DATA STRUCTURES AND ALGORITHM USING C++

COMMUNITY

EMERGENCY ALERT SYSTEM

CONNECT

"Community Connect" is an emergency alert system using SMTP servers and CURL.R for seamless email communication. It leverages IFTTT web applets for automation, ensuring real-time, customizable alerts for community safety.

MEET OUR TEAM

01 RAKESH SHARMA

SMTP SERVER,LINKING OF EMAIL USING IFTTT

02 MEGHNA SHARMA

MAIN FUNCTION,TERMINAL ,PRESENTATION

03 ANAND DUBEY

RESEARCH,REPORT

The Table
OF CONTENT

1 INTRODUCTION

2 OBJECTIVE

3 FEATURES

4 METHODOLOGY

5 CONCLUSION

6 DIVISION

INTRODUCTION

Welcome to our community management system, seamlessly integrating administration and resident functions. Admins oversee all residences and emergencies, managing data securely. Residents update family details and report emergencies intuitively. The project prioritizes security with robust sign-up processes, ensuring confidentiality. Swift emergency alerts, verified by the community, ensure a coordinated response for community safety.

Our comprehensive system promotes seamless community communication. Admins access vital databases, updating residence information and adding new administrators. Residents enjoy an intuitive interface for family management and emergency reporting. The secure signup process ensures data integrity and user privacy. Verified emergencies trigger progressive alerts, from community-wide notifications to emergency services, ensuring a swift and targeted response.

01 **SWIFT AND EFFECTIVE REPORTING OF EMERGENCIES**

02 **VERIFICATION OF REPORTED EMERGENCY TO AVOID UNNECESSARY ALLOCATION OF SERVICES**

03 **GENERATING ALERT TO THE COMMUNITY/ SOCIETY**

OBJECTIVE

Our project, "Community Connect," is a robust community management system designed to facilitate seamless communication and enhance safety during emergencies. The system integrates two main components: the Admin section, providing administrators with comprehensive oversight of residences and emergencies, and the Residence section, offering residents an intuitive platform to manage family details and report emergencies. Security is paramount, with rigorous signup processes ensuring data integrity and user confidentiality. The project features a progressive alert system, triggered by verified emergencies, to facilitate swift and targeted responses from both the community and emergency services. By prioritizing efficiency, security, and community collaboration, "Community Connect" aims to foster a safer and well-coordinated environment for residents and authorities alike.

Inefficient Emergency Communication

Problem: Existing residential communities often lack an efficient and coordinated system for communicating emergencies, leading to delays in response and potential gaps in information dissemination.

Limited Community Collaboration

Problem: Communities often face challenges in fostering collaboration among residents during emergency situations, leading to difficulties in verifying the authenticity of reported incidents.

Privacy and Security Concerns

Problem: Traditional emergency alert systems may not adequately address privacy and security concerns, potentially exposing sensitive resident information to unauthorized access.

METHODOLOGY

Communication Protocols

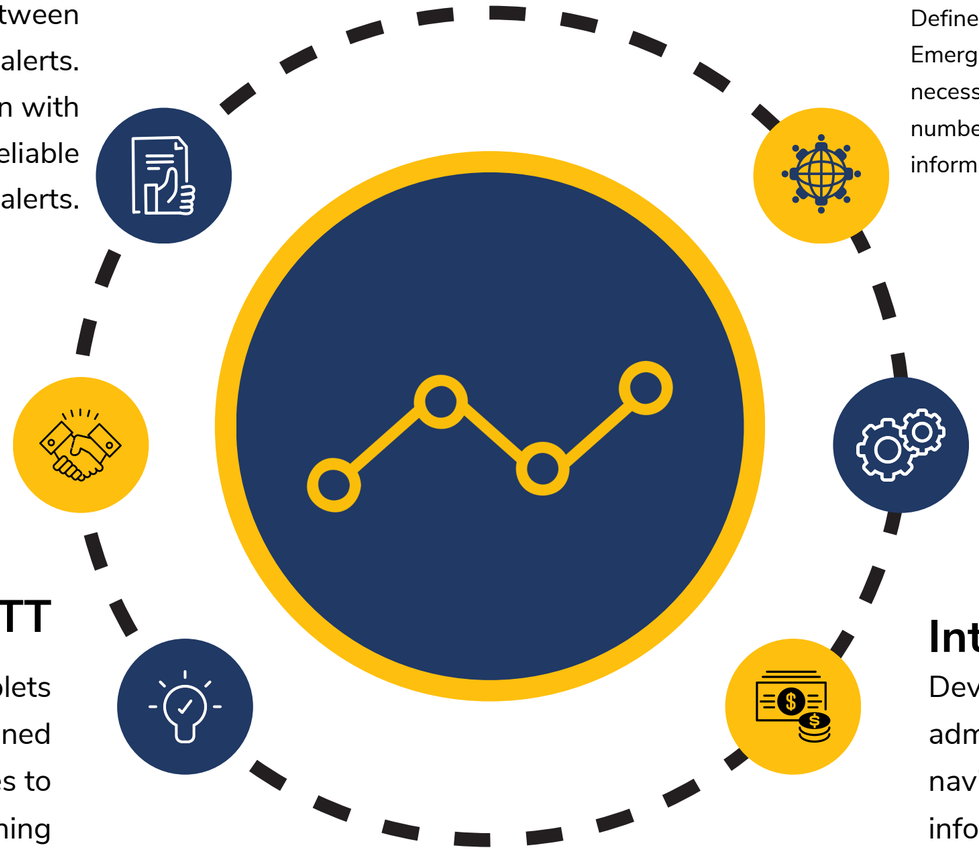
Establish communication protocols between the system and SMTP servers for email alerts. Utilize CURL.R for efficient interaction with email services, ensuring the reliable transmission of emergency alerts.

