UNIVERSITY OF GUJRAT

A WORLD CLASS UNIVERSITY

Project: Search Engine

Object Oriented Programming

Submitted to

Mr. Naveed Abbas

Submitted By

Name:	M Hanzala Zaheer
Roll no:	22014198-060
Name:	Gul e Raana
Roll no:	22014198-049
Name:	Iqra Aslam
Roll no:	22014198-044

Class:	BSSE-A-22
Course code:	SE-103
Submit Date:	1 /08/2023

“SOURCE CODE”

```
#include <iostream>
#include <istream>
#include <fstream>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include <vector>
#include <iomanip>
using namespace std;
//all function used in program
void login();
void registr();
void forgot();
void menu();
void search_engine_page();
void search();
void add_keywords();
void check_keywords();
void Edit_keyword();
void project_page();
//global variables
char ch;
int i , count = 0;
//main function
int main()
{
        if (count == 1)
        {
                search_engine_page();
        }
        else{
                project_page();
                menu();
        }
        return 0;
}
```

```

void project_page()
{
        cout << right << endl
             << endl;
        cout << setw(30) << " "
             << "____________________________________________________________________________________________________\n";
        cout << setw(30) << " "
             << "|                                                                                                   |\n";
        cout << setw(30) << " "
             << "|                                 Project:      SEARCH ENGINE                                       |\n";
        cout << setw(30) << " "
             << "|                                                                                                   |\n";
        cout << setw(30) << " "
             << "|                                 Submitted to: Mr. Naveed Abbas                                    |\n";
        cout << setw(30) << " "
             << "|                                                                                                   |\n";
        cout << setw(30) << " "
             << "|                                        submitted by                                               |\n";
        cout << setw(30) << " "
             << "|                                                                                                   |\n";
        cout << setw(30) << " "
             << "|                                 Name:         M Hanzala Zaheer                                    |\n";
        cout << setw(30) << " "
             << "|                                 Roll no:      22014198-060                                        |\n";
        cout << setw(30) << " "
             << "|                                                                                                   |\n";
        cout << setw(30) << " "
             << "|                                 Name:         Gul e Raana                                         |\n";
        cout << setw(30) << " "
             << "|                                 Roll no:      22014198-049                                        |\n";
        cout << setw(30) << " "
             << "|                                                                                                   |\n";
        cout << setw(30) << " "
             << "|                                 Name:         Iqra Aslam                                          |\n";
        cout << setw(30) << " "
             << "|                                 Roll no:      22014198-044                                        |\n";
        cout << setw(30) << " "
             << "|                                                                                                   |\n";
```

```cpp
        cout << setw(30) << " "
             << "|__________________________________________________________________________________________________|\n";

        cout << endl
             << endl
             << endl
             << setw(30) << " ";
        system("pause");
}

void Edit_keyword()
{
        // Declare variables
        string keyword;
        string line;
        string new_keyword;
        vector<string> keywords_vec;
        bool keyword_exists;
        int index = 0;

        // Open the file in read mode
        ifstream file("keywords.txt");

        // read each line and store it in a vector

        while (getline(file, line))
        {
                keywords_vec.push_back(line);
        }

        // Close the file
        file.close();

        // Get the keyword to edit
        cout << "Enter the keyword you want to edit: ";
        getline(cin, keyword);
        ifstream readf("keywords.txt");
        while (getline(readf, line))
        {
                // Check if the keyword is equal to the line
                if (keyword == line)
                {
                        keyword_exists = true;
                        break;
                }
        }
        readf.close();
        if (keyword_exists == false)
        {
                cout << keyword << " keyword does not exist in database\n";
                system("pause");
                search_engine_page();
        }

        // Find the index of the keyword in the vector
        for (int i = 0; i < keywords_vec.size(); i++)
        {
                if (keywords_vec[i] == keyword)
                {
                        index = i;
                        break;
                }
        }

        // If the keyword was found, ask the user for the new keyword
        {
                cout << "Enter the new keyword: ";
                getline(cin, new_keyword);

                // Replace the old keyword with the new keyword
                keywords_vec[index] = new_keyword;

                // Open the file in write mode
                ofstream file("keywords.txt");

                // Write the new keywords to the file
                for (string keyword : keywords_vec)
                {
```

```cpp
                file << keyword << endl;
            }

            // Close the file
            file.close();
        }
        system("pause");
        search_engine_page();
}

void check_keywords()
{
        system("cls");
        cout << "List of Keywords\n--------------------------------------------------\n";
        string line;
        ifstream read("keywords.txt");
        while (getline(read, line))
        {
                cout << line << endl;
        }
        cout << "\n\n";
        system("pause");
        search_engine_page();
}
void add_keywords()
{
        // Declare variables
        string keyword;
        string line;
        int num_keywords;

        cout << "How many keywords do you want to add? ";
        cin >> num_keywords;
        cin.ignore();
