

Laiba Maab

CID: DEP2248

Main.cpp

```
#include "contact.h"

int main() {

    std::vector<Contact> contacts;

    int choice;

    do {

        std::cout << "\n\nContact Management System\n\n";

        std::cout << "1. Add Contact\n";

        std::cout << "2. View Contacts\n";

        std::cout << "3. Delete Contact\n";

        std::cout << "4. Exit\n";

        std::cout << "Enter your choice: ";

        std::cin >> choice;

        switch (choice) {

            case 1:

                addContact(contacts);

                break;

            case 2:

                viewContacts(contacts);

                break;

            case 3:

                deleteContact(contacts);

                break;

            case 4:
```

```

        std::cout << "Exiting\n";
        break;
    default:
        std::cout << "Invalid choice, please try again.\n";
        break;
    }
} while (choice != 4);

return 0;
}

contact.h
#include <iostream>
#include <string>
#include <vector>

class Contact
{
public:
    std::string name;
    std::string phoneNumber;
    Contact(std::string n, std::string p) : name(n), phoneNumber(p)
    {}
    void displayContact()
    {
        std::cout << "Name: " << name << std::endl;
        std::cout << "Phone Number: " << phoneNumber << std::endl;
    }
};

void addContact(std::vector<Contact> &contacts)

```

```

{
    std::string name, phoneNumber;
    std::cout << "Enter name: ";
    std::cin >> name;
    std::cout << "Enter phone number: ";
    std::cin >> phoneNumber;
    contacts.push_back(Contact(name, phoneNumber));
    std::cout << "Contact added successfully."<<std::endl;
}

void viewContacts(std::vector<Contact> &contacts)
{
    if (contacts.empty())
    {
        std::cout << "No contacts available."<<std::endl;
        return;
    }
    for (auto &contact : contacts)
    {
        contact.displayContact();
    }
}

void deleteContact(std::vector<Contact> &contacts)
{
    std::string name;
    std::cout << "Enter the name of the contact to delete: ";
    std::cin >> name;
    for (auto it = contacts.begin(); it != contacts.end(); ++it)
    {

```

```
    if (it->name == name)
    {
        contacts.erase(it);
        std::cout << "Contact deleted successfully."<<std::endl;
        return;
    }
}
std::cout << "Contact not found."<<std::endl;
}
```

<https://onlinegdb.com/hZwBaPFah>