Verification Mechanism

Implement a verification mechanism for reported emergencies, involving the community in confirming incidents to prevent false alarms and optimize resource allocation

Integration with IFTTT

Integrate the system with IFTTT web applets to automate actions based on predefined conditions, facilitating efficient responses to emergency situations and streamlining communication processes.



Database Design

Define and design the Admin, Residence, and Emergency databases, specifying the necessary fields for each, such as house number, family details, and emergency information

User Authentication and Security Measures

Implement a secure user authentication system for both administrators and residents, incorporating encryption techniques and access controls to safeguard sensitive data

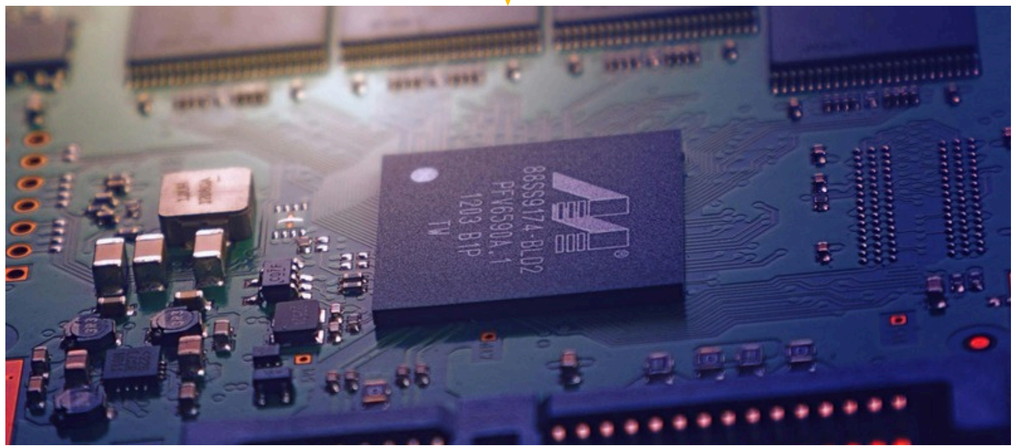
Interface Development

Develop user-friendly interfaces for both administrators and residents, ensuring intuitive navigation for tasks such as updating information, reporting emergencies, and managing user profiles.

KEY FEATURES

FEATURES

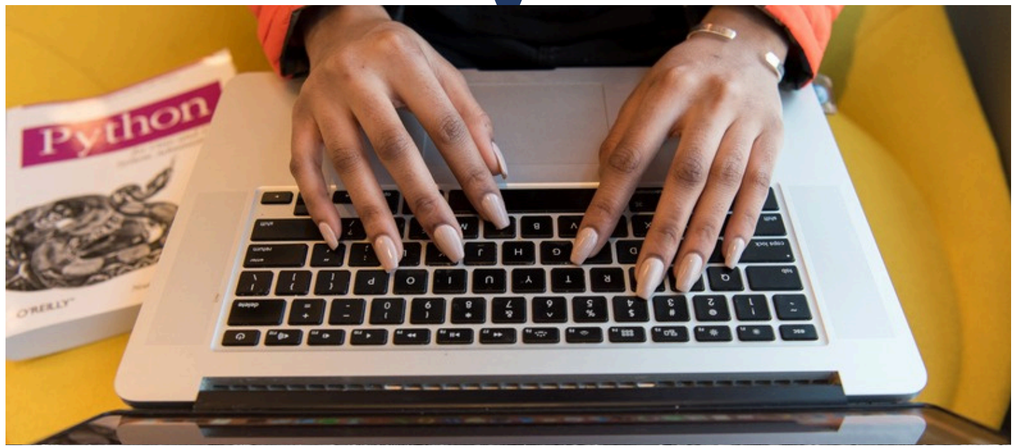
- **Comprehensive Admin Access**
- **Resident-Focused Interface**
- **Secure User Sign-Up**
- **Emergency Reporting and Verification**
- **Progressive Alert System**
- **Efficient Emergency Services Notification**
- **Data Integrity and Privacy Measures**



ALGORITHMS

DSA USED

- LINKED LIST FOR DATABASE
- LINEAR SEARCH FOR FETCHING DATA FROM DATABASE
- BINARY SORTING FOR ADDED NEW DATA IN DATABASE
- ARRAYS & VECTORS FOR STORING DATA