        // Create a file to store the keywords
        ofstream file("keywords.txt", ios::app);
        for (int i = 0; i < num_keywords; i++)
        {
                // Get input from the user
```

```
			cout << "Enter keyword no " << i + 1 << " : ";
			getline(cin, keyword);
			// Write the keyword to the file in the next line
			file << keyword << endl;
		}
		// Close the file
		file.close();
		system("pause");
		search_engine_page();
}

void search()
{
		// Declare variables
		string keyword;
		string line;
		fstream file;

		// Prompt the user for a keyword
		cout << "Enter a keyword: ";
		cin >> keyword;

		// Open the file in read mode
		file.open("keywords.txt", ios::in);

		// Search for the keyword in the file
		while (getline(file, line))
		{
			// If the line contains the keyword, print it
			if (line.find(keyword) != string::npos)
			{
				cout << line << endl;
			}
		}

		// Close the file
		file.close();
		system("pause");
		search_engine_page();
```

```cpp
}

void search_engine_page()
{
        int op;
        system("cls");
        cout << right << endl
             << endl;
        cout << setw(30) << " "
             << "________________________________________________________________________________\n";
        cout << setw(30) << " "
             << "|                                                                               |\n";
        cout << setw(30) << " "
             << "|                                 Search Engine                                 |\n";
        cout << setw(30) << " "
             << "|_______________________________________________________________________________|\n\n";
        cout << setw(30) << " "
             << " _______________________________________________________________________________\n";
        cout << setw(30) << " "
             << "|                                                                               |\n";
        cout << setw(30) << " "
             << "|                               1. Search                                       |\n";
        cout << setw(30) << " "
             << "|                               2. Add keywords or Queries                      |\n";
        cout << setw(30) << " "
             << "|                               3. Keywords List                                |\n";
        cout << setw(30) << " "
             << "|                               4. Edit Keywords                                |\n";
        cout << setw(30) << " "
             << "|                               5. Log Out                                      |\n";
        cout << setw(30) << " "
             << "|_______________________________________________________________________________|\n\n";

        cout << setw(30) << " "
             << "----> Enter your choice: ";
        cin >> op;

        cin.ignore();
        switch (op)
        {
        case 1:
                // search
                system("cls");
                search();

                break;
        case 2:
                // add keywords
                system("cls");
                add_keywords();
                break;
        case 3:
                // check keywords
                system("cls");
                check_keywords();
                break;
        case 4:
                // edit keywords
                system("cls");
                Edit_keyword();
                break;
        case 5:
                // logout
                menu();
                break;
        default:
                cout << endl
                     << setw(30) << " "
                     << "Enter a valid option!\n";
                cout << setw(30) << " ";
                system("pause");
                search_engine_page();
                break;
        }
}
```

```cpp
void menu()
{
        system("cls");
        int choice;
        cout << endl
             << endl;
        cout << setw(30) << " "
             << "________________________________________________________________________________\n";
        cout << setw(30) << " "
             << "/                                                                               /\n";
        cout << setw(30) << " "
             << "/                                Search Engine                                  /\n";
        cout << setw(30) << " "
             << "/_______________________________________________________________________________/\n\n";
        cout << setw(30) << " "
             << " _______________________________________________________________________________\n";
        cout << setw(30) << " "
             << "/                                                                               /\n";
        cout << setw(30) << " "
             << "/                              1. LOGIN                                         /\n";
        cout << setw(30) << " "
             << "/                              2. REGISTER                                      /\n";
        cout << setw(30) << " "
             << "/                              3. FORGOT PASSWORD (or) USERNAME                 /\n";
        cout << setw(30) << " "
             << "/                              4. Exit                                          /\n";
        cout << setw(30) << " "
             << "/_______________________________________________________________________________/\n\n";

        cout << setw(30) << " "
             << "----> Enter your choice: ";
        cin >> choice;
        cout << endl;
        switch (choice)
        {
        case 1:
                login();
                break;
        case 2:
                registr();
                break;
        case 3:
                forgot();
                break;
        case 4:

                cout << setw(30) << " "
                     << "Thanks for using this program.\n\n\n\n";
                break;
        default:
                cout << setw(30) << " "
                     << "Enter a valid option!\n\n";
                cout << setw(30) << " ";
                system("pause");
                menu();
        }
}



void login()
{
        string user, u;
        char pass[100], p[100];
        system("cls");
        cout << endl
             << endl;
        cout << setw(30) << " "
             << "_______________________________________________________________________________\n";
        cout << setw(30) << " "
             << "|                                                                              |\n";
        cout << setw(30) << " "
             << "|                                Log in                                        |\n";
        cout << setw(30) << " "
             << "|______________________________________________________________________________|\n\n\n";
