each offering a glimpse into their potential dream homes or investments.

This interface isn't merely a portal but a gateway to personalized experiences. The filtration mechanism ensures users peruse available properties tailored to their preferences, fostering a sense of engagement and ownership in their property search.

Fortifying Administrative Control and Security

Beyond efficiency and user engagement, this system fortifies administrative control and security—a cornerstone in the realm of digital management. The functionalities within the Administrative Interface are a bastion of control, ensuring that only authorized entities navigate the corridors of property management.

The Registration Interface serves as the genesis of administrative oversight, enabling the creation of authorized accounts. It functions not only as an entry point but as a safeguard, ensuring the system's sanctity by authorizing only legitimate access.



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



Catalyst for Technological Adaptability

Moreover, the system's significance extends beyond its present functionalities. Its architecture, meticulously crafted to adapt and evolve, stands as a testament to its readiness for future technological integrations. It serves as a catalyst for innovation, readily embracing emerging technologies to meet the evolving needs of the real estate landscape.

Fusion of Academic Rigor and Practical Innovation

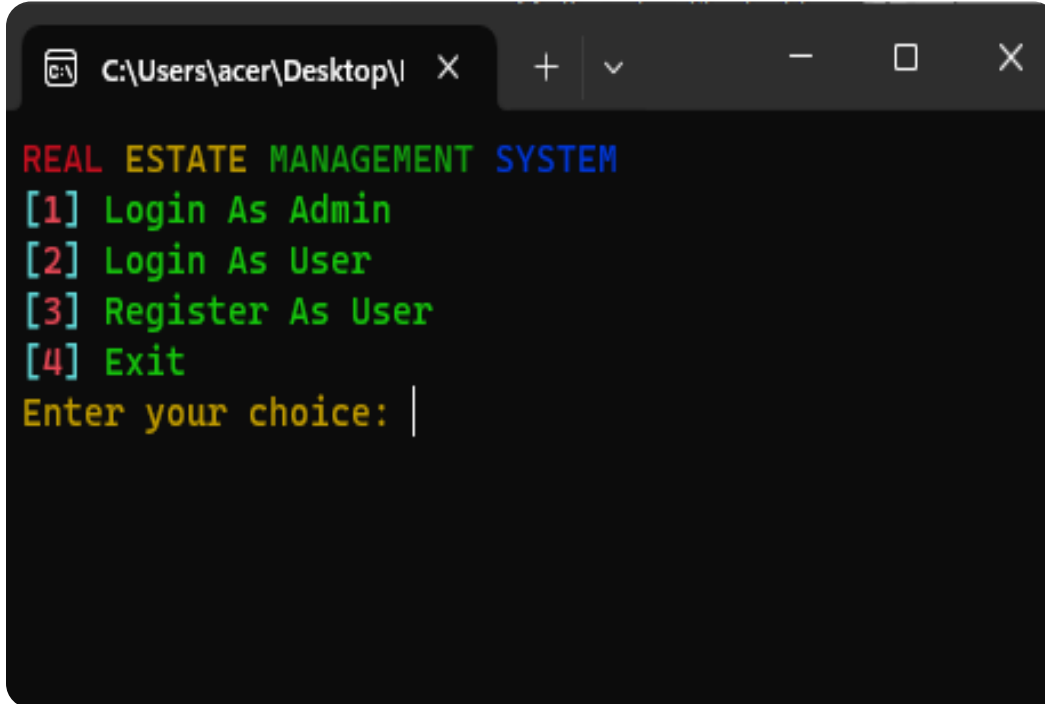
As the crowning achievement of academic pursuit, this system epitomizes the fusion of academic rigor and practical innovation. It symbolizes the culmination of theoretical knowledge, brought to life through intricate code, meticulous design, and an unwavering commitment to excellence.

Conclusion: A New Epoch in Real Estate Management

In conclusion, the Console-based Real Estate Management System is not merely a software solution; it's a harbinger of a new epoch in real estate management. Its significance lies not only in its current functionalities but in its potential to reshape and redefine the very fabric of the industry. Through efficiency enhancements, user empowerment, security fortification, adaptability to emerging technologies, and its role as an academic testament, this system emerges as a catalyst driving the industry towards a future replete with innovation and efficiency.

III. SCREENSHOTS OF ALL THE POSSIBLE PROGRAM OUTPUT

POSSIBLE PROGRAM OUTPUT 1



```
C:\Users\acer\Desktop\ X + v - □ X

REAL ESTATE MANAGEMENT SYSTEM
[1] Login As Admin
[2] Login As User
[3] Register As User
[4] Exit
Enter your choice: |
```

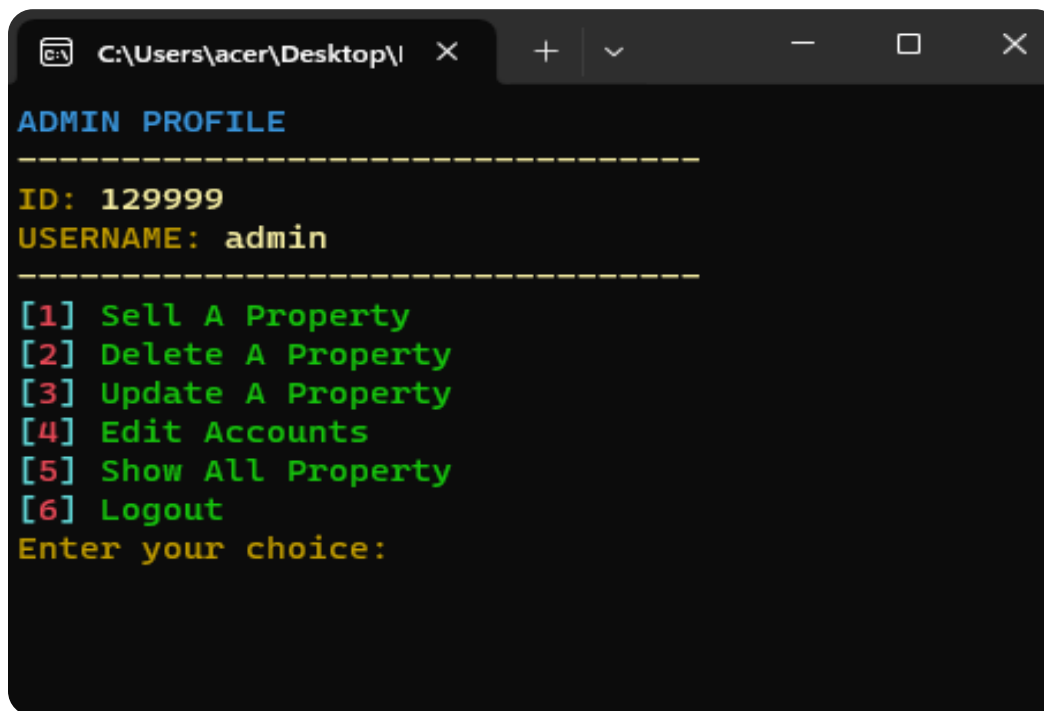
Figure 1: Main Menu of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

The figure above is the main menu of the console-based real estate management system. There are 3 main key functions the admin login interface, the user login interface and registration interface for the user.

POSSIBLE PROGRAM OUTPUT 2



```
C:\Users\acer\Desktop\l x + - □ X

ADMIN PROFILE
-----
ID: 129999
USERNAME: admin
-----
[1] Sell A Property
[2] Delete A Property
[3] Update A Property
[4] Edit Accounts
[5] Show All Property
[6] Logout
Enter your choice:
```

Figure 2: Administrator's Profile of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is the admin interface, it has 5 main key function where it can sell, delete, update, and view all the posted for sale property in the market place. It can view all the property whether it is available or not. Inside the edit accounts, the admin can edit their password, and create another admin account for the agent.

POSSIBLE PROGRAM OUTPUT 3

```
C:\Users\acer\Desktop\My Pri x + - □ x
PROPERTY'S MARKET PLACE
-----
PROPERTY'S DETAILS
Property ID: 4
Property Type: Residential
Property Address: Metro Manila
Property Lot Title: Lost City
No: of Bedrooms: 7
No: of Bathrooms: 3
Property Lot Area: 202 sqm
Price: PHP 1000000.00
Available: Yes
-----

ADMIN PROFILE
-----
ID: 129999
USERNAME: admin
-----

[1] Sell A Property
[2] Delete A Property
[3] Update A Property
[4] Edit Accounts
[5] Show All Property
[6] Logout
Enter your choice: |
```

Figure 3: Administrator's show all property of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is one of the key features of admin's user account, the show all property. It shows all the details of all property without any filtrations.

POSSIBLE PROGRAM OUTPUT 4



```
C:\Users\acer\Desktop\My Pri x + v - □ x

CREATE PROPERTY TO SELL
-----
PROPERTY'S DETAILS
Property ID: 4
Property Type: Residential
Property Address: Metro Manila
Property Lot Title: Lost City
No: of Bedrooms: 7
No: of Bathrooms: 3
Property Lot Area: 202 sqm
Price: PHP 1000000.00
Available: Yes
-----
This property will be created and post accordingly.
Do you want to continue? (Y/N): |
```

Figure 4: Administrator's sell property of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is one of the key features of admin's user account, the sell property. It's a key feature accessible only by administrator's account where they're the only one who can modify or create the property listing in the market place for the users or customers.

POSSIBLE PROGRAM OUTPUT 5

```
C:\Users\acer\Desktop\My Pr  x  +  -  □  x

UPDATE PROPERTY
OLD
-----
PROPERTY'S DETAILS
Property ID: 4
Property Type: Residential
Property Address: Metro Manila
Property Lot Title: Lost City
No: of Bedrooms: 7
No: of Bathrooms: 3
Property Lot Area: 202.00 sqm
Price: PHP 1000000.00
Available: Yes
-----
NEW
-----
PROPERTY'S DETAILS
Property ID: 4
Property Type: Single-Family Home
Property Address: Metro Manila
Property Lot Title: Lost City
No: of Bedrooms: 7
No: of Bathrooms: 3
Property Lot Area: 202.00 sqm
Price: PHP 1000000.00
Available: Yes
-----
Do you want to continue ? (Y/N) : y|
```

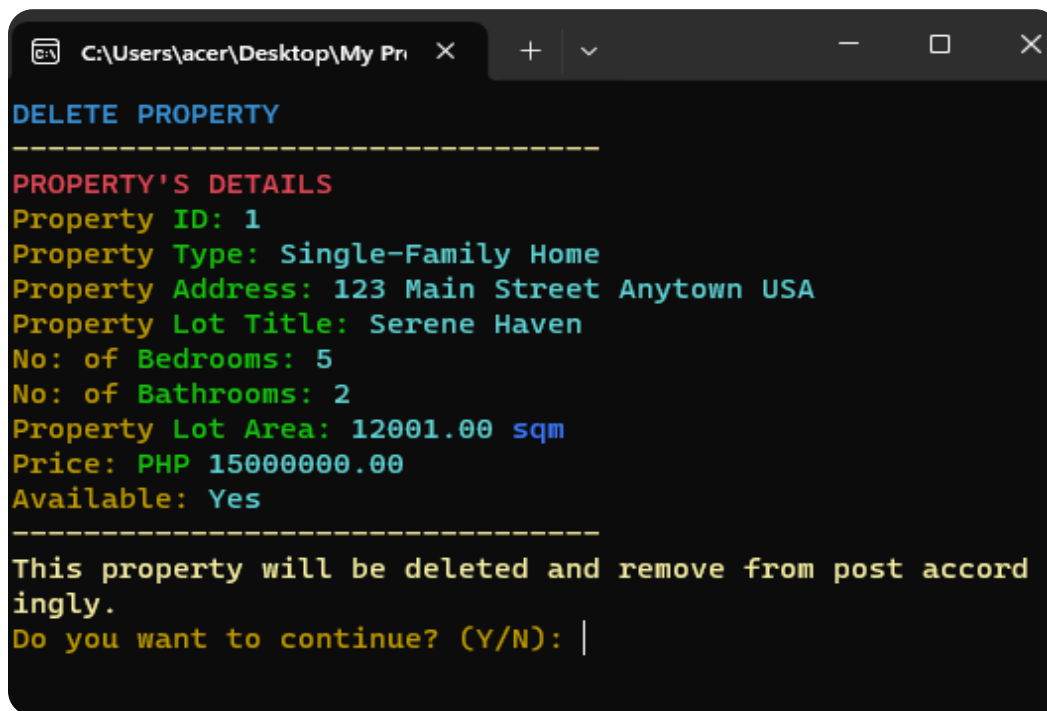
Figure 5: Administrator's update property of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is one of the key features of admin's user account, the update property. This is where the administrator can update a property details for future update, the highlighted red color on the new property details are the update made by the admin. This key feature is also exclusively accessible by the administrator's account.

POSSIBLE PROGRAM OUTPUT 6



```
DELETE PROPERTY
-----
PROPERTY'S DETAILS
Property ID: 1
Property Type: Single-Family Home
Property Address: 123 Main Street Anytown USA
Property Lot Title: Serene Haven
No: of Bedrooms: 5
No: of Bathrooms: 2
Property Lot Area: 12001.00 sqm
Price: PHP 15000000.00
Available: Yes
-----
This property will be deleted and remove from post accord
ingly.
Do you want to continue? (Y/N): |
```

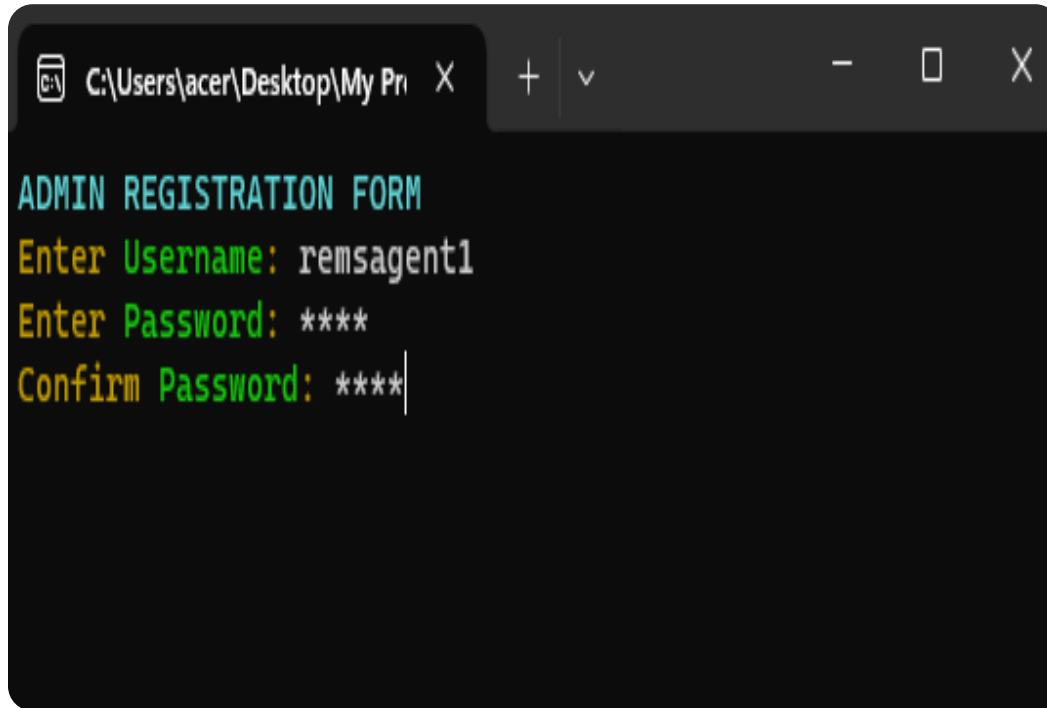
Figure 6: Administrator's delete property of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is one of the key features of admin's user account, the delete property. This is where the administrator can delete a property. This key feature is also exclusively accessible by the administrator's account.

POSSIBLE PROGRAM OUTPUT 7



```
C:\Users\acer\Desktop\My Pri X + v - □ X

ADMIN REGISTRATION FORM
Enter Username: remsagent1
Enter Password: ****
Confirm Password: ****|
```

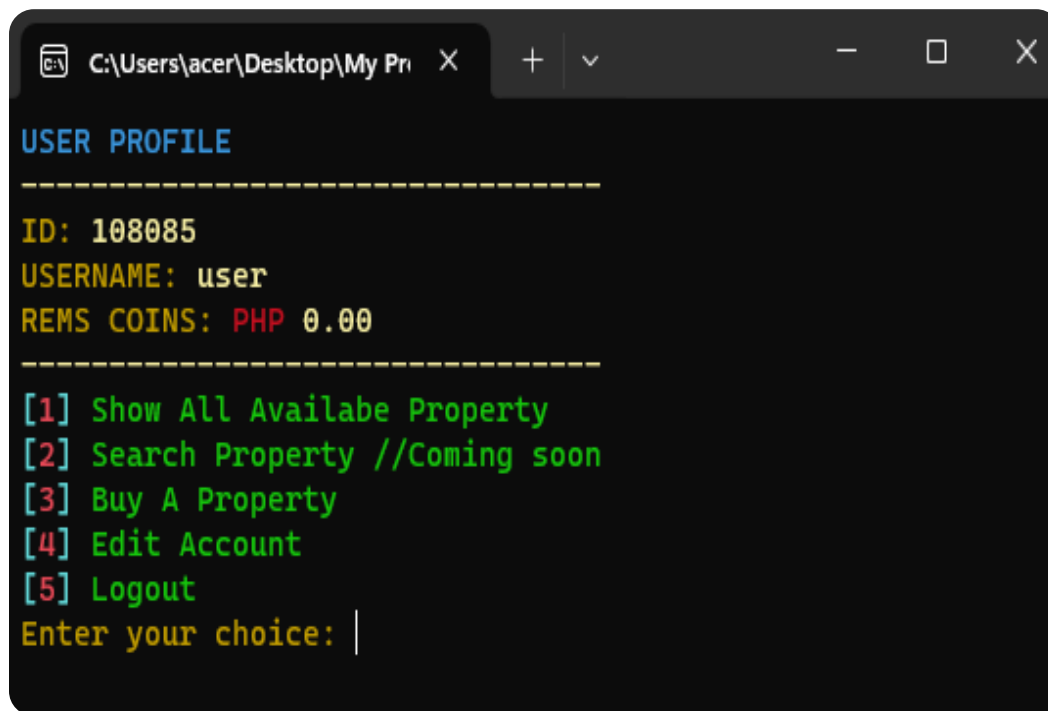
Figure 7: Administrator's registration form of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is one of the key features of admin's user account, the registration for admin account. This is where the administrator can create an account. This key feature is also exclusively accessible by the administrator's account. When there is no account existed, the program will automatically create a default admin account for the admin for first access of the admin user interface and they can change the admin's password if needed.

POSSIBLE PROGRAM OUTPUT 8



```
C:\Users\acer\Desktop\My Pr X + v - □ X

USER PROFILE
-----
ID: 108085
USERNAME: user
REMS COINS: PHP 0.00
-----
[1] Show All Availabe Property
[2] Search Property //Coming soon
[3] Buy A Property
[4] Edit Account
[5] Logout
Enter your choice: |
```

Figure 8: User's interface of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is the user's interface 4 main key features, the show all available property, search property, buy property and edit account. User's accounts are limited only to buy property and top up for their REMS coins including editing their accounts such as changing the password.

POSSIBLE PROGRAM OUTPUT 9

```
C:\Users\acer\Desktop\My Pri x + - _ □ ×

PROPERTY'S MARKET PLACE
-----
PROPERTY'S DETAILS
Property ID: 4
Property Type: Single-Family Home
Property Address: Metro Manila
Property Lot Title: Lost City
No: of Bedrooms: 7
No: of Bathrooms: 3
Property Lot Area: 202.00 sqm
Price: PHP 1000000.00
Available: Yes
-----
USER PROFILE
-----
ID: 108085
USERNAME: user
REMS COINS: PHP 0.00
-----
[1] Show All Availabe Property
[2] Search Property //Coming soon
[3] Buy A Property
[4] Edit Account
[5] Logout
Enter your choice: |
```

Figure 9: User's show all available property of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is the user's one of the key features, the show all available property. The user's accounts are only limited to see all of the available property. On the user side there's a filtration of availability of property where if the property is available then it will only show the available property and hide the not available property.