CONCLUSION

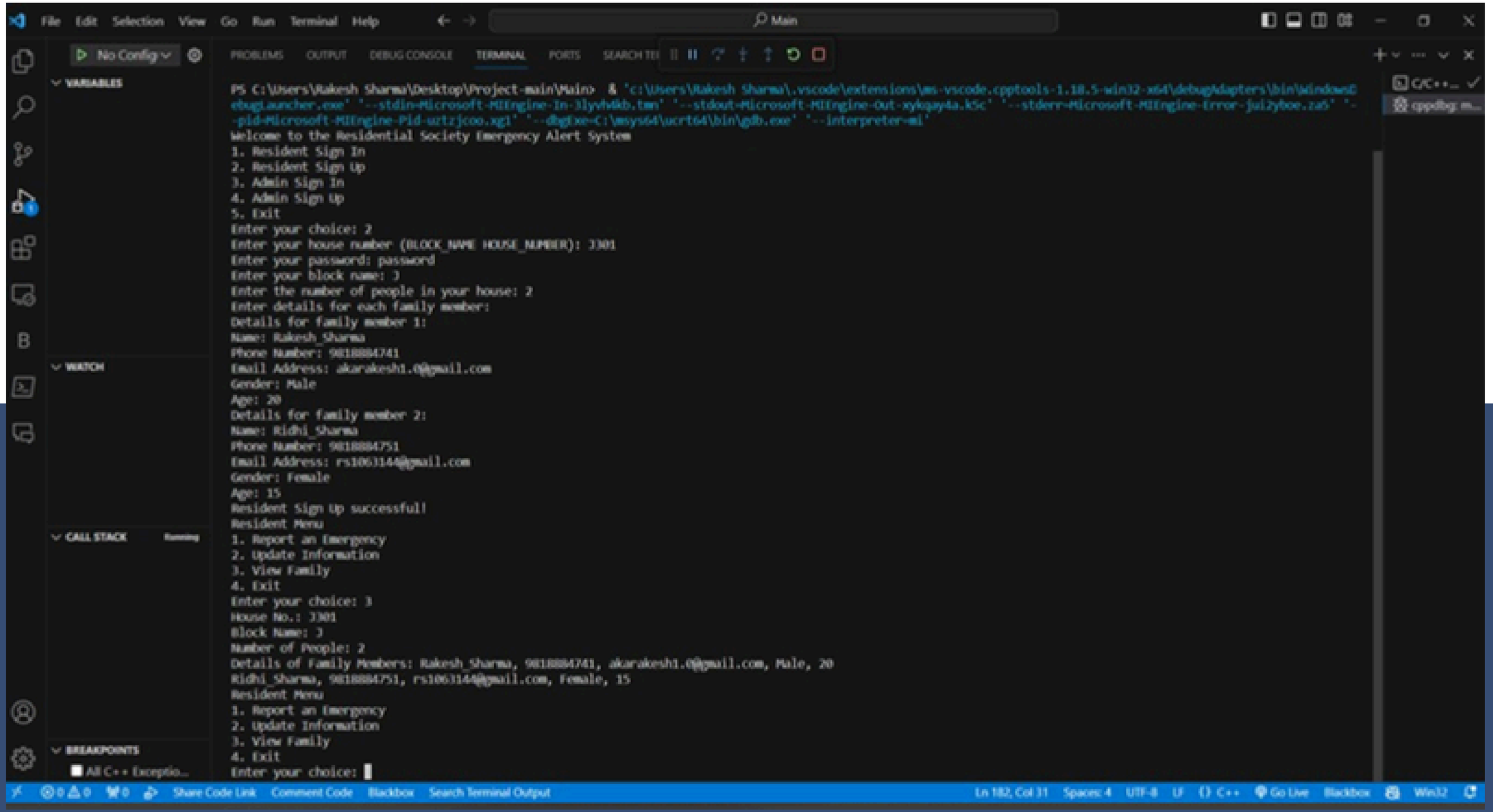
In conclusion, "Community Connect" represents an innovative and inclusive approach to community management and emergency response. By seamlessly integrating administrative oversight and resident engagement, the project fosters a sense of collective responsibility and swift action during critical situations. The robust security measures ensure the integrity of user data, prioritizing privacy. The progressive alert system, coupled with community verification, optimizes the use of resources and enables efficient collaboration with emergency services. As we navigate the dynamic landscape of community safety, "Community Connect" stands as a reliable, adaptive, and user-centric solution, fostering a safer and well-coordinated environment for residents and authorities alike.