        cout << right << setw(35) << " "
             << "Enter email or username:  ";
        cin >> user;
        cout << endl
             << setw(35) << " "
             << "Enter Password:  ";
        ch = '\0';
        i = 0;
        while ((ch = _getch()) != '\r')
        {
                if (ch == '\b' && i > 0)
                {
                        cout << "\b \b";
                        i--;
                }
                else
                {
                        pass[i++] = ch;
                        cout << '*';
                }
        }
        cout << endl;
        pass[i] = '\0';


        ifstream input("database.txt");
        while (input >> u >> p)
        {
                if (u == user && strcmp(p, pass) == 0)

                {
                        count = 1;
                        system("cls");
                }
        }
        input.close();

        if (count == 1)
        {
                cout << endl
                     << endl;
                cout << setw(30) << " "
```

```cpp
                     << "________________________________________________________________________________\n";
                cout << setw(30) << " "
                     << "|                                                                              |\n";
                cout << setw(30) << " "
                     << "|                                DASHBOARD                                     |\n";
                cout << setw(30) << " "
                     << "|______________________________________________________________________________|\n\n\n";
                cout << setw(35) << " "
                     << "Hello! " << user << endl
                     << endl;
                cout << setw(35) << " "
                     << "[LOGIN SUCCESSFUL]\n";
                cout << setw(35) << " "
                     << "You are now logged in as " << user << endl
                     << endl
                     << endl;
                cout << setw(35) << " ";
                system("pause");
                main();
        }

        else
        {
                cout << endl
                     << setw(35) << " "
                     << "LOGIN ERROR\n";
                cout << setw(35) << " "
                     << "Please check your username and password\n\n\n";
                cout << setw(35) << " ";
                system("pause");
                menu();
        }
}

void registr()
{

        string reguser;
        char regpass[100], cp[100];
```

```
up:
        system("cls");
        cout << endl
             << endl;
        cout << setw(30) << " "
             << "________________________________________________________________________________\n";
        cout << setw(30) << " "
             << "|                                                                              |\n";
        cout << setw(30) << " "
             << "|                                Sign Up                                       |\n";
        cout << setw(30) << " "
             << "|______________________________________________________________________________|\n\n\n";
        cout << setw(35) << " "
             << "Enter email or username:   ";
        cin >> reguser;
        cout << endl
             << setw(35) << " "
             << "Enter Password:            ";
        ch = '\0';
        i = 0;


        while ((ch = _getch()) != '\r')
        {
                if (ch == '\b' && i > 0)
                {
                        cout << "\b \b";
                        i--;
                }
                else
                {
                        regpass[i++] = ch;
                        cout << '*';
                }
        }
        regpass[i] = '\0';
        cout << endl
             << setw(35) << " "
             << "Enter Confirm Password:    ";
        ch = '\0';
        i = 0;
        while ((ch = _getch()) != '\r')
        {
                if (ch == '\b' && i > 0)
                {
                        cout << "\b \b";
                        i--;
                }
                else
                {
                        cp[i++] = ch;
                        cout << '*';
                }
        }
        cp[i] = '\0';
        cout << endl;



        if (strcmp(regpass, cp) == 0) // 0 mtlb true
        {
                ofstream reg("database.txt", ios::app);
                reg << reguser << ' ' << regpass << endl;
                system("cls");
                cout << endl
                     << endl;
                cout << setw(30) << " "
                     << "________________________________________________________________________________\n";
                cout << setw(30) << " "
                     << "|                                                                              |\n";
                cout << setw(30) << " "
                     << "|                                Sign Up                                       |\n";
                cout << setw(30) << " "
                     << "|______________________________________________________________________________|\n\n\n";
                cout << setw(35) << " "
                     << "Registration Sucessful\n\n\n";
                cout << setw(35) << " ";
```

```cpp
                system("pause");
                menu();
        }
        else
        {
                cout << endl
                     << setw(35) << " "
                     << "Sign Up failed! Invalid passwords, Try again\n\n\n";
                cout << setw(35) << " ";
                system("pause");
                system("cls");
                goto up;
        }
}
void forgot()
{
        system("cls");
        cout << endl
             << endl;
        cout << setw(30) << " "
             << "________________________________________________________________________________\n";
        cout << setw(30) << " "
             << "|                                                                              |\n";
        cout << setw(30) << " "
             << "|                                Help Center                                   |\n";
        cout << setw(30) << " "
             << "|______________________________________________________________________________|\n\n\n";
        cout << setw(35) << " "
             << "Forgotten ? We're here for help\n";
        cout << setw(35) << " "
             << "Mail us: 22014198-060@uog.edu.pk\n";
        cout << setw(35) << " "
             << "Mail us: 22014198-049@uog.edu.pk\n";
        cout << setw(35) << " "
             << "Mail us: 22014198-044@uog.edu.pk\n";
        cout << setw(35) << " ";
        system("pause");
        menu();
}
```