POSSIBLE PROGRAM OUTPUT 10

```
C:\Users\acer\Desktop\My Pri x + - □ ×
BUY A PROPERTY
USER PROFILE
-----
ID: 108085
USERNAME: user
REMS COINS: PHP 0.00
-----
PROPERTY'S DETAILS
Property ID: 4
Property Type: Single-Family Home
Property Address: Metro Manila
Property Lot Title: Lost City
No: of Bedrooms: 7
No: of Bathrooms: 3
Property Lot Area: 202.00 sqm
Price: PHP 1000000.00
Available: Yes
-----
You will buy this property.
Do you want to continue? (Y/N) : |
```

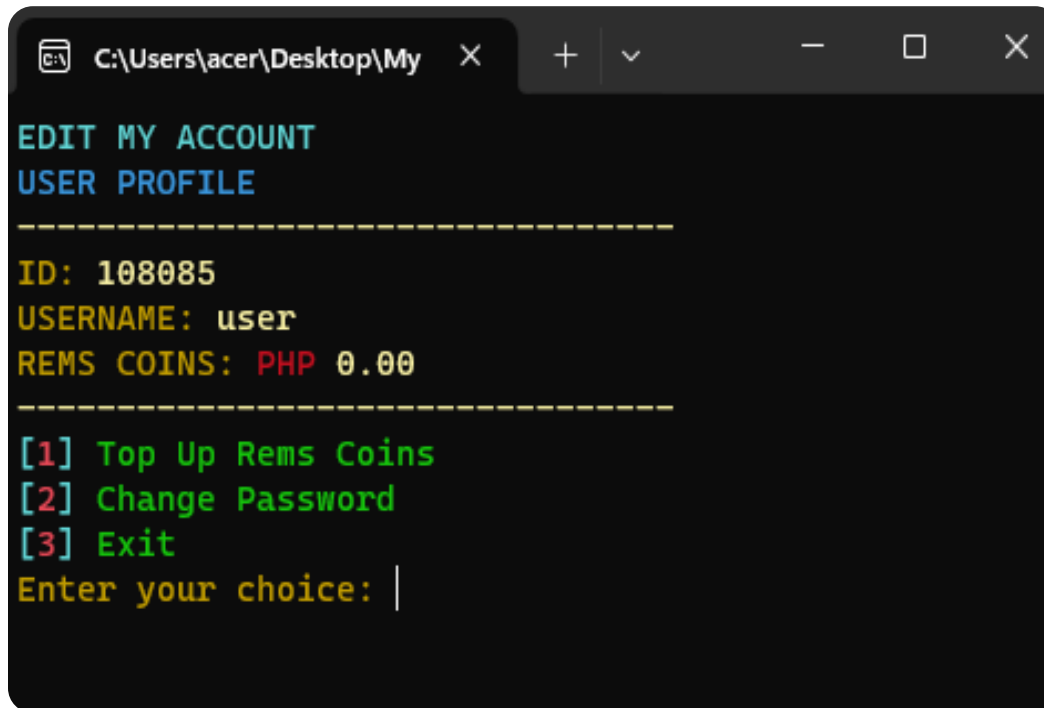
Figure 10: User's buy property of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is the user's one of the key features, the buy property. This is where you can buy a property on the available property list in the market. If the user's REMS coins are not enough for the property price then it will prompt insufficient balance otherwise the transaction will succeed.

POSSIBLE PROGRAM OUTPUT 11



```
C:\Users\acer\Desktop\My X + v - □ X

EDIT MY ACCOUNT
USER PROFILE
-----
ID: 108085
USERNAME: user
REMS COINS: PHP 0.00
-----
[1] Top Up Rems Coins
[2] Change Password
[3] Exit
Enter your choice: |
```

Figure 11: User's edit account of the Console-Based Real Estate Management System

DISCUSSION:

Discuss the figure above not less than 150 words. Just copy this table for the next figure to discuss and name it as Figure 2 Up to N. Always remember that one page only per program output with discussion.

This is the user's one of the key features, the edit account. This is where you can top up a REMS coins, and change the user's account password.



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



IV. CODE OF THE SYSTEM

// COMPLETE CODE C++

Also, you can download the file @:

<https://github.com/demongamer1998/Real-Estate-Management-System>

```
#include <iostream>
#include <windows.h>
#include <limits>
#include <conio.h> // For _getch() and _putch() functions (Windows
specific)
#include <istream>
#include <fstream>
#include <string>
#include <vector>
#include <sstream> // Include this for stringstream
#include <algorithm>
#include <cstdlib> // For rand() and srand()
#include <ctime> // For time()
#include <unistd.h>
#include <iomanip>
void userLogin();
void adminLogin();
int main();

using namespace std;

int stringToInt(const string& str) {
    int result = 0;
    stringstream ss(str);
    ss >> result;
    return result;
}

string maskedInput() {
    string pass;
    char ch;
    while ((ch = _getch()) != '\r') { // \r is the Enter key
        if (ch == '\b') { // \b is the Backspace key
            if (pass.length() > 0) {
                pass.erase(pass.length() - 1); // Erase the last character
                cout << "\b\b"; // Erase the last character visually
            }
        } else {
            pass.push_back(ch);
            cout << '*'; // Print '*' instead of the actual character
        }
    }
    cout << endl;
    return pass;
}

class Color {
public:
    enum ConsoleColor {
        Black = 0,
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Blue = FOREGROUND_BLUE,
Green = FOREGROUND_GREEN,
Cyan = FOREGROUND_GREEN | FOREGROUND_BLUE,
Red = FOREGROUND_RED,
Magenta = FOREGROUND_RED | FOREGROUND_BLUE,
Yellow = FOREGROUND_RED | FOREGROUND_GREEN,
White = FOREGROUND_RED | FOREGROUND_GREEN |
FOREGROUND_BLUE,
Gray = FOREGROUND_INTENSITY,
BrightBlue = FOREGROUND_BLUE | FOREGROUND_INTENSITY,
BrightGreen = FOREGROUND_GREEN | FOREGROUND_INTENSITY,
BrightCyan = FOREGROUND_GREEN | FOREGROUND_BLUE |
FOREGROUND_INTENSITY,
BrightRed = FOREGROUND_RED | FOREGROUND_INTENSITY,
BrightMagenta = FOREGROUND_RED | FOREGROUND_BLUE |
FOREGROUND_INTENSITY,
BrightYellow = FOREGROUND_RED | FOREGROUND_GREEN |
FOREGROUND_INTENSITY,
BrightWhite = FOREGROUND_RED | FOREGROUND_GREEN |
FOREGROUND_BLUE | FOREGROUND_INTENSITY
};

static void setTextColor(ConsoleColor color) {
    HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hConsole, static_cast<WORD>(color));
}
};

void loginCountdown(int seconds) {
    time_t start_time = time(NULL);

    while (difftime(time(NULL), start_time) < seconds) {
        Color::setTextColor(Color::White);
        cout << "Redirecting to your account in ";
        Color::setTextColor(Color::BrightGreen);
        cout << fixed << setprecision(0) << seconds -
difftime(time(NULL), start_time) ;
        Color::setTextColor(Color::White);
        cout << " seconds" << endl;

        sleep(1); // Waits for 1 second
        system("cls");
    }
}

void logoutCountdown(int seconds) {
    time_t start_time = time(NULL);

    while (difftime(time(NULL), start_time) < seconds) {
        Color::setTextColor(Color::White);
        cout << "Redirecting to main in ";
        Color::setTextColor(Color::BrightGreen);
        cout << fixed << setprecision(0) << seconds -
difftime(time(NULL), start_time) ;
        Color::setTextColor(Color::White);
        cout << " seconds" << endl;

        sleep(1); // Waits for 1 second
        system("cls");
    }
}
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
void countdown(int seconds) {
    time_t start_time = time(NULL);
    while (difftime(time(NULL), start_time) < seconds) {
        sleep(1); // Waits for 1 second
        system("cls");
    }
}

class RealEstateProperty {
private:
    string propertyId;
    string propertyType;
    string propertyAddress;
    string propertyLotTitle;
    int bedrooms;
    int bathrooms;
    double lotArea;
    double price;
    bool is_available;

public:
    // Constructor
    RealEstateProperty(string type, string addr, string lotTitle, int beds, int
baths, double lotarea, double pr) :
        propertyType(type), propertyAddress(addr), propertyLotTitle(lotTitle),
        bedrooms(beds), bathrooms(baths), lotArea(lotarea), price(pr),
        is_available(true) {}

    void showAllAvailableProperty() {
        ifstream input("globalProperty.txt");
        string line;
        string propertyId, propertyType, propertyAddress,
propertyLotTitle;
        int bedrooms, bathrooms;
        double price, lotArea;
        bool is_available = true;
        bool isAvailable;
        int count;

        if (!input.is_open() || input.peek() ==
ifstream::traits_type::eof()) {
            Color::setTextColor(Color::BrightRed);
            cout << "\n NO RECORDS AVAILABLE YET" << endl <<
endl;

            Color::setTextColor(Color::BrightYellow);
            cout << "-----" << endl;
            return;
        }

        while (getline(input, line)) {
            istringstream ss(line);
            getline(ss, propertyId, ',');
            getline(ss, propertyType, ',');
            getline(ss, propertyAddress, ',');
            getline(ss, propertyLotTitle, ',');
            ss >> bedrooms;
            ss.ignore(); // Ignore the comma
            ss >> bathrooms;
            ss.ignore(); // Ignore the comma
            ss >> lotArea;
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
ss.ignore(); // Ignore the comma
ss >> price;
ss.ignore(); // Ignore the comma
ss >> isAvailable;

if(is_available == isAvailable){
count +=1;
// Display property details here as needed
Color::setTextColor(Color::BrightRed);
cout << "PROPERTY'S DETAILS"<<endl;
Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "ID: " ;
    Color::setTextColor(Color::BrightCyan);
    cout << propertyId << endl;
    Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Type: " ;
    Color::setTextColor(Color::BrightCyan);
    cout << propertyType << endl;
    Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Address: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< propertyAddress << endl;
    Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Lot Title: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< propertyLotTitle << endl;
    Color::setTextColor(Color::Yellow);
cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
    cout << "Bedrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bedrooms << endl;
    Color::setTextColor(Color::Yellow);
cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
    cout << "Bathrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bathrooms << endl;
    Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Lot Area: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< lotArea ;
    Color::setTextColor(Color::BrightBlue);
    cout << " sqm" << endl;
    Color::setTextColor(Color::Yellow);
cout <<fixed<<setprecision(2)<< "Price: ";
Color::setTextColor(Color::BrightGreen);
    cout << "PHP " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< price << endl;
```



```
        Color::setTextColor(Color::Yellow);
        cout << "Available: ";
        Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
        cout<< (isAvailable ? "Yes" : "No") <<
endl;

        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
    }

    }
    input.close();

    if(count==0){
        Color::setTextColor(Color::BrightRed);
        cout << "\n NO RECORDS AVAILABLE YET" <<
endl << endl;

        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
    }
}

bool isPropertyAvailable(string Id) const {
    ifstream input("globalProperty.txt");
    string  propertyId,  propertyType,  propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price,lotArea;
    bool isAvailable;

    if (!input.is_open()) {
        // No database
    }

    while (getline(input, line)) {
        istringstream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
        ss >> bedrooms;
        ss.ignore(); // Ignore the comma
        ss >> bathrooms;
        ss.ignore(); // Ignore the comma
        ss >> lotArea;
        ss.ignore(); // Ignore the comma
        ss >> price;
        ss.ignore(); // Ignore the comma
        ss >> isAvailable;

        if(Id == propertyId){
            return isAvailable;
            break;
        }
    }

    input.close();
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
}

void showUpdatePropertyByLotTitle(string id,string lottitle) {
    ifstream input("globalProperty.txt");
    string propertyId, propertyType, propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    if (!input.is_open() || input.peek() ==
ifstream::traits_type::eof()) {
        Color::setTextColor(Color::BrightRed);
        cout << "\n NO RECORDS AVAILABLE YET" << endl <<
endl;

        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
        return;
    }

    while (getline(input, line)) {
        istringstream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
        ss >> bedrooms;
        ss.ignore(); // Ignore the comma
        ss >> bathrooms;
        ss.ignore(); // Ignore the comma
        ss >> lotArea;
        ss.ignore(); // Ignore the comma
        ss >> price;
        ss.ignore(); // Ignore the comma
        ss >> isAvailable;

        if(id == propertyId){

            // Display property details here as needed
            Color::setTextColor(Color::BrightRed);
            cout << "PROPERTY'S DETAILS"<<endl;
            Color::setTextColor(Color::Yellow);
            cout << "Property ";
            Color::setTextColor(Color::BrightGreen);
            cout << "ID: " ;
            Color::setTextColor(Color::BrightCyan);
            cout << propertyId << endl;
            Color::setTextColor(Color::Yellow);

            cout << "Property ";
            Color::setTextColor(Color::BrightGreen);
            cout << "Type: " ;
            Color::setTextColor(Color::BrightCyan);
            cout << propertyType << endl;
            Color::setTextColor(Color::Yellow);

            cout << "Property ";
            Color::setTextColor(Color::BrightGreen);
            cout << "Address: " ;
            Color::setTextColor(Color::BrightCyan);
            cout<< propertyAddress << endl;
```



```
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Lot Title: " ;
        Color::setTextColor(Color::BrightRed);
        cout<< lottitle << endl;
        Color::setTextColor(Color::Yellow);
    cout << "No: of ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Bedrooms: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< bedrooms << endl;
        Color::setTextColor(Color::Yellow);
    cout << "No: of ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Bathrooms: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< bathrooms << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Lot Area: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< lotArea ;
        Color::setTextColor(Color::BrightBlue);
        cout << " sqm" << endl;
        Color::setTextColor(Color::Yellow);
    cout <<fixed<<setprecision(2)<< "Price: ";
    Color::setTextColor(Color::BrightGreen);
        cout << "PHP " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< price << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Available: " ;
    Color::setTextColor(isAvailable ? Color::BrightCyan :
    Color::BrightRed);
        cout<< (isAvailable ? "Yes" : "No") <<
endl;

    Color::setTextColor(Color::BrightYellow);
    cout << "-----" << endl;
    }
}

input.close();
}

void showUpdatePropertyByLotArea(string id,double lotarea)
{
    ifstream input("globalProperty.txt");
    string propertyId, propertyType, propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    if (!input.is_open() || input.peek() ==
ifstream::traits_type::eof()) {
        Color::setTextColor(Color::BrightRed);
```



```
endl;

cout << "\n  NO RECORDS AVAILABLE YET" << endl <<

Color::setTextColor(Color::BrightYellow);
cout << "-----" << endl;
return;
}

while (getline(input, line)) {
    istringstream ss(line);
    getline(ss, propertyId, ',');
    getline(ss, propertyType, ',');
    getline(ss, propertyAddress, ',');
    getline(ss, propertyLotTitle, ',');
    ss >> bedrooms;
    ss.ignore(); // Ignore the comma
    ss >> bathrooms;
    ss.ignore(); // Ignore the comma
    ss >> lotArea;
    ss.ignore(); // Ignore the comma
    ss >> price;
    ss.ignore(); // Ignore the comma
    ss >> isAvailable;

    if(id == propertyId){

        // Display property details here as needed
        Color::setTextColor(Color::BrightRed);
        cout << "PROPERTY'S DETAILS"<<endl;
        Color::setTextColor(Color::Yellow);
        cout << "Property ";
        Color::setTextColor(Color::BrightGreen);
            cout << "ID: " ;
            Color::setTextColor(Color::BrightCyan);
            cout << propertyId << endl;
            Color::setTextColor(Color::Yellow);

        cout << "Property ";
        Color::setTextColor(Color::BrightGreen);
            cout << "Type: " ;
            Color::setTextColor(Color::BrightCyan);
            cout << propertyType << endl;
            Color::setTextColor(Color::Yellow);

        cout << "Property ";
        Color::setTextColor(Color::BrightGreen);
            cout << "Address: " ;
            Color::setTextColor(Color::BrightCyan);
            cout<< propertyAddress << endl;
            Color::setTextColor(Color::Yellow);

        cout << "Property ";
        Color::setTextColor(Color::BrightGreen);
            cout << "Lot Title: " ;
            Color::setTextColor(Color::BrightCyan);
            cout<< propertyLotTitle << endl;
            Color::setTextColor(Color::Yellow);

        cout << "No: of ";
        Color::setTextColor(Color::BrightGreen);
            cout << "Bedrooms: " ;
            Color::setTextColor(Color::BrightCyan);
            cout<< bedrooms << endl;
            Color::setTextColor(Color::Yellow);

        cout << "No: of ";
```




```
Color::setTextColor(Color::BrightGreen);
    cout << "Bathrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bathrooms << endl;
    Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Lot Area: " ;
    Color::setTextColor(Color::BrightRed);
    cout<< lotarea ;
    Color::setTextColor(Color::BrightBlue);
    cout << " sqm" << endl;
    Color::setTextColor(Color::Yellow);
    cout <<fixed<<setprecision(2)<< "Price: ";
    Color::setTextColor(Color::BrightGreen);
    cout << "PHP " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< price << endl;
    Color::setTextColor(Color::Yellow);
    cout << "Available: " ;
    Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
    cout<< (isAvailable ? "Yes" : "No") <<
endl;

    Color::setTextColor(Color::BrightYellow);
    cout << "-----" << endl;
    }
}

input.close();
}

void showUpdatePropertyByPrice(string id,double prices) {
    ifstream input("globalProperty.txt");
    string propertyId, propertyType, propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    if (!input.is_open()) || input.peek() ==
ifstream::traits_type::eof()) {
        Color::setTextColor(Color::BrightRed);
        cout << "\n NO RECORDS AVAILABLE YET" << endl <<
endl;

        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
        return;
    }

    while (getline(input, line)) {
        istream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
        ss >> bedrooms;
        ss.ignore(); // Ignore the comma
```



```
ss >> bathrooms;
ss.ignore(); // Ignore the comma
ss >> lotArea;
ss.ignore(); // Ignore the comma
ss >> price;
ss.ignore(); // Ignore the comma
ss >> isAvailable;

if(id == propertyId){

    // Display property details here as needed
    Color::setTextColor(Color::BrightRed);
    cout << "PROPERTY'S DETAILS"<<endl;
    Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "ID: " ;
        Color::setTextColor(Color::BrightCyan);
        cout << propertyId << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Type: " ;
        Color::setTextColor(Color::BrightCyan);
        cout << propertyType << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Address: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< propertyAddress << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Lot Title: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< propertyLotTitle << endl;
        Color::setTextColor(Color::Yellow);
    cout << "No: of ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Bedrooms: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< bedrooms << endl;
        Color::setTextColor(Color::Yellow);
    cout << "No: of ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Bathrooms: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< bathrooms << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Lot Area: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< lotArea ;
        Color::setTextColor(Color::BrightBlue);
        cout << " sqm" << endl;
        Color::setTextColor(Color::Yellow);
    cout <<fixed<<setprecision(2)<< "Price: ";
    Color::setTextColor(Color::BrightGreen);
```



```
        cout << "PHP " ;
        Color::setTextColor(Color::BrightRed);
        cout<< prices << endl;
        Color::setTextColor(Color::Yellow);
        cout << "Available: " ;
        Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
        cout<< (isAvailable ? "Yes" : "No") <<
endl;

        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
    }
}

input.close();
}

void showUpdatePropertyByBathrooms(string id,int
bathroom) {
    ifstream input("globalProperty.txt");
    string propertyId, propertyType, propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    if (!input.is_open() || input.peek() ==
ifstream::traits_type::eof()) {
        Color::setTextColor(Color::BrightRed);
        cout << "\n NO RECORDS AVAILABLE YET" << endl <<
endl;

        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
        return;
    }

    while (getline(input, line)) {
        istringstream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
        ss >> bedrooms;
        ss.ignore(); // Ignore the comma
        ss >> bathrooms;
        ss.ignore(); // Ignore the comma
        ss >> lotArea;
        ss.ignore(); // Ignore the comma
        ss >> price;
        ss.ignore(); // Ignore the comma
        ss >> isAvailable;

        if(id == propertyId){

            // Display property details here as needed
            Color::setTextColor(Color::BrightRed);
            cout << "PROPERTY'S DETAILS"<<endl;
            Color::setTextColor(Color::Yellow);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
cout << "ID: " ;
Color::setTextColor(Color::BrightCyan);
cout << propertyId << endl;
Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
cout << "Type: " ;
Color::setTextColor(Color::BrightCyan);
cout << propertyType << endl;
Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
cout << "Address: " ;
Color::setTextColor(Color::BrightCyan);
cout << propertyAddress << endl;
Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
cout << "Lot Title: " ;
Color::setTextColor(Color::BrightCyan);
cout << propertyLotTitle << endl;
Color::setTextColor(Color::Yellow);
cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
cout << "Bedrooms: " ;
Color::setTextColor(Color::BrightCyan);
cout << bedrooms << endl;
Color::setTextColor(Color::Yellow);
cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
cout << "Bathrooms: " ;
Color::setTextColor(Color::BrightRed);
cout << bathroom << endl;
Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
cout << "Lot Area: " ;
Color::setTextColor(Color::BrightCyan);
cout << lotArea ;
Color::setTextColor(Color::BrightBlue);
cout << " sqm" << endl;
Color::setTextColor(Color::Yellow);
cout << fixed << setprecision(2) << "Price: ";
Color::setTextColor(Color::BrightGreen);
cout << "PHP " ;
Color::setTextColor(Color::BrightCyan);
cout << price << endl;
Color::setTextColor(Color::Yellow);
cout << "Available: " ;
Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
cout << (isAvailable ? "Yes" : "No") <<
endl;

Color::setTextColor(Color::BrightYellow);
cout << "-----" << endl;
}
```



```
        input.close();
    }

    void showUpdatePropertyByBedrooms(string id,int bedroom)
    {
        ifstream input("globalProperty.txt");
        string  propertyId,  propertyType,  propertyAddress,
propertyLotTitle;
        string line;
        int bedrooms, bathrooms;
        double price, lotArea;
        bool isAvailable;

        if      (!input.is_open()) || input.peek()      ==
ifstream::traits_type::eof() {
            Color::setTextColor(Color::BrightRed);
            cout << "\n    NO RECORDS AVAILABLE YET" << endl <<
endl;
            Color::setTextColor(Color::BrightYellow);
            cout << "-----" << endl;
            return;
        }

        while (getline(input, line)) {
            istringstream ss(line);
            getline(ss, propertyId, ',');
            getline(ss, propertyType, ',');
            getline(ss, propertyAddress, ',');
            getline(ss, propertyLotTitle, ',');
            ss >> bedrooms;
            ss.ignore(); // Ignore the comma
            ss >> bathrooms;
            ss.ignore(); // Ignore the comma
            ss >> lotArea;
            ss.ignore(); // Ignore the comma
            ss >> price;
            ss.ignore(); // Ignore the comma
            ss >> isAvailable;

            if(id == propertyId){

                // Display property details here as needed
                Color::setTextColor(Color::BrightRed);
                cout << "PROPERTY'S DETAILS"<<endl;
                Color::setTextColor(Color::Yellow);
                cout << "Property ";
                Color::setTextColor(Color::BrightGreen);
                cout << "ID: " ;
                Color::setTextColor(Color::BrightCyan);
                cout << propertyId << endl;
                Color::setTextColor(Color::Yellow);

                cout << "Property ";
                Color::setTextColor(Color::BrightGreen);
                cout << "Type: " ;
                Color::setTextColor(Color::BrightCyan);
                cout << propertyType << endl;
                Color::setTextColor(Color::Yellow);

                cout << "Property ";
                Color::setTextColor(Color::BrightGreen);
```



```
        cout << "Address: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< propertyAddress << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Lot Title: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< propertyLotTitle << endl;
        Color::setTextColor(Color::Yellow);
    cout << "No: of ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Bedrooms: " ;
        Color::setTextColor(Color::BrightRed);
        cout<< bedroom << endl;
        Color::setTextColor(Color::Yellow);
    cout << "No: of ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Bathrooms: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< bathrooms << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Lot Area: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< lotArea ;
        Color::setTextColor(Color::BrightBlue);
        cout << " sqm" << endl;
        Color::setTextColor(Color::Yellow);
    cout <<fixed<<setprecision(2)<< "Price: ";
    Color::setTextColor(Color::BrightGreen);
        cout << "PHP " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< price << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Available: " ;
    Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
        cout<< (isAvailable ? "Yes" : "No") <<
endl;

    Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
    }
}

    input.close();
}

void showUpdatePropertyByAddress(string id,string address) {
    ifstream input("globalProperty.txt");
    string propertyId, propertyType, propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;
```



```
        if (!input.is_open() || input.peek() ==  
ifstream::traits_type::eof()) {  
            Color::setTextColor(Color::BrightRed);  
            cout << "\n    NO RECORDS AVAILABLE YET" << endl <<  
endl;  
            Color::setTextColor(Color::BrightYellow);  
            cout << "-----" << endl;  
            return;  
        }  
  
        while (getline(input, line)) {  
            istringstream ss(line);  
            getline(ss, propertyId, ',');  
            getline(ss, propertyType, ',');  
            getline(ss, propertyAddress, ',');  
            getline(ss, propertyLotTitle, ',');  
            ss >> bedrooms;  
            ss.ignore(); // Ignore the comma  
            ss >> bathrooms;  
            ss.ignore(); // Ignore the comma  
            ss >> lotArea;  
            ss.ignore(); // Ignore the comma  
            ss >> price;  
            ss.ignore(); // Ignore the comma  
            ss >> isAvailable;  
  
            if (id == propertyId){  
  
                // Display property details here as needed  
                Color::setTextColor(Color::BrightRed);  
                cout << "PROPERTY'S DETAILS" << endl;  
                Color::setTextColor(Color::Yellow);  
                cout << "Property ";  
                Color::setTextColor(Color::BrightGreen);  
                cout << "ID: " ;  
                Color::setTextColor(Color::BrightCyan);  
                cout << propertyId << endl;  
                Color::setTextColor(Color::Yellow);  
                cout << "Property ";  
                Color::setTextColor(Color::BrightGreen);  
                cout << "Type: " ;  
                Color::setTextColor(Color::BrightCyan);  
                cout << propertyType << endl;  
                Color::setTextColor(Color::Yellow);  
                cout << "Property ";  
                Color::setTextColor(Color::BrightGreen);  
                cout << "Address: " ;  
                Color::setTextColor(Color::BrightRed);  
                cout << address << endl;  
                Color::setTextColor(Color::Yellow);  
                cout << "Property ";  
                Color::setTextColor(Color::BrightGreen);  
                cout << "Lot Title: " ;  
                Color::setTextColor(Color::BrightCyan);  
                cout << propertyLotTitle << endl;  
                Color::setTextColor(Color::Yellow);  
                cout << "No. of ";  
                Color::setTextColor(Color::BrightGreen);  
                cout << "Bedrooms: " ;  
                Color::setTextColor(Color::BrightCyan);
```



```
        cout<< bedrooms << endl;
        Color::setTextColor(Color::Yellow);
    cout << "No: of ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Bathrooms: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< bathrooms << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
        cout << "Lot Area: " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< lotArea ;
        Color::setTextColor(Color::BrightBlue);
        cout << " sqm" << endl;
        Color::setTextColor(Color::Yellow);
    cout <<fixed<<setprecision(2)<< "Price: ";
    Color::setTextColor(Color::BrightGreen);
        cout << "PHP " ;
        Color::setTextColor(Color::BrightCyan);
        cout<< price << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Available: " ;
    Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
        cout<< (isAvailable ? "Yes" : "No") <<
endl;

    Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
    }
}

input.close();
}

void showUpdatePropertyByType(string id,string type) {
    ifstream input("globalProperty.txt");
    string  propertyId,  propertyType,  propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    if      (!input.is_open())||input.peek()      ==
ifstream::traits_type::eof()) {
        Color::setTextColor(Color::BrightRed);
        cout << "\n  NO RECORDS AVAILABLE YET" << endl <<
endl;
        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
        return;
    }

    while (getline(input, line)) {
        istream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
```




```
getline(ss, propertyLotTitle, ',');
ss >> bedrooms;
ss.ignore(); // Ignore the comma
ss >> bathrooms;
ss.ignore(); // Ignore the comma
ss >> lotArea;
ss.ignore(); // Ignore the comma
ss >> price;
ss.ignore(); // Ignore the comma
ss >> isAvailable;

if(id == propertyId){

// Display property details here as needed
Color::setTextColor(Color::BrightRed);
cout << "PROPERTY'S DETAILS"<<endl;
Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "ID: " ;
    Color::setTextColor(Color::BrightCyan);
    cout << propertyId << endl;
    Color::setTextColor(Color::Yellow);

cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Type: " ;
    Color::setTextColor(Color::BrightRed);
    cout << type << endl;
    Color::setTextColor(Color::Yellow);

cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Address: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< propertyAddress << endl;
    Color::setTextColor(Color::Yellow);

cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Lot Title: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< propertyLotTitle << endl;
    Color::setTextColor(Color::Yellow);

cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
    cout << "Bedrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bedrooms << endl;
    Color::setTextColor(Color::Yellow);

cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
    cout << "Bathrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bathrooms << endl;
    Color::setTextColor(Color::Yellow);

cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Lot Area: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< lotArea ;
    Color::setTextColor(Color::BrightBlue);
    cout << " sqm" << endl;
```



```
        Color::setTextColor(Color::Yellow);
    cout << fixed << setprecision(2) << "Price: ";
    Color::setTextColor(Color::BrightGreen);
        cout << "PHP ";
        Color::setTextColor(Color::BrightCyan);
        cout << price << endl;
        Color::setTextColor(Color::Yellow);
    cout << "Available: ";
    Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
        cout << (isAvailable ? "Yes" : "No") <<
endl;

    Color::setTextColor(Color::BrightYellow);
    cout << "-----" << endl;
    }
}

input.close();
}

double getPriceOfProperty(string id) {
    ifstream input("globalProperty.txt");
    string  propertyId,  propertyType,  propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    while (getline(input, line)) {
        istringstream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
        ss >> bedrooms;
        ss.ignore(); // Ignore the comma
        ss >> bathrooms;
        ss.ignore(); // Ignore the comma
        ss >> lotArea;
        ss.ignore(); // Ignore the comma
        ss >> price;
        ss.ignore(); // Ignore the comma
        ss >> isAvailable;

        if(id == propertyId){
            return price;
        }
    }

    input.close();
}

void showPropertyById(string id) {
    ifstream input("globalProperty.txt");
    string  propertyId,  propertyType,  propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
```



```
double price, lotArea;
bool isAvailable;

if (!input.is_open() || input.peek() ==
ifstream::traits_type::eof()) {
    Color::setTextColor(Color::BrightRed);
    cout << "\n NO RECORDS AVAILABLE YET" << endl <<
endl;

    Color::setTextColor(Color::BrightYellow);
    cout << "-----" << endl;
    return;
}

while (getline(input, line)) {
    istringstream ss(line);
    getline(ss, propertyId, ',');
    getline(ss, propertyType, ',');
    getline(ss, propertyAddress, ',');
    getline(ss, propertyLotTitle, ',');
    ss >> bedrooms;
    ss.ignore(); // Ignore the comma
    ss >> bathrooms;
    ss.ignore(); // Ignore the comma
    ss >> lotArea;
    ss.ignore(); // Ignore the comma
    ss >> price;
    ss.ignore(); // Ignore the comma
    ss >> isAvailable;

    if(id == propertyId){

        // Display property details here as needed
        Color::setTextColor(Color::BrightRed);
        cout << "PROPERTY'S DETAILS"<<endl;
        Color::setTextColor(Color::Yellow);
        cout << "Property ";
        Color::setTextColor(Color::BrightGreen);
            cout << "ID: " ;
            Color::setTextColor(Color::BrightCyan);
            cout << propertyId << endl;
            Color::setTextColor(Color::Yellow);

        cout << "Property ";
        Color::setTextColor(Color::BrightGreen);
            cout << "Type: " ;
            Color::setTextColor(Color::BrightCyan);
            cout << propertyType << endl;
            Color::setTextColor(Color::Yellow);

        cout << "Property ";
        Color::setTextColor(Color::BrightGreen);
            cout << "Address: " ;
            Color::setTextColor(Color::BrightCyan);
            cout<< propertyAddress << endl;
            Color::setTextColor(Color::Yellow);

        cout << "Property ";
        Color::setTextColor(Color::BrightGreen);
            cout << "Lot Title: " ;
            Color::setTextColor(Color::BrightCyan);
            cout<< propertyLotTitle << endl;
            Color::setTextColor(Color::Yellow);

        cout << "No: of ";
```



```
Color::setTextColor(Color::BrightGreen);
    cout << "Bedrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bedrooms << endl;
    Color::setTextColor(Color::Yellow);
    cout << "No: of ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Bathrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bathrooms << endl;
    Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Lot Area: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< lotArea ;
    Color::setTextColor(Color::BrightBlue);
    cout << " sqm" << endl;
    Color::setTextColor(Color::Yellow);
    cout <<fixed<<setprecision(2)<< "Price: ";
    Color::setTextColor(Color::BrightGreen);
    cout << "PHP " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< price << endl;
    Color::setTextColor(Color::Yellow);
    cout << "Available: " ;
    Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
    cout<< (isAvailable ? "Yes" : "No") <<
endl;

    Color::setTextColor(Color::BrightYellow);
    cout << "-----" << endl;
    }

    input.close();
}

void readPropertyById(string id) {
    ifstream input("globalProperty.txt");
    string propertyId, propertyType, propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    if (!input.is_open() || input.peek() ==
ifstream::traits_type::eof()) {
        Color::setTextColor(Color::BrightRed);
        cout << "\n NO RECORDS AVAILABLE YET" << endl <<
endl;

        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
        return;
    }

    while (getline(input, line)) {
        istringstream ss(line);
```



```
getline(ss, propertyId, ',');
getline(ss, propertyType, ',');
getline(ss, propertyAddress, ',');
getline(ss, propertyLotTitle, ',');
ss >> bedrooms;
ss.ignore(); // Ignore the comma
ss >> bathrooms;
ss.ignore(); // Ignore the comma
ss >> lotArea;
ss.ignore(); // Ignore the comma
ss >> price;
ss.ignore(); // Ignore the comma
ss >> isAvailable;

if(id == propertyId){
    propertyId = propertyId;
    propertyType = propertyType;
    propertyAddress = propertyAddress;
    propertyLotTitle = propertyLotTitle;
    bedrooms = bedrooms;
    bathrooms = bathrooms;
    lotArea = lotArea;
    price = price;
    isAvailable = isAvailable;
}

}

input.close();
}

void showAllProperty() {
    ifstream input("globalProperty.txt");
    string line;
    string    propertyId,    propertyType,    propertyAddress,
propertyLotTitle;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    if      (!input.is_open()) || input.peek()      ==
ifstream::traits_type::eof() {
        Color::setTextColor(Color::BrightRed);
        cout << "\n    NO RECORDS AVAILABLE YET" << endl <<
endl;
        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
        return;
    }

    while (getline(input, line)) {
        istringstream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
        ss >> bedrooms;
        ss.ignore(); // Ignore the comma
        ss >> bathrooms;
        ss.ignore(); // Ignore the comma
        ss >> lotArea;
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
ss.ignore(); // Ignore the comma
ss >> price;
ss.ignore(); // Ignore the comma
ss >> isAvailable;

// Display property details here as needed
Color::setTextColor(Color::BrightRed);
cout << "PROPERTY'S DETAILS"<<endl;
Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "ID: " ;
    Color::setTextColor(Color::BrightCyan);
    cout << propertyId << endl;
    Color::setTextColor(Color::Yellow);

cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Type: " ;
    Color::setTextColor(Color::BrightCyan);
    cout << propertyType << endl;
    Color::setTextColor(Color::Yellow);

cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Address: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< propertyAddress << endl;
    Color::setTextColor(Color::Yellow);

cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Lot Title: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< propertyLotTitle << endl;
    Color::setTextColor(Color::Yellow);

cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
    cout << "Bedrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bedrooms << endl;
    Color::setTextColor(Color::Yellow);

cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
    cout << "Bathrooms: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< bathrooms << endl;
    Color::setTextColor(Color::Yellow);

cout << "Property ";
Color::setTextColor(Color::BrightGreen);
    cout << "Lot Area: " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< lotArea ;
    Color::setTextColor(Color::BrightBlue);
    cout << " sqm" << endl;
    Color::setTextColor(Color::Yellow);

cout <<fixed<<setprecision(2)<< "Price: ";
Color::setTextColor(Color::BrightGreen);
    cout << "PHP " ;
    Color::setTextColor(Color::BrightCyan);
    cout<< price << endl;
    Color::setTextColor(Color::Yellow);

cout << "Available: " ;
```



```
        Color::setTextColor(isAvailable ? Color::BrightCyan :
Color::BrightRed);
        cout<< (isAvailable ? "Yes" : "No") <<
endl;

        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
    }

    input.close();
}

//CreateProperty Constructor
void createProperty(){
    string propertyType, propertyAddress,
propertyLotTitle;

    int propertyId, bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable = true;
    Color::setTextColor(Color::BrightCyan);
    cout << "CREATE PROPERTY TO SELL" << endl;
    Color::setTextColor(Color::Red);
    cout << "NOTE: ";
    Color::setTextColor(Color::Yellow);
    cout << "NO ";

    Color::setTextColor(Color::BrightGreen);
    cout << "SPACES ";
    Color::setTextColor(Color::Blue);
    cout << "REQUIRED" << endl;
    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Property Id";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);
    bool check = true;
    while (check) {
        if(!cin >> propertyId){
            system("cls");

            Color::setTextColor(Color::Red);
            cout << "Invalid ";

            Color::setTextColor(Color::White);
            cout << "input";

            Color::setTextColor(Color::White);
            cout << ". ";
            cout << "Please
enter a ";

            Color::setTextColor(Color::BrightGreen);
            cout << "valid ";

            Color::setTextColor(Color::Yellow);
            cout <<
"Property ";
```



```
        Color::setTextColor(Color::BrightGreen);
                                cout << "ID." <<
endl;
                                cin.clear();

cin.ignore(numeric_limits<streamsize>::max(), '\n');
                                }
                                else
if(propertyIdExistsInDatabase(propertyId)){
                                system("cls");

        Color::setTextColor(Color::Yellow);
                                cout << "Property ";

        Color::setTextColor(Color::BrightGreen);
                                cout << "ID ";

        Color::setTextColor(Color::White);
                                cout << "is
already ";

        Color::setTextColor(Color::Red);
                                cout << "taken ";

        Color::setTextColor(Color::White);
                                cout << ". ";
                                cout << "Please ";

        Color::setTextColor(Color::White);
                                cout << "enter ";

        Color::setTextColor(Color::White);
                                cout << "a ";

        Color::setTextColor(Color::Yellow);
                                cout << "unique
";

        Color::setTextColor(Color::BrightGreen);
                                cout << "ID." <<
endl;
                                cin.clear();

cin.ignore(numeric_limits<streamsize>::max(), '\n');
                                } else {
                                check = false;
                                break;
                                }

        Color::setTextColor(Color::BrightCyan);
        cout << "CREATE PROPERTY TO SELL" << endl;
        Color::setTextColor(Color::Red);
        cout << "NOTE: ";

        Color::setTextColor(Color::Yellow);
        cout << "NO ";

        Color::setTextColor(Color::BrightGreen);
        cout << "SPACES ";
```




Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::Blue);
                                cout << "REQUIRED" <<
endl;

    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Property Id";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);
}

    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Property Type";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);
    cin.ignore();
    getline(cin,propertyType);

    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Property Address";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);
    getline(cin,propertyAddress);

    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Property Lot Title";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);
    getline(cin,propertyLotTitle);

do {
    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Property Bedrooms";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);

    if (!(cin >> bedrooms)) {
        system("cls");
        cin.clear();

cin.ignore(numeric_limits<streamsize>::max(), '\n');
        Color::setTextColor(Color::Red);
        cout << "Invalid";
        Color::setTextColor(Color::White);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```

                                cout << " input. ";
                                cout << "Please

enter a ";

                                Color::setTextColor(Color::BrightGreen);
                                cout << "valid ";

                                Color::setTextColor(Color::White);
                                cout << "number

of ";

                                Color::setTextColor(Color::Yellow);
                                cout <<

"bedrooms." << endl;

                                Color::setTextColor(Color::BrightCyan);
                                cout << "CREATE PROPERTY TO

SELL" << endl;

                                Color::setTextColor(Color::Red);
                                cout << "NOTE: ";

                                Color::setTextColor(Color::Yellow);
                                cout << "NO ";

                                Color::setTextColor(Color::BrightGreen);
                                cout << "SPACES

";

                                Color::setTextColor(Color::Blue);
                                cout <<

"REQUIRED" << endl;

                                continue;
                                }

cin.ignore(numeric_limits<streamsize>::max(), '\n');

                                if (bedrooms <= 0) {
                                    system("cls");
                                    Color::setTextColor(Color::White);
                                    cout << "Number of bedrooms cannot be ";
                                    Color::setTextColor(Color::Red);
                                    cout << "zero ";

                                    Color::setTextColor(Color::BrightCyan);
                                    cout << "or ";

                                    Color::setTextColor(Color::Red);
                                    cout <<

"negative. ";

                                    Color::setTextColor(Color::White);
                                    cout << "Please

enter a ";

                                    Color::setTextColor(Color::BrightGreen);
                                    cout << "valid ";

                                    Color::setTextColor(Color::White);
```



```
of ";
cout << "number

Color::setTextColor(Color::Yellow);
cout <<
"bedrooms." << endl;

Color::setTextColor(Color::BrightCyan);
cout << "CREATE PROPERTY TO
SELL" << endl;

Color::setTextColor(Color::Red);
cout << "NOTE: ";

Color::setTextColor(Color::Yellow);
cout << "NO ";

Color::setTextColor(Color::BrightGreen);
cout << "SPACES
";

Color::setTextColor(Color::Blue);
cout <<
"REQUIRED" << endl;

    continue;
}

// Valid input received, break out of the loop
break;

} while (true);

do {
    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Property Bathrooms";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);

    if (!(cin >> bathrooms)) {
        system("cls");
        cin.clear();

cin.ignore(numeric_limits<streamsize>::max(), '\n');
        Color::setTextColor(Color::Red);
        cout << "Invalid ";
        Color::setTextColor(Color::White);
        cout << "input.