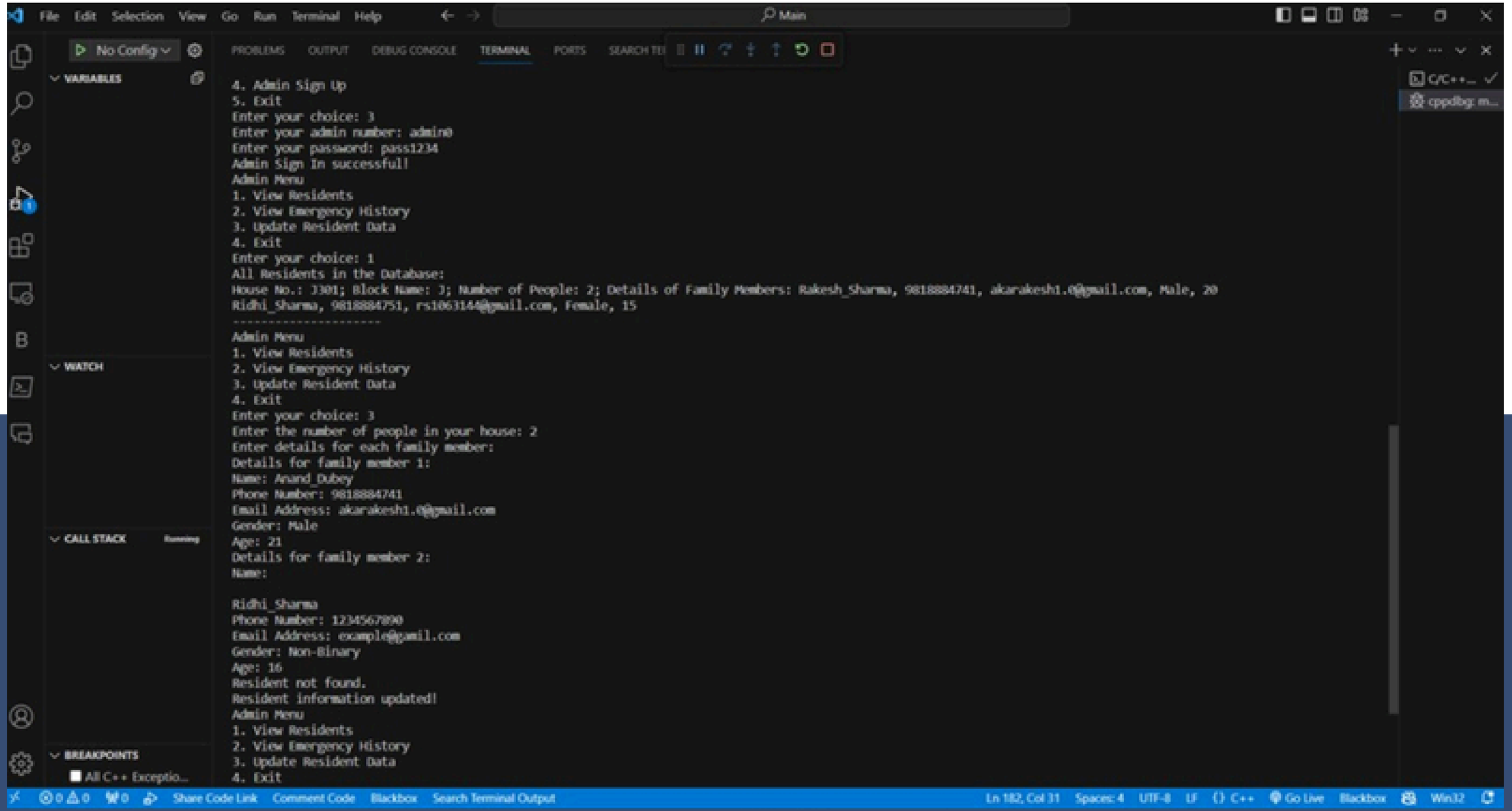
```
int main() {
    int choice;
    while (true) {
        cout << "Welcome to the Residential Society Emergency Alert System\n";
        cout << "1. Resident Sign In\n";
        cout << "2. Resident Sign Up\n";
        cout << "3. Admin Sign In\n";
        cout << "4. Admin Sign Up\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                residentSignIn();
                break;
            case 2:
                residentSignUp();
                break;
            case 3:
                adminSignIn();
                break;
            case 4:
                adminSignUp();
                break;
            case 5:
                cout << "Exiting the program. Goodbye!\n";
                return 0;
            default:
                cout << "Invalid choice. Please try again.\n";
        }
    }

    return 0;
}
```

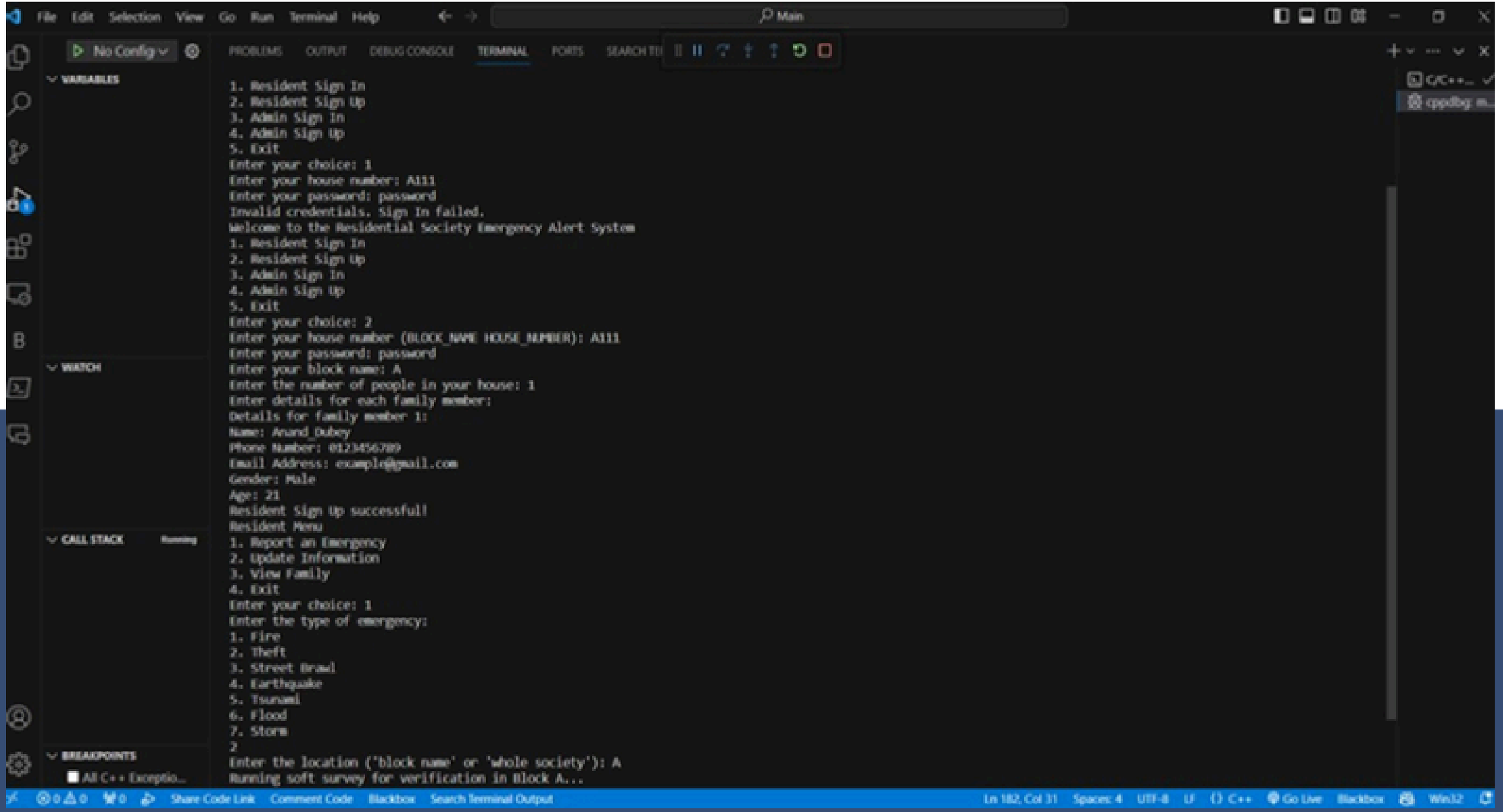
```
void residentSignIn() {
    string houseNo, password;
    cout << "Enter your house number: ";
    cin >> houseNo;
    cout << "Enter your password: ";
    cin >> password;

    Resident* resident = residentDB.getResident(houseNo);

    if (resident != nullptr && resident->password == password) {
        cout << "Resident Sign In successful!\n";
        residentMenu(houseNo);
    } else {
        cout << "Invalid credentials. Sign In failed.\n";
    }
}
```

