|  |  |
| --- | --- |
| C:\Users\Blaze\Pictures\vlcsnap-2017-01-19-16h08m14s556 (2).png  **Project:**  **Computer Science**  2016-17 Virtual Database Management: Emerald Blues | **About:**  The Emerald Blues is a database, aiming to provide people with ideas, with words, with a precious asset called knowledge. Tune in for more information…  **Virtual Database Management** |

**INDEX:**

1. CERTIFICATE

2. ACKNOWLEDGEMENT

3. INTRODUCTION

4. REQUIREMENTS:

5. OBJECTIVE

6. SOURCE CODE

7. SNAPSHOTS

8. ANALYSIS

9. FUTURE SCOPE

10. BIBLIOGRAPHY

****

**KENDRIYA VIDYALAYA NO.1, AFS I, JAMNAGAR GUJARAT**

**CERTIFICATE:**

This is to certify that Jatin of class XII have satisfactorily completed the project work in the subject of Computer Science, prescribed by the Central Board of Secondary Education (CBSE) for the year 2016-17.

**ACKNOWLEDGEMENTS:**

In the accomplishment of this project successfully, many people have

bestowed upon me their blessings and gracious support. It is time, I should

thank all the people who have been concerned with the project. Primarily I

would like to thank god for being with me throughout, in the presence of

whom, I’ve completed the assignment with success.

Then I would like to thank my Principal Mr. Arun Sharma and my

Computer Science mentor Mrs. Monika Verma, whose valuable guidance and

help has never made me feel that I am out of resources and with which, I

had anticipated the project as ‘attainable’. Their suggestions and

instructions has served as the major contributor towards the completion of

the project.

Finally, I would like to thank my parents and friends whose love, guidance

and valuable suggestions have been helpful in various phases of the

execution and completion of the project.

**REQUIREMENTS:**

* **HARDWARE:**

An Intel based Central Processing Unit capable of running any sort of windows operating system such as Pentium based workstation.

* Minimum 64 MB RAM (128 MB desirable) at server.
* Minimum 60 MB of free disk space for files.
* A CD ROM drive.
* Minimum 48 MB of RAM at workstation.
* VGA 15 color monitor for workstation.
* **SOFTWARE:**

The software requirements are as follows:

* Windows 98 or above.
* A C/C++ editor or IDE.
* Microsoft word.

**OBJECTIVE:**

The objective of this project is to create application software which can manage a virtual database, we named as the Emerald Blues.

The Emerald Blues, as said, is a database, aiming to provide people with ideas, with words, with a precious asset called knowledge. The database is comprised of fine arts of all time, from books to movies, it is a place to meet the finest people, came and lost in time, to learn from them, going through their books, passing through their movies. Basically, its place to find the ideas and 'emeralds' you desire.

What we planned was a computer based system which will serve as a journal for a user to keep track of all he is been reading or watching. What’s more, users will be scored on the basis of what they read or watch. The reason being to make learning fun and exciting or say desirable. As the user progresses, he will be levelled up from “Apprentice” to “Conformist” to “Saint”, representing level ‘1’, ‘2’ and ‘3’ respectively, ‘1’ being beginner and ‘3’ being Professional. The users will find themselves in the collection of a variety of books from nine different genres and one more special genre, with all books and movies selected with precise perfection and profound care. There is a lot more to Emerald Blues than could be written within a short word limit, of which the user will get to know with frequent use. Thank you.