Please enter a ";

        Color::setTextColor(Color::BrightGreen);
        cout << "valid ";

        Color::setTextColor(Color::White);
        cout << "number
of ";

        Color::setTextColor(Color::Yellow);

        cout <<
"bathrooms." << endl;
```



```
Color::setTextColor(Color::BrightCyan);
    cout << "CREATE PROPERTY TO
SELL" << endl;
    Color::setTextColor(Color::Red);
    cout << "NOTE: ";

    Color::setTextColor(Color::Yellow);
    cout << "NO ";

    Color::setTextColor(Color::BrightGreen);
    cout << "SPACES
";

    Color::setTextColor(Color::Blue);
    cout <<
"REQUIRED" << endl;
        continue;
    }

    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    if (bathrooms <= 0) {
        system("cls");
        Color::setTextColor(Color::White);
        cout << "Number of bathrooms cannot be ";
        Color::setTextColor(Color::Red);
        cout << "zero ";

        Color::setTextColor(Color::BrightCyan);
        cout << "or ";

        Color::setTextColor(Color::Red);
        cout <<
"negative. ";

        Color::setTextColor(Color::White);
        cout << "Please
enter a ";

        Color::setTextColor(Color::BrightGreen);
        cout << "valid ";

        Color::setTextColor(Color::White);
        cout << "number
of ";

        Color::setTextColor(Color::Yellow);
        cout <<
"bathrooms." << endl;

        Color::setTextColor(Color::BrightCyan);
        cout << "CREATE PROPERTY TO
SELL" << endl;
        Color::setTextColor(Color::Red);
        cout << "NOTE: ";

        Color::setTextColor(Color::Yellow);
        cout << "NO ";
```



```
Color::setTextColor(Color::BrightGreen);
cout << "SPACES
";

Color::setTextColor(Color::Blue);
cout <<

"REQUIRED" << endl;
    continue;
}

// Valid input received, break out of the loop
break;

} while (true);

do {
    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Property Lot Area";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);

    if (!(cin >> lotArea)) {
        system("cls");
        cin.clear();

cin.ignore(numeric_limits<streamsize>::max(), '\n');
        Color::setTextColor(Color::Red);
        cout << "Invalid ";
        Color::setTextColor(Color::White);
        cout << "input.

Please enter a ";

        Color::setTextColor(Color::BrightGreen);
        cout << "valid ";

        Color::setTextColor(Color::Yellow);
        cout << "lot
area." << endl;

Color::setTextColor(Color::BrightCyan);
        cout << "CREATE PROPERTY TO
SELL" << endl;

        Color::setTextColor(Color::Red);
        cout << "NOTE: ";

        Color::setTextColor(Color::Yellow);
        cout << "NO ";

        Color::setTextColor(Color::BrightGreen);
        cout << "SPACES
";

        Color::setTextColor(Color::Blue);
        cout <<

"REQUIRED" << endl;
        continue;
}
```



```
cin.ignore(numeric_limits<streamsize>::max(), '\n');

    if (lotArea <= 0) {
        system("cls");
        Color::setTextColor(Color::White);
        cout << "Lot area cannot be ";
        Color::setTextColor(Color::Red);
        cout << "zero ";

        Color::setTextColor(Color::BrightCyan);
        cout << "or ";

        Color::setTextColor(Color::Red);
        cout <<

"negative. ";

        Color::setTextColor(Color::White);
        cout << "Please
enter a ";

        Color::setTextColor(Color::BrightGreen);
        cout << "valid ";

        Color::setTextColor(Color::White);
        cout << "number
of ";

        Color::setTextColor(Color::Yellow);
        cout << "lot
area." << endl;

        Color::setTextColor(Color::BrightCyan);
        cout << "CREATE PROPERTY TO
SELL" << endl;

        Color::setTextColor(Color::Red);
        cout << "NOTE: ";

        Color::setTextColor(Color::Yellow);
        cout << "NO ";

        Color::setTextColor(Color::BrightGreen);
        cout << "SPACES
";

        Color::setTextColor(Color::Blue);
        cout <<

"REQUIRED" << endl;

        continue;
    }

    // Valid input received, break out of the loop
    break;

} while (true);

do {
    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
```



```
        Color::setTextColor(Color::BrightGreen);
        cout << "Property Price";
        Color::setTextColor(Color::Yellow);
        cout << ": ";
        Color::setTextColor(Color::White);

        if (!(cin >> price)) {
            system("cls");
            cin.clear();

cin.ignore(numeric_limits<streamsize>::max(), '\n');
            Color::setTextColor(Color::Red);
            cout << "Invalid";
            Color::setTextColor(Color::White);
            cout << " input.

Please enter a ";

            Color::setTextColor(Color::BrightGreen);
            cout << "valid ";

            Color::setTextColor(Color::Yellow);
            cout << "price."
<< endl;

Color::setTextColor(Color::BrightCyan);
            cout << "CREATE PROPERTY TO
SELL" << endl;

            Color::setTextColor(Color::Red);
            cout << "NOTE: ";

            Color::setTextColor(Color::Yellow);
            cout << "NO ";

            Color::setTextColor(Color::BrightGreen);
            cout << "SPACES
";

            Color::setTextColor(Color::Blue);
            cout <<
"REQUIRED" << endl;
            continue;
        }

cin.ignore(numeric_limits<streamsize>::max(), '\n');

        if (price <= 0) {
            system("cls");
            Color::setTextColor(Color::White);
            cout << "Price cannot be ";
            Color::setTextColor(Color::Red);
            cout << "zero ";

            Color::setTextColor(Color::BrightCyan);
            cout << "or ";

            Color::setTextColor(Color::Red);
            cout <<
"negative. ";

            Color::setTextColor(Color::White);
```



```
enter a ";

    cout << "Please

    Color::setTextColor(Color::BrightGreen);
    cout << "valid ";

    Color::setTextColor(Color::Yellow);
    cout << "price."

<< endl;

Color::setTextColor(Color::BrightCyan);
    cout << "CREATE PROPERTY TO
SELL" << endl;

    Color::setTextColor(Color::Red);
    cout << "NOTE: ";

    Color::setTextColor(Color::Yellow);
    cout << "NO ";

    Color::setTextColor(Color::BrightGreen);
    cout << "SPACES

";

    Color::setTextColor(Color::Blue);
    cout <<

"REQUIRED" << endl;

        continue;
    }

        system("cls");
    // Valid input received, break out of the loop
    break;

} while (true);

char ans;

do{
    Color::setTextColor(Color::BrightCyan);
    cout << "CREATE PROPERTY TO SELL" << endl;

    Color::setTextColor(Color::BrightYellow);
    cout << "-----" << endl;
    // Display property details here as needed
    Color::setTextColor(Color::BrightRed);
    cout << "PROPERTY'S DETAILS"<<endl;
    Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
    cout << "ID: " ;

    Color::setTextColor(Color::BrightCyan);
    cout << propertyId << endl;

    Color::setTextColor(Color::Yellow);
    cout << "Property ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Type: " ;

    Color::setTextColor(Color::BrightCyan);
```




```
cout << propertyType <<
endl;

Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
cout << "Address: " ;

Color::setTextColor(Color::BrightCyan);
cout<< propertyAddress <<
endl;

Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
cout << "Lot Title: " ;

Color::setTextColor(Color::BrightCyan);
cout<< propertyLotTitle <<
endl;

Color::setTextColor(Color::Yellow);
cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
cout << "Bedrooms: " ;

Color::setTextColor(Color::BrightCyan);
cout<< bedrooms << endl;

Color::setTextColor(Color::Yellow);
cout << "No: of ";
Color::setTextColor(Color::BrightGreen);
cout << "Bathrooms: " ;

Color::setTextColor(Color::BrightCyan);
cout<< bathrooms << endl;

Color::setTextColor(Color::Yellow);
cout << "Property ";
Color::setTextColor(Color::BrightGreen);
cout << "Lot Area: " ;

Color::setTextColor(Color::BrightCyan);
cout<< lotArea ;

Color::setTextColor(Color::Blue);
cout << " sqm" << endl;

Color::setTextColor(Color::Yellow);
cout <<fixed<<setprecision(2)<< "Price: ";
Color::setTextColor(Color::BrightGreen);
cout << "PHP " ;

Color::setTextColor(Color::BrightCyan);
cout<< price << endl;

Color::setTextColor(Color::Yellow);
cout << "Available: " ;
Color::setTextColor(isAvailable ?
Color::BrightCyan : Color::BrightRed);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout<< (isAvailable ? "Yes" :
"No") << endl;

Color::setTextColor(Color::BrightYellow);
cout << "-----" << endl;
Color::setTextColor(Color::BrightYellow);
cout << "This property will be created and post
accordingly." << endl;
Color::setTextColor(Color::Yellow);
cout << "Do you want to
continue? (Y/N): " ;

cin>>ans;
if(ans=='Y' || ans=='y'){
//string newId = generatePropertyUniqueID();
// Generate a unique ID

ofstream createProperty("globalProperty.txt",
ios::app);

createProperty << propertyId << "," <<
propertyType << "," << propertyAddress << "," << propertyLotTitle << ","
<<bedrooms << "," << bathrooms << "," << lotArea << "," << price << "," <<
is_available << endl;

createProperty.close();

system("cls");

Color::setTextColor(Color::BrightGreen);
cout << "Property has been created
and post successfully." << endl;
Color::setTextColor(Color::White);
break;
}else if (ans=='N' || ans=='n'){
system("cls");

Color::setTextColor(Color::BrightGreen);
cout << "Creating property has been
aborted." << endl;
break;
}else{
system("cls");
Color::setTextColor(Color::Red);
cout << "Invalid";
Color::setTextColor(Color::White);
cout << " input.

Please enter a ";

Color::setTextColor(Color::BrightGreen);
cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "choice."
<< endl;

Color::setTextColor(Color::BrightCyan);
continue;
}
}while(true);
}

void updateProperty() {
ifstream inputFile("globalProperty.txt");
```



```

ofstream tempFile("tempProperty.txt");

string propertyIdToUpdate;
system("cls");
Color::setTextColor(Color::Cyan);
cout << "UPDATE PROPERTY" << endl;
Color::setTextColor(Color::Yellow);
cout << "Enter Property ID: ";
Color::setTextColor(Color::White);
getline(cin,propertyIdToUpdate);

if (!inputFile.is_open() || !tempFile.is_open()) {
    cout << "Unable to open the property database." << endl;
    return;
}

bool propertyFound = false;
string line;
while (getline(inputFile, line)) {
    istringstream ss(line);
    string id;
    getline(ss, id, ',');

    if (id == propertyIdToUpdate) {
        propertyFound = true;

        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
        ss >> bedrooms;
        ss.ignore(); // Ignore the comma
        ss >> bathrooms;
        ss.ignore(); // Ignore the comma
        ss >> lotArea;
        ss.ignore(); // Ignore the comma
        ss >> price;
        ss.ignore(); // Ignore the comma
        ss >> is_available;

        bool updated = false; // Flag to track if any update was made
        int updatedCount=0;

        system("cls");

        while (true) {
            int choice;

            Color::setTextColor(Color::Cyan);
            cout << "UPDATE
PROPERTY" << endl;

            Color::setTextColor(Color::White);
            cout << "Property ";
            Color::setTextColor(Color::BrightGreen);
            cout << "found. "

<< endl;

            Color::setTextColor(Color::BrightYellow);
            cout << "-----"

<< endl;

            showPropertyById(propertyIdToUpdate);
            Color::setTextColor(Color::White);

```



```

would you like to ";
                                cout << "What

                                Color::setTextColor(Color::Red);
                                cout <<
"update?" << endl;

                                Color::setTextColor(Color::BrightCyan);
                                cout << "[";
                                Color::setTextColor(Color::BrightRed);
                                cout << "1";

                                Color::setTextColor(Color::BrightCyan);
                                cout << "]" ";

                                Color::setTextColor(Color::BrightGreen);
                                cout <<
"Property Type" << endl;

                                Color::setTextColor(Color::BrightCyan);
                                cout << "[";
                                Color::setTextColor(Color::BrightRed);
                                cout << "2";

                                Color::setTextColor(Color::BrightCyan);
                                cout << "]" ";

                                Color::setTextColor(Color::BrightGreen);
                                cout <<
"Property Address" << endl;

                                Color::setTextColor(Color::BrightCyan);
                                cout << "[";
                                Color::setTextColor(Color::BrightRed);
                                cout << "3";

                                Color::setTextColor(Color::BrightCyan);
                                cout << "]" ";

                                Color::setTextColor(Color::BrightGreen);
                                cout <<
"Property Lot Title" << endl;

                                Color::setTextColor(Color::BrightCyan);
                                cout << "[";
                                Color::setTextColor(Color::BrightRed);
                                cout << "4";

                                Color::setTextColor(Color::BrightCyan);
                                cout << "]" ";

                                Color::setTextColor(Color::BrightGreen);
                                cout << "Number
of Bedrooms" << endl;

                                Color::setTextColor(Color::BrightCyan);
                                cout << "[";
                                Color::setTextColor(Color::BrightRed);
                                cout << "5";
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::BrightCyan);
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout << "Number
of Bathrooms" << endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "6";

Color::setTextColor(Color::BrightCyan);
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout <<

"Property Lot Area" << endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "7";

Color::setTextColor(Color::BrightCyan);
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout << "Price"
<< endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "8";

Color::setTextColor(Color::BrightCyan);
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout << "Exit and
abort the update." << endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "9";

Color::setTextColor(Color::BrightCyan);
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout << "Exit and
save the update." << endl;

Color::setTextColor(Color::Yellow);
cout << "Enter your choice: ";
Color::setTextColor(Color::White);
cin >> choice;
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
if (choice == 9) {
    Color::setTextColor(Color::BrightGreen);
    updatedCount=0;
    cout << "Exiting update mode and save all the
made update." << endl;
    break;
}

if(choice == 8){
    Color::setTextColor(Color::BrightGreen);
    cout << "Exiting update mode and abort all the
changes." << endl;
    updatedCount=0;
    updated = false;
    break;
}

switch (choice) {
    case 1:
    {
        system("cls");

        string newPropertyType;

        Color::setTextColor(Color::Cyan);

        cout << "UPDATE PROPERTY" << endl;

        Color::setTextColor(Color::BrightYellow);

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

        showPropertyById(propertyIdToUpdate);

        Color::setTextColor(Color::Yellow);

        cout << "Enter updated Property Type: ";

        Color::setTextColor(Color::BrightYellow);

        cin.ignore(); // Clear the input buffer

        getline(cin, newPropertyType);

        system("cls");

        if
(newPropertyType == propertyType) {

            system("cls");

            Color::setTextColor(Color::White);

            cout << "No change in Property ";

            Color::setTextColor(Color::BrightGreen);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
        cout << "Type." << endl;
    }
else {
    do{

        Color::setTextColor(Color::Cyan);

        cout << "UPDATE
PROPERTY" << endl;

        Color::setTextColor(Color::BrightRed);

        cout << "OLD " << endl;

        Color::setTextColor(Color::BrightYellow);

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

        showPropertyById(propertyIdToUpdate);

        Color::setTextColor(Color::BrightGreen);

        cout << "NEW" << endl;

        Color::setTextColor(Color::BrightYellow);

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

        showUpdatePropertyByType(propertyIdToUpdate,newPrope
rtyType);

        char ans;

        cout << "Do you want to continue ? (Y/N) : ";

        cin >> ans;

        if(ans=='Y' || ans=='y' ){

            system("cls");

            updatedCount++;

            Color::setTextColor(Color::BrightCyan);

            cout << "[";

            Color::setTextColor(Color::Red);

            cout << updatedCount;
```



```
Color::setTextColor(Color::BrightCyan);

cout << "]" ";

Color::setTextColor(Color::BrightGreen);

cout << "Update
has been made. Waiting to be saved." << endl;

propertyType = newPropertyType;

updated = true;

break;

} else if
(ans=='N' || ans=='n'){

system("cls");

Color::setTextColor(Color::BrightGreen);

cout << "Update has been
aborted." << endl;

break;

} else{

system("cls");

Color::setTextColor(Color::Red);

cout << "Invalid";

Color::setTextColor(Color::White);

cout << " input.
Please enter a ";

Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "choice."
<< endl;

Color::setTextColor(Color::BrightCyan);

continue;
```




```
    }  
    }while(true);  
    }  
  
    }  
    break;  
    case 2:  
  
    {  
  
        system("cls");  
  
        string newPropertyAddress;  
  
        Color::setTextColor(Color::Cyan);  
  
        cout << "UPDATE PROPERTY" << endl;  
  
        Color::setTextColor(Color::BrightYellow);  
  
        cout << "-----" << endl;  
  
        showPropertyById(propertyIdToUpdate);  
  
        Color::setTextColor(Color::Yellow);  
  
        cout << "Enter updated Property Address: ";  
  
        Color::setTextColor(Color::BrightYellow);  
  
        cin.ignore(); // Clear the input buffer  
        getline(cin, newPropertyAddress);  
  
        system("cls");  
  
        if  
(newPropertyAddress == propertyAddress) {  
            system("cls");  
            Color::setTextColor(Color::White);  
  
            cout << "No change in Property ";  
            Color::setTextColor(Color::BrightGreen);  
  
            cout << "Address." << endl;  
        }  
    else {  
  
        do{
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::Cyan);

cout << "UPDATE

PROPERTY" << endl;

Color::setTextColor(Color::BrightRed);

cout << "OLD " << endl;

Color::setTextColor(Color::BrightYellow);

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

showPropertyById(propertyIdToUpdate);

Color::setTextColor(Color::BrightGreen);

cout << "NEW" << endl;

Color::setTextColor(Color::BrightYellow);

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

showUpdatePropertyByAddress(propertyIdToUpdate,newPro
pertyAddress);

char ans;

cout << "Do you want to continue ? (Y/N) : ";

cin >> ans;

if(ans=='Y' || ans=='y'){

    system("cls");

    updatedCount++;

    Color::setTextColor(Color::BrightCyan);

    cout << "[";

    Color::setTextColor(Color::Red);

    cout << updatedCount;

    Color::setTextColor(Color::BrightCyan);

    cout << "]" ;

    Color::setTextColor(Color::BrightGreen);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout << "Update
has been made. Waiting to be saved." << endl;

propertyAddress = newPropertyAddress;

updated = true;

break;

}else if(ans=='N' || ans=='n'){

system("cls");

Color::setTextColor(Color::BrightGreen);

cout << "Update has been
aborted." << endl;

break;

}else{

system("cls");

Color::setTextColor(Color::Red);

cout << "Invalid";

Color::setTextColor(Color::White);

cout << " input.

Please enter a ";

Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "choice."