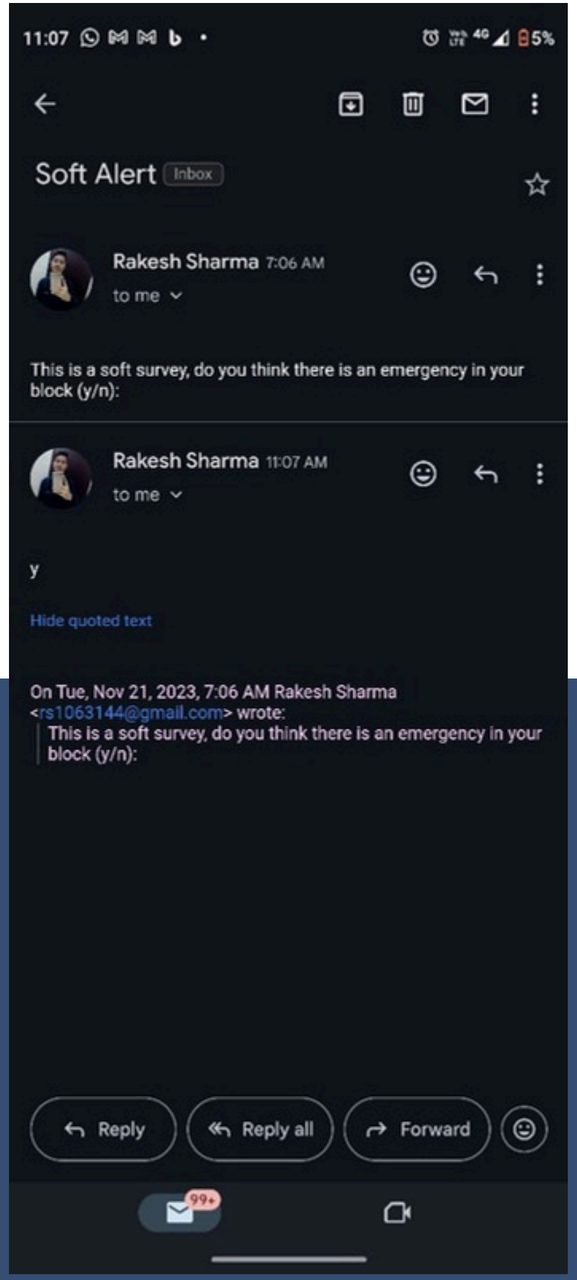
```
void adminSignIn() {
    string adminNo, password;
    cout << "Enter your admin number: ";
    cin >> adminNo;
    cout << "Enter your password: ";
    cin >> password;

    Admin* admin = adminDB.getAdmin(adminNo);

    if (admin != nullptr && admin->password == password) {
        cout << "Admin Sign In successful!\n";
        adminMenu();
    } else {
        cout << "Invalid credentials. Sign In failed.\n";
    }
}
```