Program Documentation

This document provides a comprehensive overview of the provided C++ source code, which implements a simple search engine application with user authentication and various functionalities. The code is structured into several functions, each serving a specific purpose within the program. Below is a detailed step-by-step explanation of each function and its role within the program.

1. project_page Function:

This function displays project information in a formatted manner. It showcases the names, roll numbers, and project details of the group members.

2. Edit_keyword Function:

This function allows users to edit keywords stored in the "keywords.txt" file.
It reads the keywords from the file and stores them in a vector.
The user is prompted to enter the keyword they wish to edit.
If the entered keyword exists in the vector, the user is prompted to provide a new keyword.
The old keyword is replaced with the new keyword in the vector.
The vector's contents are then written back to the "keywords.txt" file.

3. check_keywords Function:

This function reads and displays the list of keywords stored in the "keywords.txt" file.
Each keyword is printed on a separate line for easy readability.
4. add_keywords Function:

Users can input the number of keywords they want to add.
For each keyword, the user is prompted to input a keyword, and each keyword is written to the "keywords.txt" file on a separate line.
5. search Function:

Users are prompted to enter a keyword they want to search for.
The function opens the "keywords.txt" file in read mode.
It then searches for the entered keyword within the file, line by line.
If a line containing the keyword is found, it is printed on the screen.
6. search_engine_page Function:

This function serves as the main interface for the search engine application.
It presents users with a menu of options, including search, add keywords, view keywords, edit keywords, and log out.
Users input their choice, and the corresponding functionality is executed based on their selection.
The menu is looped until the user chooses to log out.

7. menu Function:

The main menu function displays the initial options for the program.
Users can choose to log in, register, recover a forgotten password, or exit the program.
The selected option directs users to the corresponding function's implementation.

8. login Function:

Users are prompted to input their username and password.
The function reads the username and password combinations from the "database.txt" file.
If a matching username and password are found, the user is logged in and directed to the search_engine_page function.