<< endl;

Color::setTextColor(Color::BrightCyan);

continue;

}

}while(true);

}

}

break;
case 3:

{
```



```
        system("cls");

        string newPropertyLotTitle;

        Color::setTextColor(Color::Cyan);

        cout << "UPDATE PROPERTY" << endl;

        Color::setTextColor(Color::BrightYellow);

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

        showPropertyById(propertyIdToUpdate);

        Color::setTextColor(Color::Yellow);

        cout << "Enter updated Property Lot Title: ";

        Color::setTextColor(Color::BrightYellow);

        cin.ignore(); // Clear the input buffer

        getline(cin, newPropertyLotTitle);

        system("cls");

        if
(newPropertyLotTitle == propertyLotTitle) {

            system("cls");

            Color::setTextColor(Color::White);

            cout << "No change in Property ";

            Color::setTextColor(Color::BrightGreen);

            cout << "Address." << endl;

        }

        else {

            do{

                Color::setTextColor(Color::Cyan);

                cout << "UPDATE

PROPERTY" << endl;

                Color::setTextColor(Color::BrightRed);

                cout << "OLD " << endl;

                Color::setTextColor(Color::BrightYellow);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



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

showPropertyById(propertyIdToUpdate);

Color::setTextColor(Color::BrightGreen);

cout << "NEW" << endl;

Color::setTextColor(Color::BrightYellow);

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

showUpdatePropertyByLotTitle(propertyIdToUpdate,newPro
pertyLotTitle);

char ans;

cout << "Do you want to continue ? (Y/N) : ";

cin >> ans;

if(ans=='Y' || ans=='y'){

    system("cls");

    updatedCount++;

    Color::setTextColor(Color::BrightCyan);

    cout << "[";

    Color::setTextColor(Color::Red);

    cout << updatedCount;

    Color::setTextColor(Color::BrightCyan);

    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);

    cout << "Update
has been made. Waiting to be saved." << endl;

    propertyLotTitle = newPropertyLotTitle;

    updated = true;

    break;

}else if(ans=='N' || ans=='n'){

    system("cls");
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::BrightGreen);

cout << "Update has been
aborted." << endl;

break;

}else{

system("cls");

Color::setTextColor(Color::Red);

cout << "Invalid";

Color::setTextColor(Color::White);

cout << " input.

Please enter a ";

Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "choice."

<< endl;

Color::setTextColor(Color::BrightCyan);

continue;

}

}while(true);

}

break;
case 4:

{

int newBedrooms;

bool isValid = false;

while (!isValid) {

system("cls");

Color::setTextColor(Color::Cyan);
```



```
cout << "UPDATE PROPERTY" << endl;

Color::setTextColor(Color::BrightYellow);

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

showPropertyById(propertyIdToUpdate);

Color::setTextColor(Color::Yellow);

cout << "Enter updated Number of Bedrooms: ";

Color::setTextColor(Color::BrightYellow);

if (!(cin >> newBedrooms) || newBedrooms <= 0) {

    // Handle invalid input

    cin.clear();

    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    Color::setTextColor(Color::Red);

    cout << "Invalid";

    Color::setTextColor(Color::White);

    cout << " input. Please enter

a ";

Color::setTextColor(Color::BrightGreen);

    cout << "valid";

Color::setTextColor(Color::White);

    cout << " number of ";

Color::setTextColor(Color::Yellow);

    cout << "bedrooms";

Color::setTextColor(Color::White);

    cout << " greater than ";

Color::setTextColor(Color::Red);

    cout << "zero." << endl;
```



```
} else {  
  
    isValid = true;  
  
}  
  
}  
  
system("cls");  
  
if (newBedrooms == bedrooms) {  
  
    system("cls");  
  
    Color::setTextColor(Color::White);  
  
    cout << "No change in Property ";  
  
    Color::setTextColor(Color::BrightGreen);  
  
    cout << "Bedrooms." << endl;  
  
} else {  
  
    do{  
  
        Color::setTextColor(Color::Cyan);  
  
        cout << "UPDATE  
PROPERTY" << endl;  
  
        Color::setTextColor(Color::BrightRed);  
  
        cout << "OLD " << endl;  
  
        Color::setTextColor(Color::BrightYellow);  
  
        cout << "-----" << endl;  
  
        showPropertyById(propertyIdToUpdate);  
  
        Color::setTextColor(Color::BrightGreen);  
  
        cout << "NEW" << endl;  
  
        Color::setTextColor(Color::BrightYellow);  
  
        cout << "-----" << endl;
```




Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
showUpdatePropertyByBedrooms(propertyIdToUpdate,new
Bedrooms);

char ans;

cout << "Do you want to continue ? (Y/N) : ";

cin >> ans;

if(ans=='Y' || ans=='y'){

    system("cls");

    bedrooms = newBedrooms;

    updated = true;

    // Perform further actions for the
updated number of bedrooms if needed

    updatedCount++;

    Color::setTextColor(Color::BrightCyan);

    cout << "[";

    Color::setTextColor(Color::Red);

    cout << updatedCount;

    Color::setTextColor(Color::BrightCyan);

    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);

    cout << "Update
has been made. Waiting to be saved." << endl;

    break;

}else if (ans=='N' || ans=='n'

){

    system("cls");

    Color::setTextColor(Color::BrightGreen);

    cout << "Update has been
aborted." << endl;

    break;

}else{

    system("cls");

    Color::setTextColor(Color::Red);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
        cout << "Invalid";

        Color::setTextColor(Color::White);

        cout << " input.

Please enter a ";

        Color::setTextColor(Color::BrightGreen);

        cout << "valid ";

        Color::setTextColor(Color::Yellow);

        cout << "choice."

<< endl;

Color::setTextColor(Color::BrightCyan);

        continue;

    }

    }while(true);

}

}

    break;
case 5:

{

    int newBathrooms;

    bool isValid = false;

    while (!isValid) {

        system("cls");

        Color::setTextColor(Color::Cyan);

        cout << "UPDATE PROPERTY" << endl;

        Color::setTextColor(Color::BrightYellow);

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

        showPropertyById(propertyIdToUpdate);

        Color::setTextColor(Color::Yellow);

        cout << "Enter updated Number of Bathrooms: ";
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::BrightYellow);

if (!(cin >> newBathrooms) || newBathrooms <= 0) {

    // Handle invalid input

    cin.clear();

    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    Color::setTextColor(Color::Red);

    cout << "Invalid";

    Color::setTextColor(Color::White);

        cout << " input. Please enter

a ";

Color::setTextColor(Color::BrightGreen);

        cout << "valid";

Color::setTextColor(Color::White);

        cout << " number of ";

Color::setTextColor(Color::Yellow);

        cout << "bathrooms";

Color::setTextColor(Color::White);

        cout << " greater than ";

Color::setTextColor(Color::Red);

        cout << "zero." << endl;

    } else {

        isValid = true;

    }

}

system("cls");

if (newBathrooms == bathrooms) {
```



```
Color::setTextColor(Color::White);

cout << "No change in Property ";

Color::setTextColor(Color::BrightGreen);

cout << "Bathrooms." << endl;

} else {

    do{

        Color::setTextColor(Color::Cyan);

        cout << "UPDATE

PROPERTY" << endl;

        Color::setTextColor(Color::BrightRed);

        cout << "OLD " << endl;

        Color::setTextColor(Color::BrightYellow);

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

        showPropertyById(propertyIdToUpdate);

        Color::setTextColor(Color::BrightGreen);

        cout << "NEW" << endl;

        Color::setTextColor(Color::BrightYellow);

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

        showUpdatePropertyByBathrooms(propertyIdToUpdate,new
Bathrooms);

        char ans;

        cout << "Do you want to continue ? (Y/N) : ";

        cin >> ans;

        if(ans=='Y' || ans=='y'){

            system("cls");

            bathrooms = newBathrooms;

            updated = true;
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
// Perform further actions for the  
updated number of bathrooms if needed
```

```
updatedCount++;
```

```
Color::setTextColor(Color::BrightCyan);
```

```
cout << "[";
```

```
Color::setTextColor(Color::Red);
```

```
cout << updatedCount;
```

```
Color::setTextColor(Color::BrightCyan);
```

```
cout << "]" ;
```

```
Color::setTextColor(Color::BrightGreen);
```

```
cout << "Update  
has been made. Waiting to be saved." << endl;
```

```
break;
```

```
Color::setTextColor(Color::Yellow);
```

```
}else if (ans=='N' || ans=='n'){
```

```
system("cls");
```

```
Color::setTextColor(Color::BrightGreen);
```

```
cout << "Update has been  
aborted." << endl;
```

```
break;
```

```
}else{
```

```
system("cls");
```

```
Color::setTextColor(Color::Red);
```

```
cout << "Invalid";
```

```
Color::setTextColor(Color::White);
```

```
cout << " input.  
Please enter a ";
```

```
Color::setTextColor(Color::BrightGreen);
```

```
cout << "valid ";
```

```
Color::setTextColor(Color::Yellow);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
<< endl;

cout << "choice."

Color::setTextColor(Color::BrightCyan);

        continue;

    }

}while(true);

}

}

    break;
case 6:
    {

double newLotArea;

bool isValid = false;

while (!isValid) {

    system("cls");

    Color::setTextColor(Color::Cyan);

    cout << "UPDATE PROPERTY" << endl;

Color::setTextColor(Color::BrightYellow);

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

showPropertyById(propertyIdToUpdate);

    Color::setTextColor(Color::Yellow);

    cout << "Enter updated lot area: ";

    Color::setTextColor(Color::BrightYellow);

    if (!(cin >> newLotArea) || newLotArea <= 0) {

        // Handle invalid input

        cin.clear();

        cin.ignore(numeric_limits<streamsize>::max(), '\n');

        Color::setTextColor(Color::Red);

        cout << "Invalid";
```



```
Color::setTextColor(Color::White);

cout << " input. Please enter

a ";

Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "price";

Color::setTextColor(Color::White);

cout << " greater than ";

Color::setTextColor(Color::Red);

cout << "zero." << endl;

} else {

    isValid = true;

}

}

system("cls");

if (newLotArea == lotArea) {

    Color::setTextColor(Color::White);

    cout << "No change in Property ";

    Color::setTextColor(Color::BrightGreen);

    cout << "Lot Area." << endl;

} else {

    do{

        Color::setTextColor(Color::Cyan);

        cout << "UPDATE

PROPERTY" << endl;

        Color::setTextColor(Color::BrightRed);

        cout << "OLD " << endl;
```



```
Color::setTextColor(Color::BrightYellow);

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

showPropertyById(propertyIdToUpdate);

Color::setTextColor(Color::BrightGreen);

    cout << "NEW" << endl;

Color::setTextColor(Color::BrightYellow);

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

showUpdatePropertyByLotArea(propertyIdToUpdate,newLot
Area);

    char ans;

    cout << "Do you want to continue ? (Y/N) : ";

    cin >> ans;

    if(ans=='Y' || ans=='y'){

        system("cls");

        lotArea = newLotArea;

        updated = true;

        // You might perform further
actions for the updated price here

        updatedCount++;

        Color::setTextColor(Color::BrightCyan);

        cout << "[";

        Color::setTextColor(Color::Red);

        cout << updatedCount;

        Color::setTextColor(Color::BrightCyan);

        cout << "]" ;

        Color::setTextColor(Color::BrightGreen);

        cout << "Update
has been made. Waiting to be saved." << endl;
```




Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
        break;

        }else if(ans=='N' || ans=='n'){

            system("cls");

Color::setTextColor(Color::BrightGreen);

            cout << "Update has been

aborted." << endl;

            break;

        }else{

            system("cls");

Color::setTextColor(Color::Red);

            cout << "Invalid";

Color::setTextColor(Color::White);

            cout << " input.

Please enter a ";

Color::setTextColor(Color::BrightGreen);

            cout << "valid ";

Color::setTextColor(Color::Yellow);

            cout << "choice."

<< endl;

Color::setTextColor(Color::BrightCyan);

            continue;

        }

    }while(true);

}

}

        break;
    case 7:
    {

        double newPrice;

        bool isValid = false;

        while (!isValid) {
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
system("cls");

Color::setTextColor(Color::Cyan);

cout << "UPDATE PROPERTY" << endl;

Color::setTextColor(Color::BrightYellow);

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

showPropertyById(propertyIdToUpdate);

Color::setTextColor(Color::Yellow);

cout << "Enter updated Price: ";

Color::setTextColor(Color::BrightYellow);

if (!(cin >> newPrice) || newPrice <= 0) {

    // Handle invalid input

    cin.clear();

    cin.ignore(numeric_limits<streamsize>::max(), '\n');

    Color::setTextColor(Color::Red);

    cout << "Invalid";

    Color::setTextColor(Color::White);

    cout << " input. Please enter

a ";

Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "price";

Color::setTextColor(Color::White);

cout << " greater than ";

Color::setTextColor(Color::Red);

cout << "zero." << endl;
```



```
    } else {  
  
        isValid = true;  
  
    }  
}  
  
    system("cls");  
  
if (newPrice == price) {  
  
    Color::setTextColor(Color::White);  
  
    cout << "No change in Property ";  
  
    Color::setTextColor(Color::BrightGreen);  
  
    cout << "Price." << endl;  
  
} else {  
  
    do{  
  
        Color::setTextColor(Color::Cyan);  
  
        cout << "UPDATE  
PROPERTY" << endl;  
  
        Color::setTextColor(Color::BrightRed);  
  
        cout << "OLD " << endl;  
  
        Color::setTextColor(Color::BrightYellow);  
  
        cout << "-----" << endl;  
  
        showPropertyById(propertyIdToUpdate);  
  
        Color::setTextColor(Color::BrightGreen);  
  
        cout << "NEW" << endl;  
  
        Color::setTextColor(Color::BrightYellow);  
  
        cout << "-----" << endl;  
  
        showUpdatePropertyByPrice(propertyIdToUpdate,newPrice);  
  
        char ans;  
  
        cout << "Do you want to continue ? (Y/N) : ";
```



```
cin >> ans;

if(ans=='Y' || ans=='y'){

    system("cls");

    price = newPrice;

    updated = true;

    // You might perform further
actions for the updated price here

    updatedCount++;

    Color::setTextColor(Color::BrightCyan);

    cout << "[";

    Color::setTextColor(Color::Red);

    cout << updatedCount;

    Color::setTextColor(Color::BrightCyan);

    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);

    cout << "Update
has been made. Waiting to be saved." << endl;

    break;

}else if(ans=='N' || ans=='n'){

    system("cls");

    Color::setTextColor(Color::BrightGreen);

    cout << "Update has been
aborted." << endl;

    break;

}else{

    system("cls");

    Color::setTextColor(Color::Red);

    cout << "Invalid";

    Color::setTextColor(Color::White);

    cout << " input.

Please enter a ";
```



```
Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "choice."

<< endl;

Color::setTextColor(Color::BrightCyan);

        continue;

    }

    }while(true);

}

}

        break;
    default:

Color::setTextColor(Color::Red);

        cout << "Invalid

choice. ";

Color::setTextColor(Color::White);

        cout << "Please enter your choice between ";

Color::setTextColor(Color::BrightCyan);

        cout << "[";

Color::setTextColor(Color::BrightGreen);

        cout << "1";

        Color::setTextColor(Color::BrightCyan);

        cout << "] ";

Color::setTextColor(Color::Yellow);

        cout << "to ";

Color::setTextColor(Color::BrightCyan);

        cout << "[";

Color::setTextColor(Color::BrightGreen);

        cout << "6";

        Color::setTextColor(Color::BrightCyan);

        cout << "] ";

Color::setTextColor(Color::White);
```



```

cout << "only.\n";
    continue; // Repeat the loop for a valid choice
}
}

// Write the updated or unchanged data to the temp file
if (updated) {
    tempFile << propertyIdToUpdate << "," << propertyType << ","
<< propertyAddress << "," << propertyLotTitle << ","

        << bedrooms << "," << bathrooms << "," << lotArea << "," <<
price << "," << is_available << endl;
        system("cls");
        Color::setTextColor(Color::White);
        cout << "Property with ";
        Color::setTextColor(Color::Red);

                                cout << "ID " ;

        Color::setTextColor(Color::BrightCyan);

                                cout << "[";

        Color::setTextColor(Color::BrightGreen);

                                cout          <<

propertyIdToUpdate ;

        Color::setTextColor(Color::BrightCyan);

                                cout << "]";

        Color::setTextColor(Color::White);

                                cout  << "  has

been updated ";

        Color::setTextColor(Color::BrightGreen);

                                cout          <<

"successfully." << endl;

        Color::setTextColor(Color::White);
    } else {
        // If no changes were made, write the existing line to the temp
file
        system("cls");
        Color::setTextColor(Color::White);
        cout << "No ";
        Color::setTextColor(Color::BrightGreen);

                                cout << "changes

";

        Color::setTextColor(Color::Red);

                                cout          <<

"detected";

        Color::setTextColor(Color::White);

                                cout << ". ";

        Color::setTextColor(Color::BrightGreen);

                                cout << " Update

";

        Color::setTextColor(Color::Red);

```



```
        cout << endl;
        "canceled";

        Color::setTextColor(Color::White);
        cout << " " << endl;
    end;
    tempFile << line << endl;
}
} else {
    tempFile << line << endl; // Write the existing line to the temp
file
}
}

inputFile.close();
tempFile.close();

if (propertyFound) {
    remove("globalProperty.txt");
    rename("tempProperty.txt", "globalProperty.txt");
} else {
    remove("tempProperty.txt");
    system("cls");
    Color::setTextColor(Color::White);
    cout << "Property with ";
    Color::setTextColor(Color::Yellow);
    cout << "ID " ;
    Color::setTextColor(Color::BrightCyan);
    cout << "[" ;

    Color::setTextColor(Color::BrightGreen);
    cout << propertyIdToUpdate ;
    Color::setTextColor(Color::BrightCyan);
    cout << "]" ;
    Color::setTextColor(Color::White);
    cout << " not ";
    Color::setTextColor(Color::Red);
    cout << "found." << endl;
}
}

//DeleteProperty Constructor
void deleteProperty() {
    ifstream inputFile("globalProperty.txt");
    ofstream tempFile("tempProperty.txt");
    string propertyIdToDelete;

    Color::setTextColor(Color::Cyan);
    cout << "DELETE PROPERTY" << endl;
    Color::setTextColor(Color::Yellow);
    cout << "Enter Property ID: ";
    Color::setTextColor(Color::White);
    while(!cin>>propertyIdToDelete){
        Color::setTextColor(Color::Yellow);
        cout << "Enter Property ID: ";
        Color::setTextColor(Color::White);
    }

    if (!inputFile.is_open() || !tempFile.is_open()) {
        cout << "Unable to open the property database." << endl;
    }
}
```



```
        return;
    }

    bool propertyFound = false;
    string line;
    while (getline(inputFile, line)) {
        istringstream ss(line);
        string id;
        getline(ss, id, ',');

        if (id == propertyIdToDelete) {
            propertyFound = true;
            continue; // Skip writing this line to temp file
        }

        tempFile << line << endl;
    }

    inputFile.close();
    tempFile.close();

    if (propertyFound) {
        char ans;
        do{
            system("cls");
            Color::setTextColor(Color::Cyan);
            cout << "DELETE PROPERTY" << endl;

            Color::setTextColor(Color::BrightYellow);
            cout << "-----" << endl;
            showPropertyById(propertyIdToDelete);
            Color::setTextColor(Color::BrightYellow);
            cout << "This property will be deleted and
remove from post accordingly." << endl;
            Color::setTextColor(Color::Yellow);
            cout << "Do you want to
continue? (Y/N): " ;

            cin>>ans;

            if(ans=='Y' || ans=='y'){
                remove("globalProperty.txt");
                rename("tempProperty.txt",
"globalProperty.txt");

                system("cls");

                Color::setTextColor(Color::White);
                cout << "Property with ";
                Color::setTextColor(Color::Red);
                cout << "ID " ;

                Color::setTextColor(Color::BrightCyan);
                cout << "[";

                Color::setTextColor(Color::BrightGreen);
                cout <<

propertyIdToDelete ;