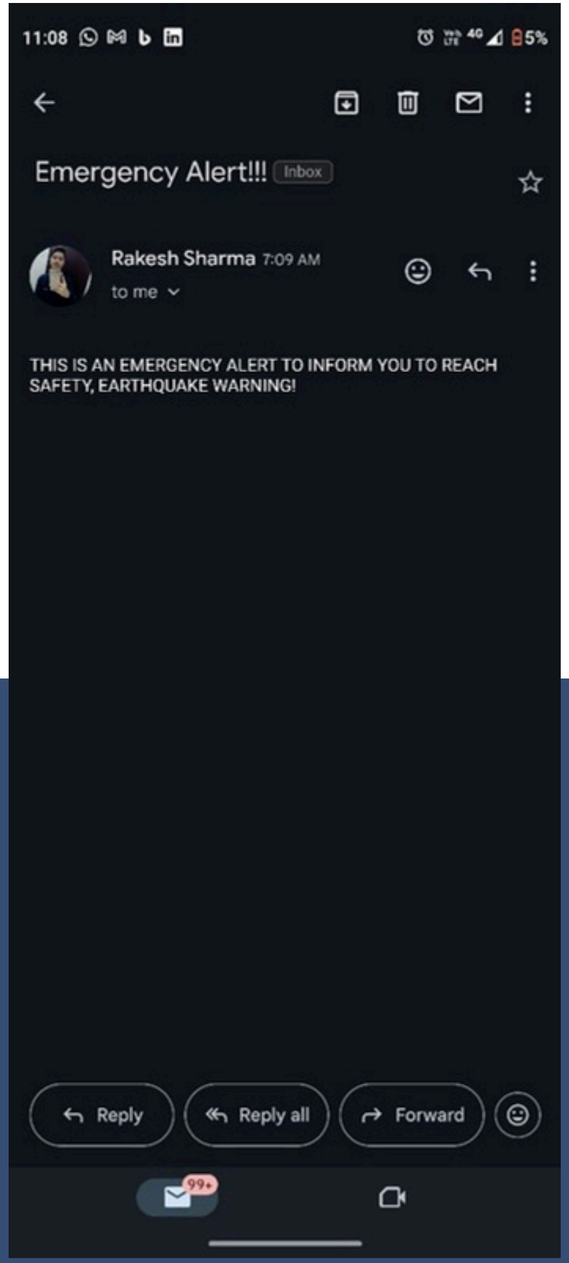
```
void adminMenu() {
    int choice;
    while (true) {
        cout << "Admin Menu\n";
        cout << "1. View Residents\n";
        cout << "2. View Emergency History\n";
        cout << "3. Update Resident Data\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        string houseNo;

        switch (choice) {
            case 1:
                cout << "All Residents in the Database:\n";
                residentDB.displayAllResidents(); // Display all residents
                break;
            case 2:
                cout << "All Emergencies in the Database:\n";
                emergencyDB.displayAllEmergencies(); // Display all emergencies
                break;
            case 3:
                int numOfPeople;
                cout << "Enter the number of people in your house: ";
                cin >> numOfPeople;

                string* names = new string[numOfPeople];
                string* phoneNumbers = new string[numOfPeople];
                string* emailAddresses = new string[numOfPeople];
                string* gender = new string[numOfPeople];
                int* age = new int[numOfPeople];

                cout << "Enter details for each family member:\n";
                for (int i = 0; i < numOfPeople; ++i) {
                    cout << "Details for family member " << i + 1 << ":\n";
                    cout << "Name: ";
                    cin >> names[i];
                    cout << "Phone Number: ";
                    cin >> phoneNumbers[i];
                    cout << "Email Address: ";
                    cin >> emailAddresses[i];
                    cout << "Gender: ";
                    cin >> gender[i];
                    cout << "Age: ";
                    cin >> age[i];
                }
                residentDB.updateResident(houseNo, numOfPeople, names, phoneNumbers, emailAddresses, ge
                cout << "Resident information updated!\n";
                break;
            case 4:
                cout << "Exiting Admin Menu.\n";
                return;
            default:
                cout << "Invalid choice. Please try again.\n";
        }
```