9. registr Function:

Users can register by providing a username and password.
The function prompts users to enter their desired username and password.
The password is confirmed by asking the user to re-enter it.
If the passwords match, the username and password are written to the "database.txt" file for future authentication.

10. forgot Function:

This function provides contact information for users who need help with forgotten passwords.
Users are directed to contact the specified email addresses for assistance.

11. main Function:

The program's entry point that initializes the application.
It checks the value of the count variable to determine whether a user is logged in or not.
If a user is logged in (count is 1), the search_engine_page function is called to provide access to search functionalities.
If no user is logged in (count is 0), the project_page function is displayed to showcase the project details, and then the main menu is shown using the menu function.
Overall Summary:

The provided C++ program implements a basic search engine application with user authentication, keyword management, and project information display. Users can log in, register, search for keywords, add keywords, view a list of keywords, edit keywords, and log out. The program showcases the use of functions to modularize different functionalities, and each function contributes to creating a user-friendly search engine experience.

Step-by-Step explanation of how the program works in sequence when executed

Main Function (main):

The program starts execution from the main function.
The count variable is checked. If count is 1, it means a user is logged in, so the search_engine_page function is called.
If count is 0, it means no user is logged in, so the project_page function is displayed followed by the menu function.
project_page Function:

This function displays formatted project details and group member information.
It presents a visually appealing representation of the project and contributors' names and roll numbers.
Users can press any key to continue, after which the menu function is called.

menu Function:

The main menu function is displayed, offering options like login, registration, forgotten password recovery, and program exit.
Users input their choice based on the provided options.
Depending on the choice, the relevant function is called:
If "1" is selected, the login function is called.
If "2" is selected, the registr function is called.
If "3" is selected, the forgot function is called.
If "4" is selected, a farewell message is displayed, and the program terminates.
login Function:
Users enter their username and password.
The function reads the "database.txt" file line by line, matching entered username and password combinations.
If a match is found, the count variable is set to 1, and the user is redirected to the search_engine_page function.
If no match is found, an error message is displayed, and users are redirected back to the menu function.

registr Function:

Users enter their desired username and password.
The password is confirmed by entering it again.
If the passwords match, the username and password are written to the "database.txt" file for future authentication.
If the passwords do not match, an error message is displayed, and the user is prompted to retry.

forgot Function:

Users are provided with contact information for assistance with forgotten passwords.
Email addresses are displayed for users to reach out for help.

search_engine_page Function:

Users who are logged in (count is 1) are directed to this function.
The function displays a menu with various options like search, adding keywords, viewing keywords, editing keywords, and logging out.
Users input their choice, and the corresponding functionality is executed.
After each action, users are brought back to the search_engine_page menu.

search Function:

Users input a keyword they want to search for.
The function searches for the keyword within the "keywords.txt" file.
If a match is found, the corresponding line (keyword) is printed on the screen.

add_keywords Function:

Users specify the number of keywords they want to add.
For each keyword, users input the keyword itself.
Keywords are appended to the "keywords.txt" file.

check_keywords Function:

This function reads the "keywords.txt" file and prints out all stored keywords line by line.