                Color::setTextColor(Color::BrightCyan);
                cout << "]" ;
```




Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::White);
cout << " has
been ";

Color::setTextColor(Color::Red);
cout << "deleted
";

Color::setTextColor(Color::BrightGreen);
cout <<
"successfully." << endl;

Color::setTextColor(Color::White);
break;
}else if(ans=='N' || ans=='n'){
system("cls");

Color::setTextColor(Color::BrightGreen);
cout << "Deleting
property has been aborted." << endl;
break;
}else{
system("cls");
Color::setTextColor(Color::Red);
cout << "Invalid";
Color::setTextColor(Color::White);
cout << " input.
Please enter a ";
Color::setTextColor(Color::BrightGreen);
cout << "valid ";

Color::setTextColor(Color::Yellow);
cout << "choice."
<< endl;

Color::setTextColor(Color::BrightCyan);
continue;
}
}while(true);
} else {
system("cls");
remove("tempProperty.txt");
Color::setTextColor(Color::White);
cout << "Property with ";
Color::setTextColor(Color::Yellow);
cout << "ID ";
Color::setTextColor(Color::BrightCyan);
cout << "[";

Color::setTextColor(Color::BrightGreen);
cout << propertyIdToDelete ;
Color::setTextColor(Color::BrightCyan);
cout << "]";
Color::setTextColor(Color::White);
cout << " not ";
Color::setTextColor(Color::Red);
cout << "found." << endl;
Color::setTextColor(Color::White);
}
}
```



```
}

// Function to check if the generated PROPERTYID already
exists in the database
bool propertyIdExistsInDatabase(int idToCheck) {
    ifstream input("globalProperty.txt");
    int id;

    while (input >> id) {
        if (id == idToCheck) {
            input.close();
            return true; // ID exists in the database
        }
        // Skip the rest of the line
        input.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    input.close();
    return false; // ID doesn't exist in the database
}

// Function to check if the generated PROPERTYID already
exists in the database
bool propertyIdExists(string idToCheck) {
    ifstream input("globalProperty.txt");
    string line;
    string    propertyId,    propertyType,    propertyAddress,
propertyLotTitle;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    while (getline(input, line)) {
        istringstream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
        ss >> bedrooms;
        ss.ignore(); // Ignore the comma
        ss >> bathrooms;
        ss.ignore(); // Ignore the comma
        ss >> lotArea;
        ss.ignore(); // Ignore the comma
        ss >> price;
        ss.ignore(); // Ignore the comma
        ss >> isAvailable;

        if (propertyId == idToCheck) {
            input.close();
            return true; // ID exists in the database
        }
        // Skip the rest of the line
        input.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    input.close();
    return false; // ID doesn't exist in the database
}

string getPropertyId() const {return propertyId;}
string getPropertyType() const {return propertyType;}
```



```
string getPropertyAddress() const {return propertyAddress;}
string getLotTitle() const {return propertyLotTitle;}
int getBedrooms() const {return bedrooms;}
int getBathrooms() const {return bathrooms;}
double getLotArea() const {return lotArea;}
double getPrice() const {return price;}
bool isAvailable() const {return is_available;} ;
};

class UserProfile {
private:
    int id;
    string username;
    string password;
    double coins;
    bool is_logged_in;
    vector<RealEstateProperty> ownedProperties;

public:
    UserProfile(int user_id, string uname, string pwd)
        : id(user_id), username(uname), password(pwd), is_logged_in(false)
    {}

    void login(const string& entered_username, const string&
entered_password) {
        ifstream input("userDatabase.txt");

        if (!input.is_open()) {
            // Handle unable to open the file
            return;
        }

        string line;
        bool loginSuccess = false;

        while (getline(input, line)) {
            istream ss(line);
            ss >> id;
            ss.ignore(); // Ignore the comma
            getline(ss, username, ',');
            getline(ss, password, ',');
            ss >> coins;
            ss.ignore(); // Ignore the comma
            ss >> is_logged_in;

            if (username == entered_username && password ==
entered_password) {
                is_logged_in = true;
                loginSuccess = true;
                setUserData(id, username, password, coins);
            }
        }
        input.close();
        if (loginSuccess) {
            // Successful login actions
            Color::setTextColor(Color::BrightGreen);
            cout << "LoggedIn successfully." << endl; // If username and
password exist
            Color::setTextColor(Color::White);
            cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
```



```
countdown(1);
loginCountdown(3);
updateUserIsLoggedIn(loginSuccess);
system("cls");

    } else {
        // Handle failed login
    }
}

void buyProperty(int userId ,string propertyId) {
    while(true){
        string propertyType, propertyAddress,
propertyLotTitle ;

        int bedrooms, bathrooms;
        double price, lotArea;

        RealEstateProperty
property(propertyType, propertyAddress, propertyLotTitle, bedrooms,
bathrooms, lotArea,price);

        property.readPropertyById(propertyId);
        readUserProfileById(userId);
        // Check if property exists and is available
        if (propertyExists(propertyId)      &&
property.isPropertyAvailable(propertyId)) {
            double          propertyPrice      =
property.getPriceOfProperty(propertyId);
            ifstream input("userDatabase.txt");

            if (!input.is_open()) {
                // Handle unable to open the file
                return;
            }

            string line;
            bool loginSuccess = false;

            while (getline(input, line)) {
                istringstream ss(line);
                ss >> id;

                ss.ignore(); // Ignore the comma
                getline(ss, username, ',');
                getline(ss, password, ',');
                ss >> coins;
                ss.ignore(); // Ignore the comma
                ss >> is_logged_in;

                if (userId == id) {
                    id=id;

username=username;

password=password;

coins=coins;

is_logged_in=is_logged_in;

setUserData(id, username,
password, coins);
                }
            }
        }
    }
}
```



```
        }
        input.close();
    if (getCoins() >= propertyPrice) {
        ownedProperties.push_back(property);
        updatePropertyAvailability(propertyId, false);
        updateUserCoins(propertyPrice);
        savePropertyToFile(propertyId, property);
        Color::setTextColor(Color::BrightGreen);
        cout << "Property purchased successfully!" << endl;
    } else {
        Color::setTextColor(Color::BrightRed);
        cout << "Insufficient coins to purchase the property."
<< endl;
    }
} else if (propertyExists(propertyId) &&
!(property.isPropertyAvailable(propertyId))) {
    Color::setTextColor(Color::Red);
    cout << "Property not available." << endl;
} else {
    Color::setTextColor(Color::Red);
    cout << "Property does not exist." << endl;
}
break;
}
}

void setData(int new_id, const string& new_username,
const string& new_password, double new_coins) {
    id = new_id;
    username = new_username;
    password = new_password;
    coins = new_coins;
}

void readUserProfileById(int id) {
    ifstream inFile("userDatabase.txt");

    if (!inFile.is_open()) {
        // Handle unable to open the file
        return;
    }

    string line;
    while (getline(inFile, line)) {
        istringstream ss(line);
        int file_id;
        string file_username, file_password;
        double file_coins;
        bool islogin;

        ss >> file_id;
        getline(ss, file_username, ',');
        getline(ss, file_password, ',');
        ss >> file_coins;
        ss >> islogin;

        if (file_id == id) {
            id = file_id;
            username = file_username;
```



```
        password=file_password;
        coins=file_coins;
        is_logged_in=islogin;
    }
}

// Function to update user password
void updateUserPassword(const string& new_password) {
    password = new_password;
    saveUserToFile(); // Save the updated password to file
}

void updateUserIsLoggedIn(bool login){
    is_logged_in = login;
    saveUserToFile();
}

// Function to add REMS coins to the user's account
void addRemsCoins(double additional_coins) {
    coins += additional_coins;
    saveUserToFile(); // Save the updated coins to file
}

int userOption(int id, string username, string password){
    int option;
    while (true) {
        Color::setTextColor(Color::Yellow);
        cout << "Enter your choice: ";
        Color::setTextColor(Color::White);

        if (!(cin >> option) || option < 1 || option > 5)
        {
            cin.clear(); // Clear error flags

            cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Clear input buffer
            system("cls");
            Color::setTextColor(Color::Red);
            cout << "Invalid choice. ";
            Color::setTextColor(Color::White);
            cout << "Please
enter your choice between ";
            Color::setTextColor(Color::BrightCyan);
            cout << "[";
            Color::setTextColor(Color::BrightGreen);
            cout << "1";

            Color::setTextColor(Color::BrightCyan);
            cout << "] ";
            Color::setTextColor(Color::Yellow);
            cout << "to ";
            Color::setTextColor(Color::BrightCyan);
            cout << "[";
            Color::setTextColor(Color::BrightGreen);
            cout << "5";

            Color::setTextColor(Color::BrightCyan);
            cout << "] ";

            Color::setTextColor(Color::White);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout <<
"only.\n";

showProfile(getId(),getUsername(),getPassword(),getCoins()); // If
logged in successfully, show user profile
    // Display menu options
    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "1";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Show All

Availabe Property"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "2";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Search

Property //Coming soon"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "3";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Buy A

Property"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "4";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Edit

Account"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "5";

    Color::setTextColor(Color::BrightCyan);
```



```
        cout << "]" ";

        Color::setTextColor(Color::BrightGreen);

        cout <<

"Logout"<<endl;

        Color::setTextColor(Color::White);
        Color::setTextColor(Color::Yellow);
        // Display other menu options similarly
        Color::setTextColor(Color::White);
        continue; // Restart the loop
    }

    // Clear any additional characters in the input
    buffer

    cin.ignore(numeric_limits<streamsize>::max(), '\n');

        // Valid input, break the loop
        break;
    }
    return option;
}

int userAccountOption(int id, string username, string
password){
    int option;
    while (true) {
        Color::setTextColor(Color::Yellow);
        cout << "Enter your choice: ";
        Color::setTextColor(Color::White);

        if (!(cin >> option) || option < 1 || option > 3)
        {
            cin.clear(); // Clear error flags

            cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Clear input buffer
            system("cls");
            Color::setTextColor(Color::Red);
            cout << "Invalid choice. ";
            Color::setTextColor(Color::White);
            cout << "Please
enter your choice between ";
            Color::setTextColor(Color::BrightCyan);
            cout << "[";
            Color::setTextColor(Color::BrightGreen);
            cout << "1";

            Color::setTextColor(Color::BrightCyan);
            cout << "]" ";
            Color::setTextColor(Color::Yellow);
            cout << "to ";
            Color::setTextColor(Color::BrightCyan);
            cout << "[";
            Color::setTextColor(Color::BrightGreen);
            cout << "3";

            Color::setTextColor(Color::BrightCyan);
            cout << "]" ";

            Color::setTextColor(Color::White);
```




Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout <<
"only.\n";

showProfile(getId(), getUsername(), getPassword(), getCoins()); // If
logged in successfully, show user profile
    // Display menu options
    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "1";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Top Up

Rems Coins"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "2";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Change

Password"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "3";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout <<

"Exit"<<endl;

    Color::setTextColor(Color::White);
    Color::setTextColor(Color::Yellow);
    // Display other menu options similarly
    Color::setTextColor(Color::White);
    continue; // Restart the loop
}

// Clear any additional characters in the input
buffer

cin.ignore(numeric_limits<streamsize>::max(), '\n');

    // Valid input, break the loop
    break;
}
return option;
}

void showProfileWithNewCoins(int coins){
```



```
double newCoins;
Color::setTextColor(Color::Cyan);
cout<<"USER PROFILE"<<endl;
Color::setTextColor(Color::Yellow);
cout<<"ID: ";
Color::setTextColor(Color::BrightYellow);
cout<<getId()<<endl;
Color::setTextColor(Color::Yellow);
cout<<"USERNAME: ";
Color::setTextColor(Color::BrightYellow);
cout<<getUsername()<<endl;
Color::setTextColor(Color::Yellow);
cout<<"REMS COINS: ";
Color::setTextColor(Color::Red);
cout<<"PHP ";
Color::setTextColor(Color::BrightYellow);
newCoins = coins+getCoins();
cout<<fixed<<setprecision(2)<<newCoins<<endl;
}

void showProfile(int userId, string username, string password, double
coins){
    ifstream input("userDatabase.txt");

    if (!input.is_open()) {
        // Handle unable to open the file
        return;
    }

    string line;
    bool loginSuccess = false;

    while (getline(input, line)) {
        istringstream ss(line);
        ss >> id;
        ss.ignore(); // Ignore the comma
        getline(ss, username, ',');
        getline(ss, password, ',');
        ss >> coins;
        ss.ignore(); // Ignore the comma
        ss >> is_logged_in;

        if (username == username && password == password) {
            is_logged_in = true;
            loginSuccess = true;
            setUserData(id, username, password, coins);
        }
    }
    input.close();

    Color::setTextColor(Color::Cyan);
    cout<<"USER PROFILE"<<endl;
        Color::setTextColor(Color::BrightYellow);
        cout << "-----" << endl;
    Color::setTextColor(Color::Yellow);
    cout<<"ID: ";
    Color::setTextColor(Color::BrightYellow);
    cout<<userId<<endl;
    Color::setTextColor(Color::Yellow);
    cout<<"USERNAME: ";
```



```
        Color::setTextColor(Color::BrightYellow);
        cout<<username<<endl;
        Color::setTextColor(Color::Yellow);
        cout<<"REMS COINS: ";
        Color::setTextColor(Color::Red);
        cout<<"PHP ";
        Color::setTextColor(Color::BrightYellow);
        cout<<fixed<<setprecision(2)<<coins<<endl;
            Color::setTextColor(Color::BrightYellow);
            cout << "-----" << endl;

    }

void logout() {
    is_logged_in = false;
    Color::setTextColor(Color::BrightGreen);
    cout << "Logged out successfully." << endl;
    countdown(1);
    logoutCountdown(3);
    system("cls");
    updateUserIsLoggedIn(is_logged_in);
}

// Accessors (getters)
int getId() const { return id; }
string getPassword() const {return password;}
string getUsername() const { return username; }
double getCoins() const {return coins;}
bool isLoggedIn() const { return is_logged_in; }

private:
    // Function to check if a property exists
    bool propertyExists(string id) {
        ifstream propertyFile("globalProperty.txt");
        if (!propertyFile.is_open()) {
            // Handle unable to open the file
            return false;
        }

        string line;
        while (getline(propertyFile, line)) {
            istringstream ss(line);
            string propertyId;
            // Assuming propertyId is the first value in each line of the
property file
            getline(ss, propertyId, ',');

            if (propertyId == id) {
                propertyFile.close();
                return true;
            }
        }

        propertyFile.close();
        return false;
    }

    // Function to update property availability
    void updatePropertyAvailability(string id, bool availability) {
```



```
ifstream inFile("globalProperty.txt");
ofstream outFile("temp.txt");

if (!inFile.is_open() || !outFile.is_open()) {
    // Handle unable to open the file
    return;
}

string line;
while (getline(inFile, line)) {
    istringstream ss(line);
    string      propertyId,      propertyType,
propertyAddress, propertyLotTitle;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool is_available = true;
    bool isAvailable;

    // Assuming the format in propertyDatabase.txt is: ID,
Type, Address, Availability
    getline(ss, propertyId, ',');
    getline(ss, propertyType, ',');
    getline(ss, propertyAddress, ',');
    getline(ss, propertyLotTitle, ',');
    ss >> bedrooms;
    ss.ignore(); // Ignore the comma
    ss >> bathrooms;
    ss.ignore(); // Ignore the comma
    ss >> lotArea;
    ss.ignore(); // Ignore the comma
    ss >> price;
    ss.ignore(); // Ignore the comma
    ss >> isAvailable;

    if (propertyId == id && is_available == isAvailable) {
        // Update the line with new availability
        outFile << propertyId << "," << propertyType << "," <<
propertyAddress << "," << propertyLotTitle << ","
<< bedrooms << "," << bathrooms << "," << lotArea
<< "," << price << "," << availability << endl;
    } else {
        // Write the existing line as is
        outFile << line << endl;
    }
}

inFile.close();
outFile.close();

remove("globalProperty.txt");
rename("temp.txt", "globalProperty.txt");
}

void updateUserCoins(double updatedCoins) {
    coins -= updatedCoins;
    saveUserToFile(); // Save the updated coins to file
}

void saveUserToFile() {
```



```
ifstream inFile("userDatabase.txt");
ofstream outFile("temp.txt");

if (!inFile.is_open() || !outFile.is_open()) {
    // Handle unable to open the file
    return;
}

string line;
while (getline(inFile, line)) {
    istringstream ss(line);
    int file_id;
    string file_username, file_password;
    double file_coins;
    bool islogin;

    ss >> file_id;
    getline(ss, file_username, ',');
    getline(ss, file_password, ',');
    ss >> file_coins;
    ss >> islogin;

    if (file_id == id) {
        // Update the line with new data
        outFile << id << " " << username << " " << password << " " <<
coins << " " << is_logged_in << endl;
    } else {
        // Write the existing line as is
        outFile << line << endl;
    }
}

inFile.close();
outFile.close();

remove("userDatabase.txt");
rename("temp.txt", "userDatabase.txt");
}

void savePropertyToFile(string propId, const RealEstateProperty&
property) {
    ifstream input("globalProperty.txt");
    string propertyId, propertyType, propertyAddress,
propertyLotTitle;
    string line;
    int bedrooms, bathrooms;
    double price, lotArea;
    bool isAvailable;

    if (!input.is_open()) {
        // No database
    }

    while (getline(input, line)) {
        istringstream ss(line);
        getline(ss, propertyId, ',');
        getline(ss, propertyType, ',');
        getline(ss, propertyAddress, ',');
        getline(ss, propertyLotTitle, ',');
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
ss >> bedrooms;
ss.ignore(); // Ignore the comma
ss >> bathrooms;
ss.ignore(); // Ignore the comma
ss >> lotArea;
ss.ignore(); // Ignore the comma
ss >> price;
ss.ignore(); // Ignore the comma
ss >> isAvailable;

if(propId == propertyId){
    ofstream outFile("soldProperty.txt", ios::app);
    if (outFile.is_open()) {
        // Save property details to file
        outFile << id << "," << propertyId << "," <<
property.getPropertyType() << "," << property.getPropertyAddress() <<
"," << property.getLotTitle() << ","

        << property.getBedrooms() << "," << property.getBathrooms()
<< "," << property.getLotArea() << "," << property.getPrice() << "," <<
property.getPropertyType() << endl;
        outFile.close();
    } else {
        cout << "Unable to save property details."
<< endl;
    }
}

input.close();
};

class AdminProfile {
private:
    int id;
    string username;
    string password;
    bool is_logged_in;

public:
    AdminProfile(int user_id, string uname, string pwd)
        : id(user_id), username(uname), password(pwd), is_logged_in(false)
    {}

    void login(const string& entered_username, const string&
entered_password) {
        ifstream input("adminDatabase.txt");

        if (!input.is_open()) {
            // Handle unable to open the file
            return;
        }

        string line;

        while (getline(input, line)) {
            istringstream ss(line);
            ss >> id;
            ss.ignore(); // Ignore the comma
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
getline(ss, username, ',');
getline(ss, password, ',');

if (username == entered_username && password ==
entered_password) {
    is_logged_in = true;
    this->id = id; // Assign the id to the class member
    this->username = username;
    this->password = password;
    Color::setTextColor(Color::BrightGreen);
    cout << "LoggedIn successfully." << endl; // If username
and password exist
    Color::setTextColor(Color::White);
        countdown(1);
        loginCountdown(3);
        system("cls");
    Color::setTextColor(Color::White);
    return;
} else {
    is_logged_in = false;
}
}
input.close();
}

// Function to update user password
void updateUserPassword(const string& new_password) {
    password = new_password;
    saveUserToFile(); // Save the updated password to file
}

void saveUserToFile() {
    ifstream inFile("adminDatabase.txt");
    ofstream outFile("adminTemp.txt");

    if (!inFile.is_open() || !outFile.is_open()) {
        // Handle unable to open the file
        return;
    }

    string line;
    while (getline(inFile, line)) {
        istringstream ss(line);
        int file_id;
        string file_username, file_password;
        int login;

        ss >> file_id;
        ss.ignore();
        getline(ss, file_username, ',');
        getline(ss, file_password, ',');
        ss >> login;

        if (file_id == id) {
            // Update the line with new data
            outFile << id << "," << username << "," << password << "," <<
login << endl;
        } else {
            // Write the existing line as is
            outFile << line << endl;
        }
    }
}
```



```
}  
}  
  
inFile.close();  
outFile.close();  
  
remove("adminDatabase.txt");  
rename("adminTemp.txt", "adminDatabase.txt");  
}  
  
// Function to check if the generated USERID already exists in  
the database  
bool adminIdExistsInDatabase(int idToCheck) {  
    ifstream input("adminDatabase.txt");  
    int id;  
  
    while (input >> id) {  
        if (id == idToCheck) {  
            input.close();  
            return true; // ID exists in the database  
        }  
        // Skip the rest of the line  
        input.ignore(numeric_limits<streamsize>::max(), '\n');  
    }  
    input.close();  
    return false; // ID doesn't exist in the database  
}  
  
// Function to generate a unique 6-digit ID not present in the  
user database  
int generateAdminUniqueID() {  
    ifstream input("adminDatabase.txt");  
    int maxID = 0;  
    int id;  
  
    while (input >> id) {  
        if (id > maxID) {  
            maxID = id;  
        }  
        // Skip the rest of the line  
        input.ignore(numeric_limits<streamsize>::max(), '\n');  
    }  
    input.close();  
  
    int newID;  
    do {  
        newID = generateRandom6DigitNumber();  
    } while (adminIdExistsInDatabase(newID));  
  
    return newID;  
}  
  
// Function to generate a random 6-digit number  
int generateRandom6DigitNumber() {  
    srand(static_cast<unsigned int>(time(NULL))); // Seed the  
random number generator  
    return rand() % 900000 + 100000; // Generate a random  
number in the range 100000 to 999999  
}
```




```
bool checkAdminUsernameExists(const string&
enteredUsername) {
    ifstream input("adminDatabase.txt");
    string line;
    int userId;
    string username, password;
    double coins;

    if(!input.is_open()){
        // if no data base exist
        Color::setTextColor(Color::BrightRed);
        cout << "Database do not exist. ";