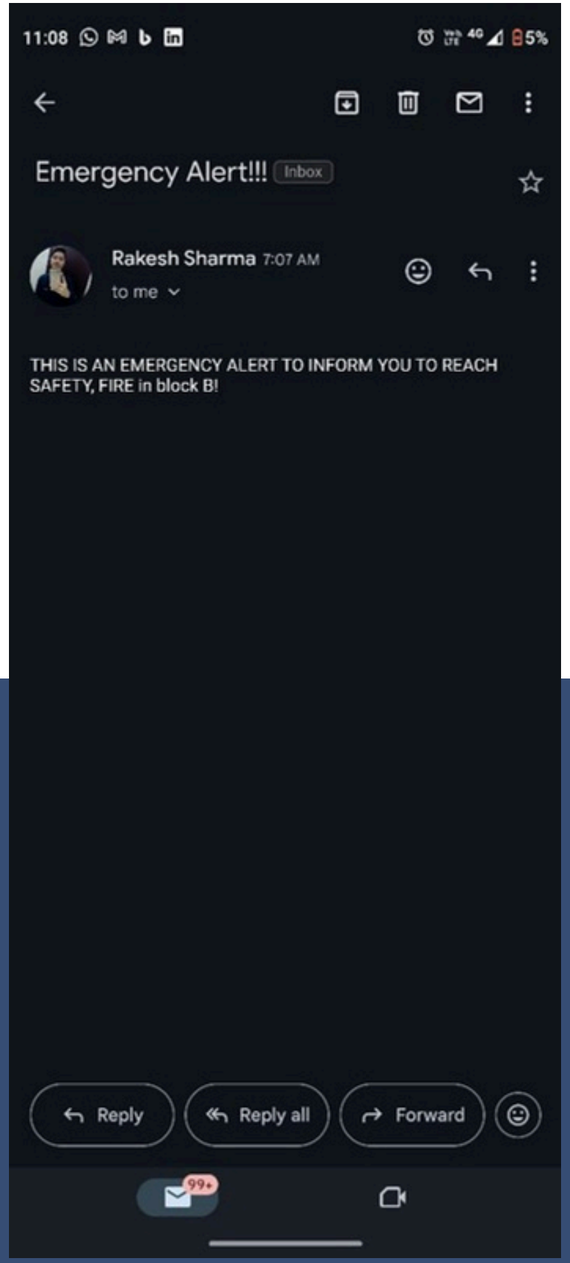

File Edit Selection View Go Run Terminal Help
Main
No Config
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH
VARIABLES
WATCH
CALL STACK Running
BREAKPOINTS
All C++ Exceptio...
PS C:\Users\Rakesh Sharma\Desktop\Project-main\Main> & 'c:\Users\Rakesh Sharma\.vscode\extensions\ms-vscode.cpptools-1.18.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-3lyvhvkb.tmn' '--stdout=Microsoft-MIEngine-Out-xykqay4a.k5c' '--stderr=Microsoft-MIEngine-Error-jui2yboe.za5' '--pid=Microsoft-MIEngine-Pid-uztzjcoo.xg1' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Welcome to the Residential Society Emergency Alert System
1. Resident Sign In
2. Resident Sign Up
3. Admin Sign In
4. Admin Sign Up
5. Exit
Enter your choice: 2
Enter your house number (BLOCK_NAME HOUSE_NUMBER): J301
Enter your password: password
Enter your block name: J
Enter the number of people in your house: 2
Enter details for each family member:
Details for family member 1:
Name: Rakesh_Sharma
Phone Number: 9818884741
Email Address: akarakesh1.0@gmail.com
Gender: Male
Age: 20
Details for family member 2:
Name: Ridhi_Sharma
Phone Number: 9818884751
Email Address: rs10631144@gmail.com
Gender: Female
Age: 15
Resident Sign Up successful!
Resident Menu
1. Report an Emergency
2. Update Information
3. View Family
4. Exit
Enter your choice: 3
House No.: J301
Block Name: J
Number of People: 2
Details of Family Members: Rakesh_Sharma, 9818884741, akarakesh1.0@gmail.com, Male, 20
Ridhi_Sharma, 9818884751, rs10631144@gmail.com, Female, 15
Resident Menu
1. Report an Emergency
2. Update Information
3. View Family
4. Exit
Enter your choice:
GdC++...
cppdbg: m...
Share Code Link Comment Code Blackbox Search Terminal Output
Ln 182, Col 31 Spaces: 4 UTF-8 LF C++ Go Live Blackbox Win32

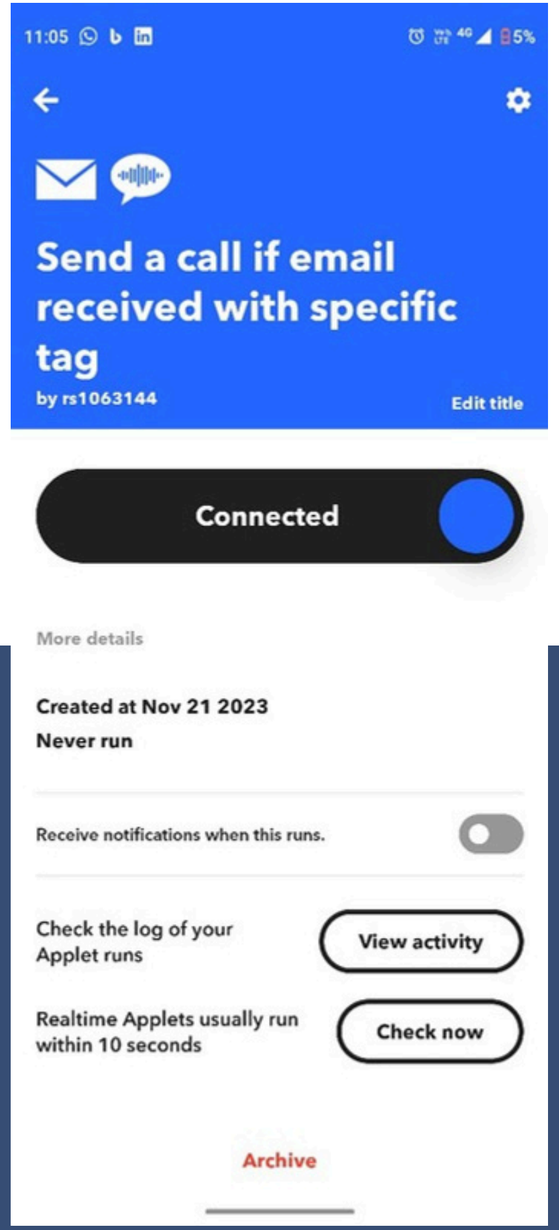

File Edit Selection View Go Run Terminal Help
Main
No Config
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
VARIABLES
WATCH
CALL STACK Running
BREAKPOINTS
All C++ Exceptio...
4. Admin Sign Up
5. Exit
Enter your choice: 3
Enter your admin number: admin0
Enter your password: pass1234
Admin Sign In successful!
Admin Menu
1. View Residents
2. View Emergency History
3. Update Resident Data
4. Exit
Enter your choice: 1
All Residents in the Database:
House No.: 3301; Block Name: J; Number of People: 2; Details of Family Members: Rakesh_Sharma, 9818884741, akarakesh1.0@gmail.com, Male, 20
Ridhi_Sharma, 9818884751, rs1063144@gmail.com, Female, 15
Admin Menu
1. View Residents
2. View Emergency History
3. Update Resident Data
4. Exit
Enter your choice: 3
Enter the number of people in your house: 2
Enter details for each family member:
Details for family member 1:
Name: Anand_Dubey
Phone Number: 9818884741
Email Address: akarakesh1.0@gmail.com
Gender: Male
Age: 21
Details for family member 2:
Name:
Ridhi_Sharma
Phone Number: 1234567890
Email Address: example@gmail.com
Gender: Non-Binary
Age: 16
Resident not found.
Resident information updated!
Admin Menu
1. View Residents
2. View Emergency History
3. Update Resident Data
4. Exit
C/C++...
cppdbg: m...
Share Code Link Comment Code Blackbox Search Terminal Output
Ln 182, Col 31 Spaces: 4 UTF-8 LF C++ Go Live Blackbox Win32