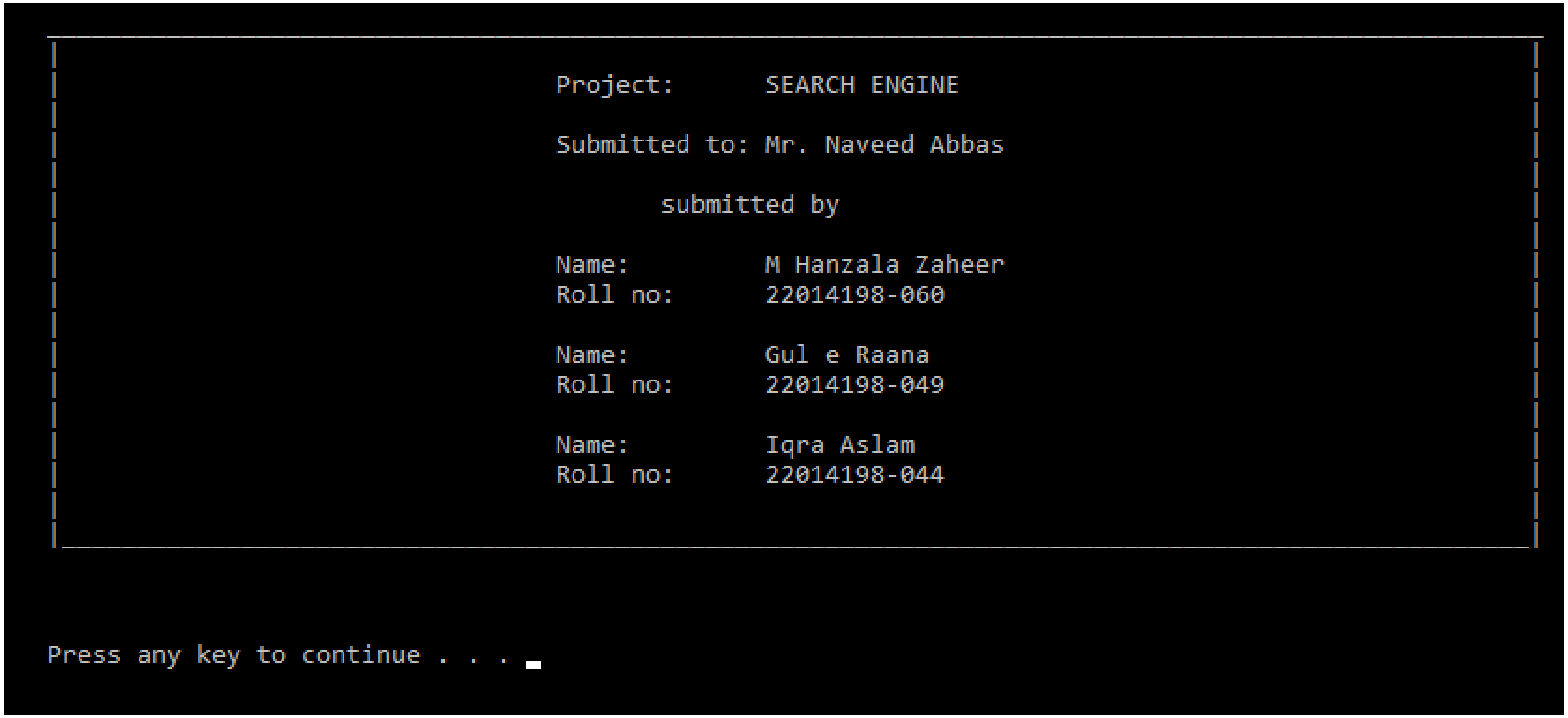
Edit_keyword Function:

Users input a keyword they wish to edit.
The function checks if the keyword exists.
If the keyword exists, users input a new keyword, which replaces the old keyword in the "keywords.txt" file.