        Color::setTextColor(Color::BrightGreen);
        cout << "Default admin account have
been created successfully." << endl;
    }

    while (getline(input, line)) {
        istringstream ss(line);
        ss >> userId;
        ss.ignore(); // Ignore the comma
        getline(ss, username, ',');
        getline(ss, password, ',');
        if(username==enteredUsername){
            return true;
        }
    }

    input.close();
    return false;
}

void createDefaultAdminAccount(){
    string enteredUsername="admin",confirmPassword="admin";
    bool exist = false;

    do {

        exist = checkAdminUsernameExists(enteredUsername);

        if (exist) {
            break;
        }else {

            int newId =
generateAdminUniqueID(); // Generate a unique ID

            ofstream reg("adminDatabase.txt",
ios::app);

            reg << newId << " " <<
enteredUsername << " " << confirmPassword << " " << "0" << endl;
            reg.close();
        }
    } while (exist);
}

void showProfile(int adminID){
    Color::setTextColor(Color::Cyan);
```



```
cout<<"ADMIN PROFILE"<<endl;
Color::setTextColor(Color::BrightYellow);
cout << "-----" << endl;
Color::setTextColor(Color::Yellow);
cout<<"ID: ";
Color::setTextColor(Color::BrightYellow);
cout<<adminID<<endl;
Color::setTextColor(Color::Yellow);
cout<<"USERNAME: ";
Color::setTextColor(Color::BrightYellow);
cout<<getUsername()<<endl;
Color::setTextColor(Color::BrightYellow);
cout << "-----" << endl;
}

int adminAccountOption(int id, string username, string
password){
    int option;
    while (true) {
        Color::setTextColor(Color::Yellow);
        cout << "Enter your choice: ";
        Color::setTextColor(Color::White);

        if (!(cin >> option) || option < 1 || option > 3)
        {
            cin.clear(); // Clear error flags

            cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Clear input buffer
            system("cls");
            Color::setTextColor(Color::Red);
            cout << "Invalid choice. ";
            Color::setTextColor(Color::White);
            cout << "Please
enter your choice between ";
            Color::setTextColor(Color::BrightCyan);
            cout << "[";
            Color::setTextColor(Color::BrightGreen);
            cout << "1";

            Color::setTextColor(Color::BrightCyan);
            cout << "] ";
            Color::setTextColor(Color::Yellow);
            cout << "to ";
            Color::setTextColor(Color::BrightCyan);
            cout << "[";
            Color::setTextColor(Color::BrightGreen);
            cout << "3";

            Color::setTextColor(Color::BrightCyan);
            cout << "] ";

            Color::setTextColor(Color::White);
            cout <<
"only.\n";

            showProfile(getId()); // If logged in
successfully, show user profile
            // Display menu options
            Color::setTextColor(Color::BrightCyan);
            cout << "[";
            Color::setTextColor(Color::BrightRed);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout << "1";

Color::setTextColor(Color::BrightCyan);
cout << "] ";

Color::setTextColor(Color::BrightGreen);
cout << "Create
Admin Account"<<endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "2";

Color::setTextColor(Color::BrightCyan);
cout << "] ";

Color::setTextColor(Color::BrightGreen);
cout << "Change
Password"<<endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "3";

Color::setTextColor(Color::BrightCyan);
cout << "] ";

Color::setTextColor(Color::BrightGreen);
cout <<

"Exit"<<endl;

Color::setTextColor(Color::White);
Color::setTextColor(Color::Yellow);
// Display other menu options similarly
Color::setTextColor(Color::White);
continue; // Restart the loop
}

// Clear any additional characters in the input
buffer

cin.ignore(numeric_limits<streamsize>::max(), '\n');

// Valid input, break the loop
break;
}
return option;
}

int adminOption(int id, string username, string password){
int option;
while (true) {
Color::setTextColor(Color::Yellow);
cout << "Enter your choice: ";
Color::setTextColor(Color::White);

if (!(cin >> option) || option < 1 || option > 6)

{
cin.clear(); // Clear error flags
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cin.ignore(numeric_limits<streamsize>::max(), '\n'); // Clear input buffer
system("cls");
Color::setTextColor(Color::Red);
cout << "Invalid choice. ";
Color::setTextColor(Color::White);
cout << "Please
enter your choice between ";
Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightGreen);
cout << "1";

Color::setTextColor(Color::BrightCyan);
cout << "] ";
Color::setTextColor(Color::Yellow);
cout << "to ";
Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightGreen);
cout << "6";

Color::setTextColor(Color::BrightCyan);
cout << "] ";
Color::setTextColor(Color::White);
cout <<
"only.\n";
showProfile(getId()); // If logged in
successfully, show user profile
// Display menu options
Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "1";

Color::setTextColor(Color::BrightCyan);
cout << "] ";

Color::setTextColor(Color::BrightGreen);
cout << "Sell A
Property"<<endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "2";

Color::setTextColor(Color::BrightCyan);
cout << "] ";

Color::setTextColor(Color::BrightGreen);
cout << "Delete A
Property"<<endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "3";

Color::setTextColor(Color::BrightCyan);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout<< "Update
A Property"<<endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "4";

Color::setTextColor(Color::BrightCyan);
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout << "Edit
Accounts //Coming soon"<<endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "5";

Color::setTextColor(Color::BrightCyan);
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout << "Show All
Property"<<endl;

Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "6";

Color::setTextColor(Color::BrightCyan);
cout << "]" ";

Color::setTextColor(Color::BrightGreen);
cout <<

"Logout"<<endl;
    Color::setTextColor(Color::White);
    Color::setTextColor(Color::Yellow);
    // Display other menu options similarly
    Color::setTextColor(Color::White);
    continue; // Restart the loop
}

// Clear any additional characters in the input
buffer

cin.ignore(numeric_limits<streamsize>::max(), '\n');

    // Valid input, break the loop
    break;
}
return option;
}

void logout() {
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
is_logged_in = false;
Color::setTextColor(Color::BrightGreen);
cout << "Logged out successfully." << endl;
countdown(1);
logoutCountdown(3);
}

// Accessors (getters)
int getId() const { return id; }
string getUsername() const { return username; }
string getPassword() const { return password; }
bool isLoggedIn() const { return is_logged_in; }
};

// Function to generate a random 6-digit number
int generateRandom6DigitNumber() {
    srand(static_cast<unsigned int>(time(NULL))); // Seed the random
    number generator
    return rand() % 900000 + 100000; // Generate a random number in the
    range 100000 to 999999
}

// Function to check if the generated USERID already exists in the
    database
bool userIdExistsInDatabase(int idToCheck) {
    ifstream input("userDatabase.txt");
    int id;

    while (input >> id) {
        if (id == idToCheck) {
            input.close();
            return true; // ID exists in the database
        }
        // Skip the rest of the line
        input.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    input.close();
    return false; // ID doesn't exist in the database
}

// Function to generate a unique 6-digit ID not present in the user
    database
int generateUserUniqueID() {
    ifstream input("userDatabase.txt");
    int maxID = 0;
    int id;

    while (input >> id) {
        if (id > maxID) {
            maxID = id;
        }
        // Skip the rest of the line
        input.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    input.close();

    int newID;
    do {
        newID = generateRandom6DigitNumber();
    } while (userIdExistsInDatabase(newID));
}
```



```
return newID;
}

bool checkUserUsernameExists(const string& enteredUsername) {
    ifstream input("userDatabase.txt");
    string line;
    int userId;
    string username, password;
    double coins;

    if(!input.is_open()){
        // if no database exist
    }

    while (getline(input, line)) {
        istringstream ss(line);
        ss >> userId;
        ss.ignore(); // Ignore the comma
        getline(ss, username, ',');
        getline(ss, password, ',');
        ss >> coins;
        if(username==enteredUsername){
            return true;
        }
    }

    input.close();
    return false;
}

void registerUser() {
    string enteredUsername, enteredPassword, confirmPassword;
    double usercoin = 0;
    bool exist = false;
    bool loggin = false;

    do {
        Color::setTextColor(Color::BrightCyan);
        cout << "USER REGISTRATION FORM" << endl;

        Color::setTextColor(Color::Yellow);
        cout << "Enter ";
        Color::setTextColor(Color::BrightGreen);
        cout << "Username: ";
        Color::setTextColor(Color::White);
        cin >> enteredUsername;

        Color::setTextColor(Color::Yellow);
        cout << "Enter ";
        Color::setTextColor(Color::BrightGreen);
        cout << "Password";
        Color::setTextColor(Color::Yellow);
        cout << ": ";
        Color::setTextColor(Color::White);
        enteredPassword = maskedInput();

        Color::setTextColor(Color::Yellow);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout << "Confirm ";
Color::setTextColor(Color::BrightGreen);
cout << "Password";
Color::setTextColor(Color::Yellow);
cout << ": ";
Color::setTextColor(Color::White);
confirmPassword = maskedInput();

exist = checkUserUsernameExists(enteredUsername);

if (confirmPassword != enteredPassword) {
    system("cls");
    Color::setTextColor(Color::Red);
    cout << "Passwords do not match.\n";
} else if (exist) {
    system("cls");
    Color::setTextColor(Color::Red);
    cout << "Username is already taken." << endl;
}
} while (confirmPassword != enteredPassword || exist);

int newId = generateUserUniqueID(); // Generate a unique ID

ofstream reg("userDatabase.txt", ios::app);
reg << newId << ", " << enteredUsername << ", " << confirmPassword <<
", " << usercoin << ", " << loggin << endl;
reg.close();

system("cls");
Color::setTextColor(Color::BrightGreen);
cout << "User registered successfully!" << endl;
Color::setTextColor(Color::White);
countdown(1);
logoutCountdown(3);
main();
}

void userLogin() {
    string username="", password="";
    int id = 0; // Initialize ID here
    int counter = 0; // Counter to track login attempts

    Color::setTextColor(Color::Cyan);
    cout << "USER LOGIN PAGE" << endl;
    Color::setTextColor(Color::Yellow);
    cout << "Username: ";
    Color::setTextColor(Color::White);
    cin >> username;
    Color::setTextColor(Color::Yellow);
    cout << "Password: ";
    Color::setTextColor(Color::White);
    password = maskedInput();

    UserProfile user(id, username, password);
    system("cls");

    while (counter < 3) {
        user.login(username, password);
        if (user.isLoggedIn()) {
```




```
bool exitRequested = false;
    while(!exitRequested){

user.showProfile(user.getId(),user.getUsername(),user.getPassword(),user.getCoins()); // If logged in successfully, break out of the loop
    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "1";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Show All Availabe
Property"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "2";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Search Property
//Coming soon"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "3";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Buy A
Property"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "4";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";

    Color::setTextColor(Color::BrightGreen);
    cout << "Edit
Account"<<endl;

    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "5";

    Color::setTextColor(Color::BrightCyan);
    cout << "]" ";
```



```

        Color::setTextColor(Color::BrightGreen);
        cout << "Logout"<<endl;
        Color::setTextColor(Color::White);
        Color::setTextColor(Color::Yellow);
        int option = user.userOption(user.getId(),
user.getUsername(), user.getPassword());

        switch(option){
            case 1:
                {
                    system("cls");

                    string propertyType, propertyAddress, propertyLotTitle ;

                    int bedrooms, bathrooms;

                    double price, lotArea;

                    RealEstateProperty property(propertyType, propertyAddress,
propertyLotTitle, bedrooms, bathrooms, lotArea,price);

                    Color::setTextColor(Color::BrightCyan);
                    cout <<
"PROPERTY'S MARKET PLACE"<<endl;

                    Color::setTextColor(Color::BrightYellow);
                    cout << "-----
-----" << endl;

                    property.showAllAvailableProperty();
                }
                break;
            case 2:
                {
                    system("cls");
                    cout<<"Search Property is ";

                    Color::setTextColor(Color::Red);

                    cout << "**";

                    Color::setTextColor(Color::BrightGreen);

                    cout << "Coming Soon";

                    Color::setTextColor(Color::Yellow);

                    cout << "!";

                    Color::setTextColor(Color::Red);

                    cout << "**"<<endl;

                    Color::setTextColor(Color::White);
                }
                break;
            case 3:
    
```



```
{  
  
string propertyId, propertyType, propertyAddress, propertyLotTitle;  
                                                                    int  
bedrooms, bathrooms;  
  
double price, lotArea;  
  
RealEstateProperty    property(propertyType,    propertyAddress,  
propertyLotTitle, bedrooms, bathrooms, lotArea, price);  
  
    system("cls");  
                                string propertyIdToBuy;  
  
Color::setTextColor(Color::BrightCyan);  
  
cout << "BUY A PROPERTY" << endl;  
  
user.showProfile(user.getId(), user.getUsername(), user.getPassword(), user.getCoins());  
  
Color::setTextColor(Color::Yellow);  
  
cout << "Enter Property ID: ";  
  
Color::setTextColor(Color::White);  
  
while(! (cin >> propertyIdToBuy)) {  
  
    Color::setTextColor(Color::Yellow);  
  
    cout << "Enter Property ID: ";  
  
    Color::setTextColor(Color::White);  
  
    }  
  
    bool isItExist = property.propertyIdExists(propertyIdToBuy);  
  
    if(isItExist){  
  
        bool                isItAvailable                =  
property.isPropertyAvailable(propertyIdToBuy);  
  
        if(isItAvailable){  
  
            char ans;  
  
            system("cls");  
  
            while(true){
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::BrightCyan);

        cout << "BUY A PROPERTY" << endl;

user.showProfile(user.getId(),user.getUsername(),user.getPassword(),user.getCoins());

        property.showPropertyById(propertyIdToBuy);

        Color::setTextColor(Color::BrightYellow);

        cout << "You will buy this property." <<
endl;

        Color::setTextColor(Color::Yellow);

        cout << "Do you want to continue?
(Y/N) : ";

        Color::setTextColor(Color::White);

        cin>>ans;

        if(ans=='Y' || ans=='y'){

                system("cls");

        user.buyProperty(user.getId(),propertyIdToBuy);

                countdown(1);

        loginCountdown(3);

                break;

        }else if(ans=='N' || ans=='n'){

                system("cls");

        Color::setTextColor(Color::BrightGreen);

        cout << "Buying
property have been canceled.";

                countdown(1);

        loginCountdown(3);

                break;
```



```
        }else{

            system("cls");

            Color::setTextColor(Color::Red);

            cout << "Invalid";

            Color::setTextColor(Color::White);

            cout << " input.

Please enter a ";

            Color::setTextColor(Color::BrightGreen);

            cout << "valid ";

            Color::setTextColor(Color::Yellow);

            cout << "choice."

        << endl;

        Color::setTextColor(Color::BrightCyan);

        continue;

    }

}

}else{

    system("cls");

    Color::setTextColor(Color::Red);

    cout

    << "Property not available." << endl;

    }

}else{

    system("cls");

    Color::setTextColor(Color::Red);

    cout

    << "Property does not exist." << endl;

    }

}

        break;

    case 4:

    {

        bool

        exitRequested = false;
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
while(!exitRequested){  
  
    system("cls");  
  
        Color::setTextColor(Color::BrightCyan);  
  
    cout << "EDIT MY ACCOUNT"<<endl;  
  
        Color::setTextColor(Color::BrightYellow);  
  
    user.showProfile(user.getId(),user.getUsername(),user.getPassword(),user.getCoins()); // If logged in successfully, break out of the loop  
  
    Color::setTextColor(Color::BrightCyan);  
  
    cout << "[";  
  
    Color::setTextColor(Color::BrightRed);  
  
        cout << "1";  
  
        Color::setTextColor(Color::BrightCyan);  
  
        cout << "]" ";  
  
    Color::setTextColor(Color::BrightGreen);  
  
        cout << "Top Up Rems Coins"<<endl;  
  
        Color::setTextColor(Color::BrightCyan);  
  
    cout << "[";  
  
    Color::setTextColor(Color::BrightRed);  
  
        cout << "2";  
  
        Color::setTextColor(Color::BrightCyan);  
  
        cout << "]" ";  
  
    Color::setTextColor(Color::BrightGreen);  
  
        cout << "Change Password"<<endl;  
  
        Color::setTextColor(Color::BrightCyan);  
  
    cout << "[";  
  
    Color::setTextColor(Color::BrightRed);  
  
        cout << "3";  
  
        Color::setTextColor(Color::BrightCyan);  
  
        cout << "]" ";
```



```
Color::setTextColor(Color::BrightGreen);

cout << "Exit" << endl;

Color::setTextColor(Color::White);

int option = user.userAccountOption(user.getId(), user.getUsername(),
user.getPassword());

switch(option){

    case 1:

        {

            double coins;

            system("cls");

            Color::setTextColor(Color::BrightCyan);

            cout

<< "TOP UP REMS COINS" << endl;

            do {

                user.showProfile(user.getId(),user.getUsername(),user.getPa
ssword(),user.getCoins());

                Color::setTextColor(Color::Yellow);

                cout << "Enter coins amount ";

                Color::setTextColor(Color::BrightCyan);

                cout << "(";

                Color::setTextColor(Color::BrightGreen);

                cout << "minimum 100";

                Color::setTextColor(Color::BrightCyan);

                cout << ");";
```



```
Color::setTextColor(Color::Yellow);

cout << ": ";

Color::setTextColor(Color::White);

cin

>> coins;

if

(coins < 100 || coins == 0) {

system("cls");

Color::setTextColor(Color::Red);

cout << "Invalid amount. Please enter an amount of at least 100." << endl;

Color::setTextColor(Color::BrightCyan);

cout << "TOP UP REMS COINS" << endl;

}

else if (coins != static_cast<int>(coins)) {

system("cls");

Color::setTextColor(Color::Red);

cout << "Invalid amount. Please enter a valid amount. " << endl;

Color::setTextColor(Color::BrightCyan);

cout << "TOP UP REMS COINS" << endl;

}else if(coins>100000000){

system("cls");

Color::setTextColor(Color::Red);
```




Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cout << "Invalid amount. Please enter an amount not exceeding " ;

Color::setTextColor(Color::BrightGreen);

cout << "PHP " ;

Color::setTextColor(Color::BrightCyan);

        cout << fixed << setprecision(2) << 100000000 <<
endl;

Color::setTextColor(Color::BrightCyan);

cout << "TOP UP REMS COINS" << endl;

    }

while (coins < 100 || coins == 0 || coins !=
static_cast<int>(coins) || coins>100000000);

do{

    system("cls");

    Color::setTextColor(Color::BrightCyan);

    cout << "TOP UP REMS COINS" << endl;

    user.showProfileWithNewCoins(coins);

    char ans;

    Color::setTextColor(Color::Yellow);

    cout << "Do you want to continue? (Y/N): ";

    Color::setTextColor(Color::White);
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
cin >> ans;
```

```
if(ans == 'Y' || ans == 'y'){
```

```
    user.addRemsCoins(coins);
```

```
    system("cls");
```

```
    Color::setTextColor(Color::BrightGreen);
```

```
    cout << "Top up has been successful.";
```

```
    countdown(1);
```

```
    loginCountdown(3);
```

```
    break;
```

```
}else if(ans == 'N' || ans == 'n'){
```

```
    system("cls");
```

```
    Color::setTextColor(Color::BrightGreen);
```

```
    cout << "Top up has been aborted.";
```

```
    countdown(1);
```

```
    loginCountdown(3);
```

```
    break;
```

```
}else{
```

```
    system("cls");
```

```
Color::setTextColor(Color::Red);
```



```
cout <<
"Invalid";

Color::setTextColor(Color::White);

cout << " input. Please enter a ";

Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "choice." << endl;

Color::setTextColor(Color::BrightCyan);

continue;

}

}while(true);

}

break;

case 2:

{

string

newPassword, password, confirmPassword;

system("cls");

Color::setTextColor(Color::BrightCyan);

cout

<< "CHANGE PASSWORD" << endl;

do{
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
user.showProfile(user.getId(),user.getUsername(),user.getPa  
ssword(),user.getCoins());
```

```
Color::setTextColor(Color::Yellow);
```

```
cout << "Enter current password: ";
```

```
Color::setTextColor(Color::White);
```

```
password = maskedInput();
```

```
if(user.getPassword()!=password){
```

```
system("cls");
```

```
Color::setTextColor(Color::Red);
```

```
cout << "Incorrect Password." << endl;
```

```
Color::setTextColor(Color::BrightCyan);
```

```
cout << "CHANGE PASSWORD" << endl;
```

```
}
```

```
}while(user.getPassword()!=password);
```

```
system("cls");
```

```
do{
```

```
Color::setTextColor(Color::BrightCyan);
```

```
cout << "CHANGE PASSWORD" << endl;
```

```
user.showProfile(user.getId(),user.getUsername(),user.getPa  
ssword(),user.getCoins());
```

```
Color::setTextColor(Color::Yellow);
```



```
cout << "Enter new password: ";

Color::setTextColor(Color::White);

newPassword = maskedInput();

Color::setTextColor(Color::Yellow);

cout << "Confirm password: ";

Color::setTextColor(Color::White);

confirmPassword = maskedInput();

if(newPassword!=confirmPassword){

system("cls");

Color::setTextColor(Color::Red);

cout << "Password do not match." << endl;

    } else if(confirmPassword==user.getPassword()){

system("cls");

Color::setTextColor(Color::Red);

cout << "Cannot change for the same password." << endl;

    }

}

while(newPassword!=confirmPassword | confirmPassword==user.getPassword());
```



```
do{

system("cls");

Color::setTextColor(Color::BrightCyan);

cout << "CHANGE PASSWORD" << endl;

user.showProfile(user.getId(),user.getUsername(),user.getPassword(),user.getCoin());

char ans;

Color::setTextColor(Color::BrightYellow);

cout << "Your password will be changed." << endl;

Color::setTextColor(Color::Yellow);

cout << "Do you want to continue? (Y/N): ";

Color::setTextColor(Color::White);

cin >> ans;

if(ans == 'Y' || ans == 'y'){

    user.updateUserPassword(confirmPassword);

    system("cls");

    Color::setTextColor(Color::BrightGreen);

    cout << "Successfully changed the password.";

    countdown(1);
```



```
loginCountdown(3);

break;

}else if(ans == 'N' || ans == 'n'){

system("cls");

Color::setTextColor(Color::BrightGreen);

cout << "Abort to changed the password.";

countdown(1);

loginCountdown(3);

break;

}else{

system("cls");

Color::setTextColor(Color::Red);

cout <<

"Invalid";

Color::setTextColor(Color::White);

cout << " input. Please enter a ";

Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "choice." << endl;

Color::setTextColor(Color::BrightCyan);
```



continue;

