yahia.cpp > main()

```cpp
#include <iostream>
#include <string>
using namespace std;

const int MAX_BOOKS = 10;
const int MAX_USERS = 10;
const int MAX_PAGES = 5;

class Book {
public:
    string title;
    string author;
    string pages[MAX_PAGES];

    Book() {
        for (int i = 0; i < MAX_PAGES; i++) {
            pages[i] = "Page " + to_string(i + 1) + " content of the book.";
        }
    }
};

class User {
public:
    string username;
    string password;

    virtual void viewProfile() = 0;
};
```

```cpp
class User {
public:
    string username;
    string password;

    virtual void viewProfile() = 0;
};

class Admin : public User {
public:
    void viewProfile() override {
        cout << "Admin Username: " << username << endl;
    }

    void addBook(Book books[], int& bookCount) {
        if (bookCount >= MAX_BOOKS) {
            cout << "Book limit reached!\n";
            return;
        }
        cout << "Enter book title: ";
        cin.ignore();
        getline(cin, books[bookCount].title);
        cout << "Enter author name: ";
        getline(cin, books[bookCount].author);
        bookCount++;
        cout << "Book added successfully!\n";
    }
};
```

```
class Session {
public:
    Book* book;
    int currentPage;

    Session() {
        book = nullptr;
        currentPage = 0;
    }

    void start(Book* b) {
        book = b;
        currentPage = 0;
    }

    void showPage() {
        if (book) {
            cout << "\nReading: " << book->title << " | Page " << (currentPage + 1) << "/" << MAX_PAGES << endl;
            cout << book->pages[currentPage] << endl;
        } else {
            cout << "No book selected.\n";
        }
    }

    void nextPage() {
        if (book && currentPage < MAX_PAGES - 1) {
            currentPage++;
            showPage();
        } else {
            cout << "You are on the last page.\n";
        }
    }
```

```
    void nextPage() {
        if (book && currentPage < MAX_PAGES - 1) {
            currentPage++;
            showPage();
        } else {
            cout << "You are on the last page.\n";
        }
    }

    void prevPage() {
        if (book && currentPage > 0) {
            currentPage--;
            showPage();
        } else {
            cout << "You are on the first page.\n";
        }
    }
};

class Customer : public User {
public:
    Session session;

    void viewProfile() override {
        cout << "Customer Username: " << username << endl;
    }

    void viewBooks(Book books[], int bookCount) {
        if (bookCount == 0) {
            cout << "No books available.\n";
            return;
        }
        for (int i = 0; i < bookCount; i++) {
            cout << i + 1 << ". " << books[i].title << " by " << books[i].author << endl;
        }
    }
```

```
112     void startReading(Book books[], int bookCount) {
113         int choice;
114         viewBooks(books, bookCount);
115         cout << "Enter book number to read: ";
116         cin >> choice;
117         if (choice < 1 || choice > bookCount) {
118             cout << "Invalid choice.\n";
119             return;
120         }
121         session.start(&books[choice - 1]);
122         session.showPage();
123         char op;
124         do {
125             cout << "\n[n]ext, [p]revious, [q]uit: ";
126             cin >> op;
127             if (op == 'n') session.nextPage();
128             else if (op == 'p') session.prevPage();
129         } while (op != 'q');
130     }
131 };
132 
133 int login(User* users[], int userCount, string type) {
134     string user, pass;
135     cout << "Enter username: ";
136     cin >> user;
137     cout << "Enter password: ";
138     cin >> pass;
139     for (int i = 0; i < userCount; i++) {
140         if (users[i]->username == user && users[i]->password == pass) {
141             if ((type == "admin" && dynamic_cast<Admin*>(users[i])) ||
142                 (type == "customer" && dynamic_cast<Customer*>(users[i]))) {
143                 return i;
144             }
145         }
146     }
147     return -1;
```

```cpp
int main() {
    Book books[MAX_BOOKS];
    int bookCount = 0;

    User* users[MAX_USERS];
    int userCount = 0;

    Admin* admin1 = new Admin();
    admin1->username = "admin";
    admin1->password = "123";
    users[userCount++] = admin1;

    Customer* cust1 = new Customer();
    cust1->username = "user";
    cust1->password = "123";
    users[userCount++] = cust1;

    while (true) {
        cout << "\n===== Main Menu =====\n";
        cout << "1. Admin Login\n";
        cout << "2. Customer Login\n";
        cout << "3. Sign Up\n";
        cout << "4. Exit\nChoose: ";
        int choice;
        cin >> choice;

        if (choice == 1) {
            int idx = login(users, userCount, "admin");
            if (idx == -1) {
                cout << "Invalid admin credentials.\n";
                continue;
            }
            Admin* a = dynamic_cast<Admin*>(users[idx]);
            int op;
            do {
                cout << "\n--- Admin Menu ---\n";
```

```
184         do {
185             cout << "\n--- Admin Menu ---\n";
186             cout << "1. View Profile\n";
187             cout << "2. Add Book\n";
188             cout << "3. Logout\nChoose: ";
189             cin >> op;
190             if (op == 1) a->viewProfile();
191             else if (op == 2) a->addBook(books, bookCount);
192         } while (op != 3);
193     }
194
195     else if (choice == 2) {
196         int idx = login(users, userCount, "customer");
197         if (idx == -1) {
198             cout << "Invalid customer credentials.\n";
199             continue;
200         }
201         Customer* c = dynamic_cast<Customer*>(users[idx]);
202         int op;
203         do {
204             cout << "\n--- Customer Menu ---\n";
205             cout << "1. View Profile\n";
206             cout << "2. View Books\n";
207             cout << "3. Read Book\n";
208             cout << "4. Logout\nChoose: ";
209             cin >> op;
210             if (op == 1) c->viewProfile();
211             else if (op == 2) c->viewBooks(books, bookCount);
212             else if (op == 3) c->startReading(books, bookCount);
213         } while (op != 4);
214     }
215
216     else if (choice == 3) {
217         if (userCount >= MAX_USERS) {
```

```
    else if (choice == 3) {
        if (userCount >= MAX_USERS) {
            cout << "User limit reached!\n";
            continue;
        }
        int userType;
        cout << "Sign Up as: 1. Admin  2. Customer: ";
        cin >> userType;

        string user, pass;
        cout << "Choose username: ";
        cin >> user;
        cout << "Choose password: ";
        cin >> pass;

        bool exists = false;
        for (int i = 0; i < userCount; i++) {
            if (users[i]->username == user) {
                exists = true;
                break;
            }
        }

        if (exists) {
            cout << "Username already exists!\n";
            continue;
        }
```

```
            if (exists) {
                cout << "Username already exists!\n";
                continue;
            }

            if (userType == 1) {
                Admin* a = new Admin();
                a->username = user;
                a->password = pass;
                users[userCount++] = a;
                cout << "Admin registered successfully.\n";
            } else if (userType == 2) {
                Customer* c = new Customer();
                c->username = user;
                c->password = pass;
                users[userCount++] = c;
                cout << "Customer registered successfully.\n";
            } else {
                cout << "Invalid user type.\n";
            }
        }

        else if (choice == 4) {
            cout << "Exiting program. Goodbye!\n";
            break;
        }
    }

    return 0;
}
```