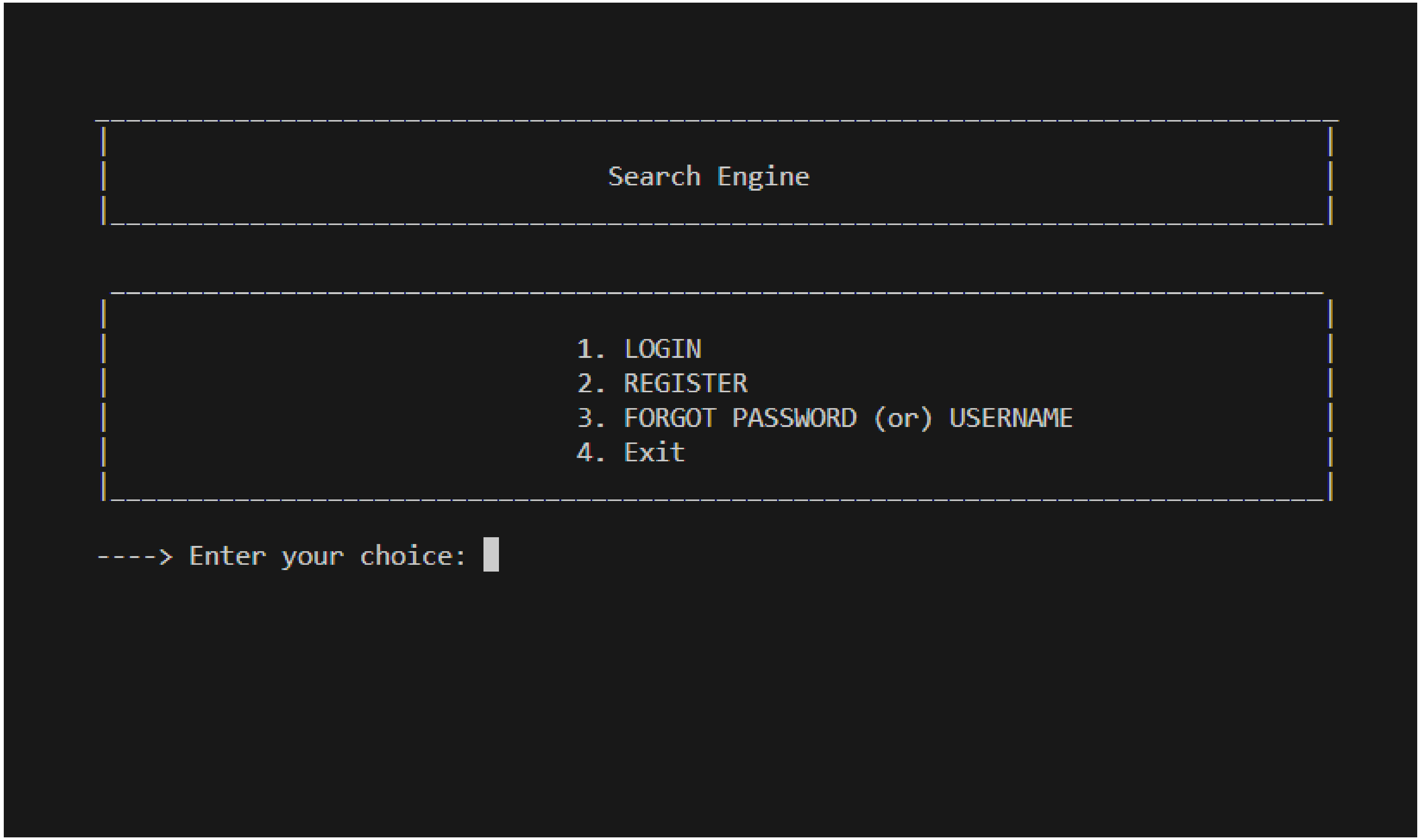
Logout and Continuation:

Upon logging out (count is set to 0), users are redirected back to the menu function.
Users can continue using the program by logging in again or performing other actions through the main menu.
Overall, the program provides an interactive and organized environment for users to interact with a simulated search engine, manage keywords, and navigate through various functionalities. It achieves this through a series of modularized functions, making the user experience seamless and intuitive.