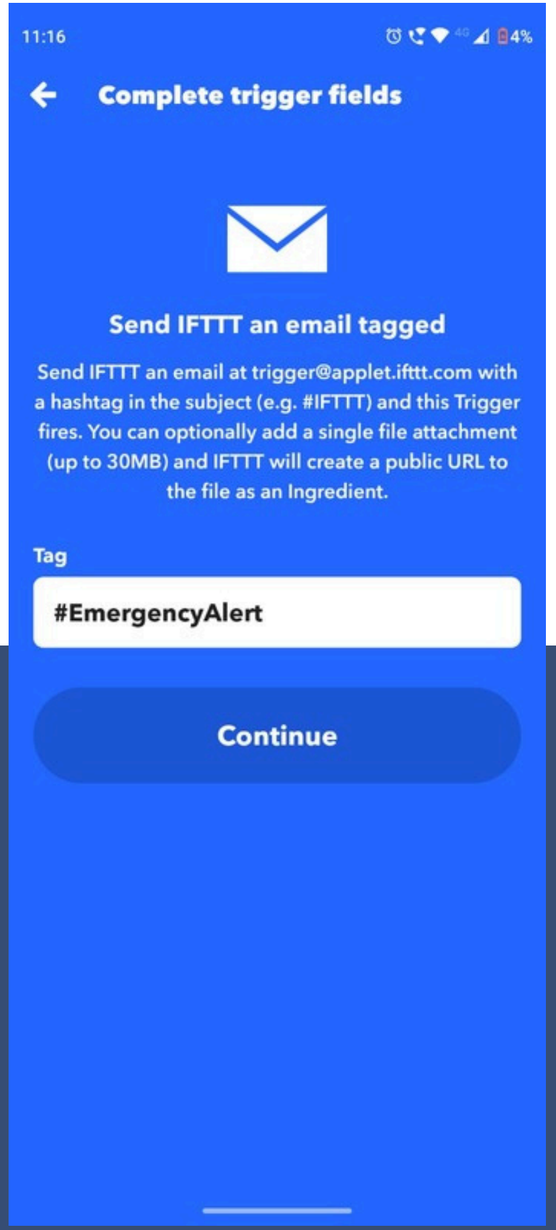

File Edit Selection View Go Run Terminal Help
Main
No Config
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
VARIABLES
WATCH
CALL STACK
BREAKPOINTS
1. Resident Sign In
2. Resident Sign Up
3. Admin Sign In
4. Admin Sign Up
5. Exit
Enter your choice: 1
Enter your house number: A111
Enter your password: password
Invalid credentials. Sign In failed.
Welcome to the Residential Society Emergency Alert System
1. Resident Sign In
2. Resident Sign Up
3. Admin Sign In
4. Admin Sign Up
5. Exit
Enter your choice: 2
Enter your house number (BLOCK_NAME HOUSE_NUMBER): A111
Enter your password: password
Enter your block name: A
Enter the number of people in your house: 1
Enter details for each family member:
Details for family member 1:
Name: Anand Dubey
Phone Number: 0123456789
Email Address: example@gmail.com
Gender: Male
Age: 21
Resident Sign Up successful!
Resident Menu
1. Report an Emergency
2. Update Information
3. View Family
4. Exit
Enter your choice: 1
Enter the type of emergency:
1. Fire
2. Theft
3. Street Brawl
4. Earthquake
5. Tsunami
6. Flood
7. Storm
2
Enter the location ('block name' or 'whole society'): A
Running soft survey for verification in Block A...
Share Code Link Comment Code Blackbox Search Terminal Output
Ln 182, Col 31 Spaces: 4 UTF-8 LF C++ Go Live Blackbox Win32


11:07
Soft Alert
Inbox
Rakesh Sharma 7:06 AM
to me
This is a soft survey, do you think there is an emergency in your block (y/n):
Rakesh Sharma 11:07 AM
to me
y
Hide quoted text
On Tue, Nov 21, 2023, 7:06 AM Rakesh Sharma <rs1063144@gmail.com> wrote:
This is a soft survey, do you think there is an emergency in your block (y/n):
Reply
Reply all
Forward


11:08
Emergency Alert!!!
Inbox
Rakesh Sharma 7:09 AM
to me
THIS IS AN EMERGENCY ALERT TO INFORM YOU TO REACH SAFETY, EARTHQUAKE WARNING!
Reply
Reply all
Forward


11:08
Emergency Alert!!!
Inbox
Rakesh Sharma 7:07 AM
to me
THIS IS AN EMERGENCY ALERT TO INFORM YOU TO REACH SAFETY, FIRE in block B!
Reply
Reply all
Forward


11:05
5%
Send a call if email received with specific tag
by rs1063144
Edit title
Connected
More details
Created at Nov 21 2023
Never run
Receive notifications when this runs.
Check the log of your Applet runs
View activity
Realtime Applets usually run within 10 seconds
Check now
Archive


11:16
4%
Complete trigger fields
Send IFTTT an email tagged
Send IFTTT an email at trigger@applet.ifttt.com with a hashtag in the subject (e.g. #IFTTT) and this Trigger fires. You can optionally add a single file attachment (up to 30MB) and IFTTT will create a public URL to the file as an Ingredient.
Tag
#EmergencyAlert
Continue


11:14
4%
IFTTT
VoIP Calls Audio 00:02
Speaker
Leave call

THANK YOU