}

}while(true);

}

break;

case 3:

{

system("cls");

exitRequested = true;

}

break;

default:

break;

}

}

}

break;

case 5:

exitRequested = true;

system("cls");

user.logout();

main();

break;

default:

break;

}

}

break;

} else {

counter++;

Color::setTextColor(Color::Red);

cout << "Incorrect password. ";

Color::setTextColor(Color::BrightGreen);

cout << "Login ";

Color::setTextColor(Color::Red);

cout << "failed. ";

Color::setTextColor(Color::White);

cout << "Please try again." << endl;



```
Color::setTextColor(Color::BrightGreen);
cout << "Login ";
Color::setTextColor(Color::Red);
        cout << "attempt ";
        Color::setTextColor(Color::BrightCyan);
        cout << "[" ;

        Color::setTextColor(Color::BrightGreen);
        cout << counter ;
        Color::setTextColor(Color::BrightCyan);
        cout << "]" << endl;

    if (counter < 3) {
        // Allow re-entry of credentials for remaining attempts
        Color::setTextColor(Color::Yellow);
        cout << "Username: ";
        Color::setTextColor(Color::White);
        cin >> username;
        Color::setTextColor(Color::Yellow);
        cout << "Password: ";
        Color::setTextColor(Color::White);
        password = maskedInput();
        system("cls");
    }
}

if (counter == 3 && !user.isLoggedIn()) {
    system("cls");
    Color::setTextColor(Color::Red);
    cout << "Reached maximum ";
    Color::setTextColor(Color::BrightGreen);
        cout << "login ";
        Color::setTextColor(Color::Red);
        cout << "attempts.";
    Color::setTextColor(Color::White);
    cout<<"Please create an account to ";
    Color::setTextColor(Color::BrightGreen);
    cout<<"login";
    Color::setTextColor(Color::White);
    cout<<". "<<endl;
    main(); // Redirect to main menu
}

}

// Function to check if the generated USERID already exists in the
database
bool adminIdExistsInDatabase(int idToCheck) {
    ifstream input("adminDatabase.txt");
    int id;

    while (input >> id) {
        if (id == idToCheck) {
            input.close();
            return true; // ID exists in the database
        }
        // Skip the rest of the line
        input.ignore(numeric_limits<streamsize>::max(), '\n');
    }
}
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
input.close();
return false; // ID doesn't exist in the database
}

// Function to generate a unique 6-digit ID not present in the user
database
int generateAdminUniqueID() {
    ifstream input("adminDatabase.txt");
    int maxID = 0;
    int id;

    while (input >> id) {
        if (id > maxID) {
            maxID = id;
        }
        // Skip the rest of the line
        input.ignore(numeric_limits<streamsize>::max(), '\n');
    }
    input.close();

    int newID;
    do {
        newID = generateRandom6DigitNumber();
    } while (adminIdExistsInDatabase(newID));

    return newID;
}

bool checkAdminUsernameExists(const string& enteredUsername) {
    ifstream input("adminDatabase.txt");
    string line;
    int userId;
    string username, password;
    double coins;

    if(!input.is_open()){
        system("cls");
        Color::setTextColor(Color::BrightRed);
        cout<<"Cannot connect to database...";
    }

    while (getline(input, line)) {
        istringstream ss(line);
        ss >> userId;
        ss.ignore(); // Ignore the comma
        getline(ss, username, ',');
        getline(ss, password, ',');
        if(username==enteredUsername){
            return true;
        }
    }

    input.close();
    return false;
}

void registerAdmin() {
    string enteredUsername, enteredPassword, confirmPassword;
    bool exist = false;
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
do {
    Color::setTextColor(Color::BrightCyan);
    cout << "ADMIN REGISTRATION FORM" << endl;
    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Username";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);
    cin >> enteredUsername;

    Color::setTextColor(Color::Yellow);
    cout << "Enter ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Password";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);
    enteredPassword = maskedInput();

    Color::setTextColor(Color::Yellow);
    cout << "Confirm ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Password";
    Color::setTextColor(Color::Yellow);
    cout << ": ";
    Color::setTextColor(Color::White);
    confirmPassword = maskedInput();

    exist = checkAdminUsernameExists(enteredUsername);

    if (confirmPassword != enteredPassword) {
        system("cls");
        Color::setTextColor(Color::Red);
        cout << "Passwords do not match.\n";
    } else if (exist) {
        system("cls");
        Color::setTextColor(Color::Red);
        cout << "Username is already taken." << endl;
    }
} while (confirmPassword != enteredPassword || exist);

int newId = generateAdminUniqueID(); // Generate a unique ID

ofstream reg("adminDatabase.txt", ios::app);
reg << newId << ", " << enteredUsername << ", " << confirmPassword <<
", " << "0" << endl;
reg.close();

system("cls");
Color::setTextColor(Color::BrightGreen);
cout << "Admin account registered successfully!" << endl;
Color::setTextColor(Color::White);
}

void adminLogin() {
    string username, password;
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
int id = 0; // Initialize ID here
int counter = 0; // Counter to track login attempts

Color::setTextColor(Color::Cyan);
cout << "ADMIN LOGIN PAGE" << endl;
Color::setTextColor(Color::Yellow);
cout << "Username: ";
Color::setTextColor(Color::White);
cin >> username;
Color::setTextColor(Color::Yellow);
cout << "Password: ";
Color::setTextColor(Color::White);
password = maskedInput();

AdminProfile admin(id, username, password);
system("cls");

while (counter < 3) {
    admin.login(username, password);
    if (admin.isLoggedIn()) {
        bool exitRequested = false;
        while(!exitRequested){
            admin.showProfile(admin.getId()); // If logged in successfully,
break out of the loop
            Color::setTextColor(Color::BrightCyan);
            cout << "[";
            Color::setTextColor(Color::BrightRed);
            cout << "1";
            Color::setTextColor(Color::BrightCyan);
            cout << "] ";

            Color::setTextColor(Color::BrightGreen);
            cout << "Sell A Property"<<endl;
            Color::setTextColor(Color::BrightCyan);

            cout << "[";
            Color::setTextColor(Color::BrightRed);
            cout << "2";
            Color::setTextColor(Color::BrightCyan);
            cout << "] ";

            Color::setTextColor(Color::BrightGreen);
            cout << "Delete A Property"<<endl;
            Color::setTextColor(Color::BrightCyan);

            cout << "[";
            Color::setTextColor(Color::BrightRed);
            cout << "3";
            Color::setTextColor(Color::BrightCyan);
            cout << "] ";

            Color::setTextColor(Color::BrightGreen);
            cout<< "Update A Property"<<endl;
            Color::setTextColor(Color::BrightCyan);

            cout << "[";
            Color::setTextColor(Color::BrightRed);
            cout << "4";
            Color::setTextColor(Color::BrightCyan);
            cout << "] ";

            Color::setTextColor(Color::BrightGreen);
            cout << "Edit Accounts"<<endl;
```



```
        Color::setTextColor(Color::BrightCyan);

cout << "[";
Color::setTextColor(Color::BrightRed);
        cout << "5";
        Color::setTextColor(Color::BrightCyan);
        cout << "]" ";

        Color::setTextColor(Color::BrightGreen);
        cout << "Show All Property"<<endl;
        Color::setTextColor(Color::BrightCyan);

cout << "[";
Color::setTextColor(Color::BrightRed);
        cout << "6";
        Color::setTextColor(Color::BrightCyan);
        cout << "]" ";

        Color::setTextColor(Color::BrightGreen);
        cout << "Logout"<<endl;
Color::setTextColor(Color::White);
Color::setTextColor(Color::Yellow);
int    option    =    admin.adminOption(admin.getId(),
admin.getUsername(), admin.getPassword());

        switch(option){
            case 1:

                                {

                system("cls");

                string propertyType, propertyAddress, propertyLotTitle;

                int bedrooms, bathrooms;

                double price, lotArea;

                RealEstateProperty property(propertyType, propertyAddress,
propertyLotTitle, bedrooms, bathrooms, lotArea, price);

                property.createProperty();

                countdown(1);

loginCountdown(3);

                                }

                break;

            case 2:

                {

                system("cls");

                string propertyType, propertyAddress, propertyLotTitle;

                int bedrooms, bathrooms;

                double price, lotArea;
```



```
RealEstateProperty property(propertyType, propertyAddress,  
propertyLotTitle, bedrooms, bathrooms, lotArea, price);
```

```
property.deleteProperty();
```

```
countdown(1);
```

```
loginCountdown(3);
```

```
}
```

```
break;
```

```
case 3:
```

```
{
```

```
string propertyType, propertyAddress, propertyLotTitle;
```

```
int bedrooms, bathrooms;
```

```
double price, lotArea;
```

```
RealEstateProperty property(propertyType, propertyAddress,  
propertyLotTitle, bedrooms, bathrooms, lotArea, price);
```

```
property.updateProperty();
```

```
countdown(1);
```

```
loginCountdown(3);
```

```
}
```

```
break;
```

```
case 4:
```

```
{
```

```
bool
```

```
exitRequested = false;
```

```
while(!exitRequested){
```

```
system("cls");
```

```
Color::setTextColor(Color::BrightCyan);
```

```
cout << "EDIT MY ACCOUNT"<<endl;
```

```
Color::setTextColor(Color::BrightYellow);
```

```
admin.showProfile(admin.getId()); // If logged in successfully, break out  
of the loop
```

```
Color::setTextColor(Color::BrightCyan);
```

```
cout << "[";
```

```
Color::setTextColor(Color::BrightRed);
```

```
cout << "1";
```

```
Color::setTextColor(Color::BrightCyan);
```



```
        cout << " ] ";

        Color::setTextColor(Color::BrightGreen);

        cout << "Create Admin Account"<<endl;

        Color::setTextColor(Color::BrightCyan);

    cout << "[";

    Color::setTextColor(Color::BrightRed);

        cout << "2";

        Color::setTextColor(Color::BrightCyan);

        cout << " ] ";

        Color::setTextColor(Color::BrightGreen);

        cout << "Change Password"<<endl;

        Color::setTextColor(Color::BrightCyan);

    cout << "[";

    Color::setTextColor(Color::BrightRed);

        cout << "3";

        Color::setTextColor(Color::BrightCyan);

        cout << " ] ";

        Color::setTextColor(Color::BrightGreen);

        cout << "Exit"<<endl;

    Color::setTextColor(Color::White);

    int option =
    admin.adminAccountOption(admin.getId(),admin.getUsername(),admin.
    getPassword());

    switch(option){

        case 1:

            {

                system("cls");

                registerAdmin();
```



```
countdown(1);

                                loginCountdown(3);
                                }

                                break;

                                case 2:
                                {
                                string
                                newPassword, password, confirmPassword;

                                system("cls");

                                Color::setTextColor(Color::BrightCyan);

                                cout

                                << "CHANGE PASSWORD" << endl;

                                do{

                                admin.showProfile(admin.getId());

                                Color::setTextColor(Color::Yellow);

                                cout << "Enter current password: ";

                                Color::setTextColor(Color::White);

                                password = maskedInput();

                                if(admin.getPassword()!=password){

                                system("cls");

                                Color::setTextColor(Color::Red);

                                cout << "Incorrect Password." << endl;

                                Color::setTextColor(Color::BrightCyan);
```




```
        cout << "CHANGE PASSWORD" << endl;

    }

}while(admin.getPassword()!=password);

system("cls");

do{

    Color::setTextColor(Color::BrightCyan);

    cout << "CHANGE PASSWORD" << endl;

    admin.showProfile(admin.getId());

    Color::setTextColor(Color::Yellow);

    cout << "Enter new password: ";

    Color::setTextColor(Color::White);

    newPassword = maskedInput();

    Color::setTextColor(Color::Yellow);

    cout << "Confirm password: ";

    Color::setTextColor(Color::White);

    confirmPassword = maskedInput();

    if(newPassword!=confirmPassword){

        system("cls");
```



```
Color::setTextColor(Color::Red);

cout << "Password do not match." << endl;

    } else if(confirmPassword==admin.getPassword()){

        system("cls");

Color::setTextColor(Color::Red);

cout << "Cannot change for the same password." << endl;

    }

    }

while(newPassword!=confirmPassword || confirmPassword==admin.get
Password());

do{

    system("cls");

    Color::setTextColor(Color::BrightCyan);

    cout << "CHANGE PASSWORD" << endl;

    admin.showProfile(admin.getId());

    char ans;

    Color::setTextColor(Color::BrightYellow);

    cout << "Your password will be changed." << endl;

    Color::setTextColor(Color::Yellow);

    cout << "Do you want to continue? (Y/N): ";
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::White);
```

```
cin >> ans;
```

```
if(ans == 'Y' || ans == 'y'){
```

```
    admin.updateUserPassword(confirmPassword);
```

```
    system("cls");
```

```
    Color::setTextColor(Color::BrightGreen);
```

```
    cout << "Successfully changed the password.";
```

```
    countdown(1);
```

```
    loginCountdown(3);
```

```
    break;
```

```
}else if(ans == 'N' || ans == 'n'){
```

```
    system("cls");
```

```
    Color::setTextColor(Color::BrightGreen);
```

```
    cout << "Abort to changed the password.";
```

```
    countdown(1);
```

```
    loginCountdown(3);
```

```
    break;
```

```
}else{
```

```
    system("cls");
```



Republic of the Philippines
MARIKINA POLYTECHNIC COLLEGE
Sta. Elena, Marikina City - 1800
INSTITUTE OF ENGINEERING
Bachelor of Science in Electronics
Engineering



```
Color::setTextColor(Color::Red);

cout <<

"Invalid";

Color::setTextColor(Color::White);

cout << " input. Please enter a ";

Color::setTextColor(Color::BrightGreen);

cout << "valid ";

Color::setTextColor(Color::Yellow);

cout << "choice." << endl;

Color::setTextColor(Color::BrightCyan);

continue;

}

}while(true);

}

break;

case 3:

{

system("cls");

exitRequested = true;

}

break;

default:

break;
```



```
        }
    }
    }

    case 5:
    {
        system("cls");

        string propertyType, propertyAddress, propertyLotTitle;

        int bedrooms, bathrooms;

        double price, lotArea;

        RealEstateProperty property(propertyType, propertyAddress,
        propertyLotTitle, bedrooms, bathrooms, lotArea, price);

        Color::setTextColor(Color::Cyan);

        cout <<

        "PROPERTY'S MARKET PLACE"<<endl;

        Color::setTextColor(Color::BrightYellow);

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

        property.showAllProperty();

        cout << endl;

    }

    case 6:
    {
        exitRequested = true;
        system("cls");
        admin.logout();
        main();
        break;
        default:
        {
            break;
        }
    }

    break;
} else {
    counter++;
    Color::setTextColor(Color::Red);
    cout << "Incorrect password. ";
    Color::setTextColor(Color::BrightGreen);
    cout << "Login ";
    Color::setTextColor(Color::Red);
    cout << "failed. ";

    Color::setTextColor(Color::White);
    cout << "Please try again." << endl;
    Color::setTextColor(Color::BrightGreen);
    cout << "Login ";
    Color::setTextColor(Color::Red);
    cout << "attempt ";
    Color::setTextColor(Color::BrightCyan);
    cout << "[";
```



```
        Color::setTextColor(Color::BrightGreen);
        cout << counter;
        Color::setTextColor(Color::BrightCyan);
        cout << "]" << endl;

    if (counter < 3) {
        // Allow re-entry of credentials for remaining attempts
        Color::setTextColor(Color::Yellow);
        cout << "Username: ";
        Color::setTextColor(Color::White);
        cin >> username;
        Color::setTextColor(Color::Yellow);
        cout << "Password: ";
        Color::setTextColor(Color::White);
        password = maskedInput();

        AdminProfile    admin(id,
username, password);
        system("cls");
    }
}

if (counter == 3 && !admin.isLoggedIn()) {
    system("cls");
    Color::setTextColor(Color::Red);
    cout << "Reached maximum ";
    Color::setTextColor(Color::BrightGreen);
    cout << "login ";
    Color::setTextColor(Color::Red);
    cout << "attempts.";

    Color::setTextColor(Color::White);
    cout<<"Please create an account to ";
    Color::setTextColor(Color::BrightGreen);
    cout<<"login";
    Color::setTextColor(Color::White);
    cout<<". "<<endl;
    main(); // Redirect to main menu
}

}

int main(){

    int choice;

    Color::setTextColor(Color::Red);
    cout << "REAL ";
    Color::setTextColor(Color::Yellow);
    cout << "ESTATE ";
    Color::setTextColor(Color::Green);
    cout << "MANAGEMENT ";
    Color::setTextColor(Color::Blue);
    cout << "SYSTEM" << endl;
    Color::setTextColor(Color::BrightCyan);
    cout << "[";
    Color::setTextColor(Color::BrightRed);
    cout << "1";
    Color::setTextColor(Color::BrightCyan);
    cout << "]" << endl;
    Color::setTextColor(Color::BrightGreen);
```



```
cout << "Login As Admin" << endl;
Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "2";
Color::setTextColor(Color::BrightCyan);
cout << "]" << endl;
Color::setTextColor(Color::BrightGreen);
cout << "Login As User" << endl;
Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "3";
Color::setTextColor(Color::BrightCyan);
cout << "]" << endl;
Color::setTextColor(Color::BrightGreen);
cout << "Register As User" << endl;
Color::setTextColor(Color::BrightCyan);
cout << "[";
Color::setTextColor(Color::BrightRed);
cout << "4";
Color::setTextColor(Color::BrightCyan);
cout << "]" << endl;
Color::setTextColor(Color::BrightGreen);
cout << "Exit" << endl;

Color::setTextColor(Color::Yellow);
cout << "Enter your choice: ";
// Set white text
Color::setTextColor(Color::White);
while (!(cin >> choice) || choice < 1 || choice > 4) {
    system("cls");
    Color::setTextColor(Color::Red);
    cout << "Invalid choice. ";
    Color::setTextColor(Color::White);
    cout << "Please enter your choice between ";
    Color::setTextColor(Color::BrightGreen);
    cout << "[1] ";
    Color::setTextColor(Color::Yellow);
    cout << "to ";
    Color::setTextColor(Color::BrightGreen);
    cout << "[4]";
    Color::setTextColor(Color::White);
    cout << " only.\n";

    cin.clear();
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    main();
}

switch (choice) {
case 1:
{
    system("cls");
    int id;
    string u, p;
    AdminProfile admin(id,u,p);
    admin.createDefaultAdminAccount();
    adminLogin();
}

break;
```



```
case 2:
{
    system("cls");
    userLogin();
}

break;
case 3:
{
    system("cls");
    registerUser();
}

break;
case 4:
{
    cout << "Exiting the program." << endl;
    system(0);
}

break;
default:
    cout << "Invalid choice!" << endl;
    break;
}
return 0;
}
```