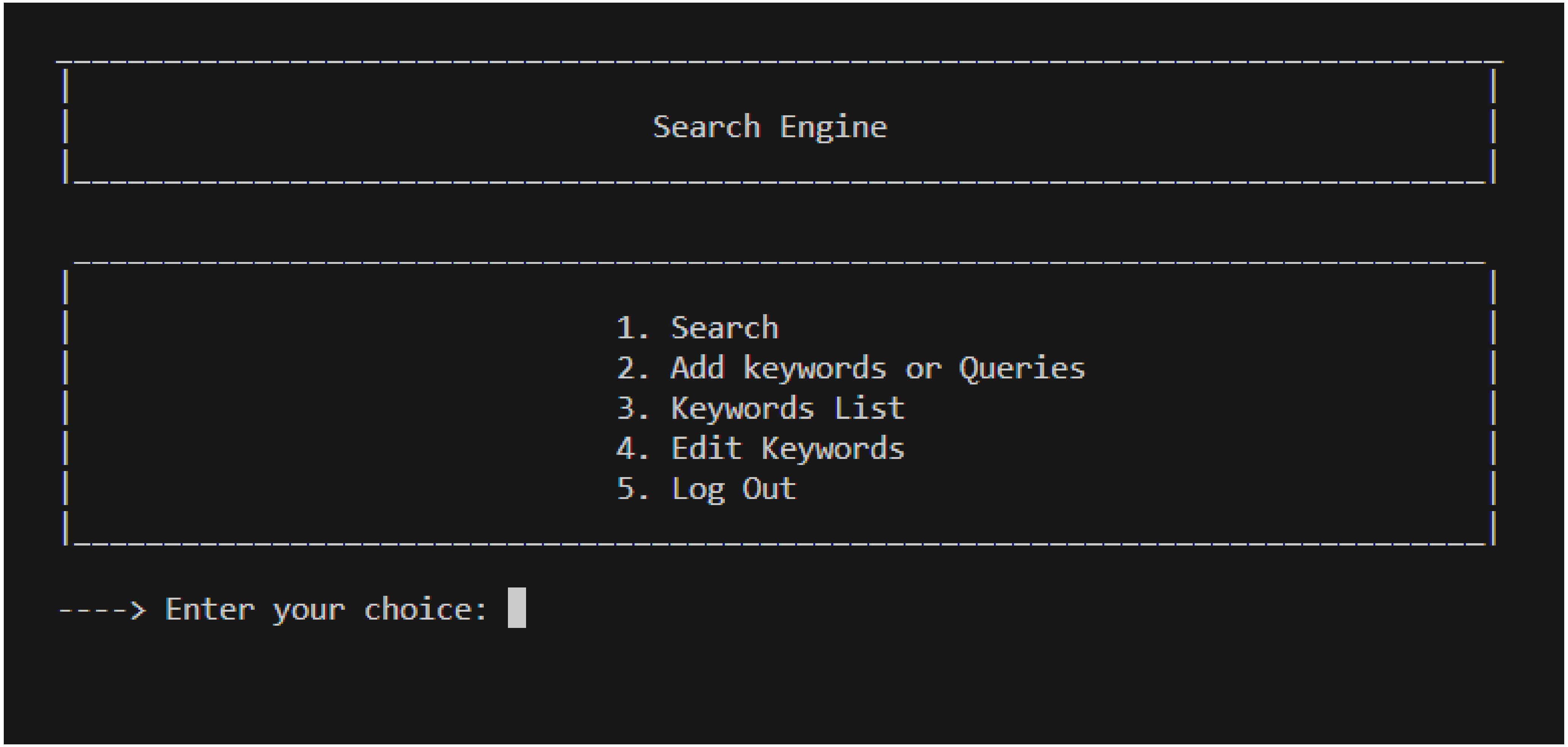
Different Interfaces of the Program



```
 ____________________________________________________________________________________________________
|                                                                                                    |
|                                  Project:      SEARCH ENGINE                                       |
|                                                                                                    |
|                                  Submitted to: Mr. Naveed Abbas                                    |
|                                                                                                    |
|                                         submitted by                                               |
|                                                                                                    |
|                                  Name:         M Hanzala Zaheer                                    |
|                                  Roll no:      22014198-060                                        |
|                                                                                                    |
|                                  Name:         Gul e Raana                                         |
|                                  Roll no:      22014198-049                                        |
|                                                                                                    |
|                                  Name:         Iqra Aslam                                          |
|                                  Roll no:      22014198-044                                        |
|                                                                                                    |
|____________________________________________________________________________________________________|



Press any key to continue . . .
```



```
 _______________________________________________________________________________
|                                                                               |
|                                 Search Engine                                 |
|_______________________________________________________________________________|

 _______________________________________________________________________________
|                                                                               |
|                              1. LOGIN                                         |
|                              2. REGISTER                                      |
|                              3. FORGOT PASSWORD (or) USERNAME                 |
|                              4. Exit                                          |
|_______________________________________________________________________________|

----> Enter your choice:
```



```
 ________________________________________________________________________________
|                                                                                |
|                                 Search Engine                                  |
|________________________________________________________________________________|


  ______________________________________________________________________________
|                                                                                |
|                              1. Search                                         |
|                              2. Add keywords or Queries                        |
|                              3. Keywords List                                  |
|                              4. Edit Keywords                                  |
|                              5. Log Out                                        |
|________________________________________________________________________________|

----> Enter your choice:
```