**SOURCE CODE:**

#include<iostream>

#include<fstream>

#include<conio.h>

#include<string.h>

#include<process.h>

#include<stdio.h>

#include<ctype.h>

#include<stdlib.h>

#include <time.h>

using namespace std;

class Emerald

{

public:

char Genre[10][25],Books[10][10][100],Movies[10][10][100];

void Database();

};

void Emerald::Database()

{

strcpy(Books[0][0],"Don Quixote: Miguel De Cervantes");

strcpy(Books[0][1],"Ulysses: James Joyce");

strcpy(Books[0][2],"The Great Gatsby: F.Scott Fitzgerald");

strcpy(Books[0][3],"Catcher In The Rye: J.D. Salinger");

strcpy(Books[0][4],"Lolita: Vladimir Nabokov");

strcpy(Books[0][5],"To Kill A Mockingbird: Harper Lee");

strcpy(Books[0][6],"Pride And Prejudice: Jane Austen");

strcpy(Books[0][7],"The Iliad and The Odyssey: Homer");

strcpy(Books[0][8],"War And Peace: Leo Tolstoy");

strcpy(Books[0][9],"The Portrait Of A Lady: Henry James");

strcpy(Books[1][0],"The Lord Of The Rings: J. R. R. Tolkien");

strcpy(Books[1][1],"Harry Potter: J.K. Rowling");

strcpy(Books[1][2],"The Hunger Games: Suzanne Collins");

strcpy(Books[1][3],"Percy Jackson: Rick Riordan");

strcpy(Books[1][4],"Alice's Adventures In Wonderland: Lewis Carrol");

strcpy(Books[1][5],"Gulliver's Travels: Jonathan Swift");

strcpy(Books[1][6],"The Chronicles Of Narnia: C.S. Lewis");

strcpy(Books[1][7],"Game Of Thrones: George R.R. Martin");

strcpy(Books[1][8],"Alchemist: Paulo Coelho");

strcpy(Books[1][9],"The Three Musketeers: Alexandre Dumas");

strcpy(Books[2][0],"Frankenstein: Mary Shelley");

strcpy(Books[2][1],"It: Stephen King");

strcpy(Books[2][2],"Dracula: Bram Stoker");

strcpy(Books[2][3],"The Turn Of The Screw: Henry James");

strcpy(Books[2][4],"Strange Case Of Dr. Jekyll And Mr. Hyde: Robert

Louis Stevenson");

strcpy(Books[2][5],"Eisenheim The Illusionist: Steven Millhauser");

strcpy(Books[2][6],"Haunting Of Hill House: Shirley Jackson");

strcpy(Books[2][7],"The Shining: Stephen King");

strcpy(Books[2][8],"The Exorcist: William Peter Blatty");

strcpy(Books[2][9],"Heart Shaped Box: Joe Hill");

strcpy(Books[3][0],"The Scarlet Letter: Nathaniel Hawthorne");

strcpy(Books[3][1],"The Girl On The Train: Paula Hawkins");

strcpy(Books[3][2],"The Maltese Falcon: Dahiell Hammet");

strcpy(Books[3][3],"Murder On The Orient Express: Agatha Cristie");

strcpy(Books[3][4],"The Da Vinci Code: Dan Brown");

strcpy(Books[3][5],"Gone Girl: Gillian Flynn");

strcpy(Books[3][6],"The Count of Monte Cristo: Alexandre Dumas");

strcpy(Books[3][7],"The Woman In White: Wilkie Collins");

strcpy(Books[3][8],"The Big Sleep: Raymond Chandler");

strcpy(Books[3][9],"Sherlock Holmes: Sir Arthur Conan Doyle");

strcpy(Books[4][0],"As I Lay Dying: William Faulkner");

strcpy(Books[4][1],"Why Not Me?: Mindy Kaling");

strcpy(Books[4][2],"The Two Gentlemen Of Verona: William

Shakespeare");

strcpy(Books[4][3],"Three Men In A Boat: Jerome K. Jerome");

strcpy(Books[4][4],"Pilgrim's Progress: John Bunyan");

strcpy(Books[4][5],"Animal Farm: George Orwell");

strcpy(Books[4][6],"Gone With The Wind: Margaret Mitchell");

strcpy(Books[4][7],"Dr. Zhivago: Boris Pasternak");

strcpy(Books[4][8],"Tess Of The D'Urbervilles: Thomas Hardy");

strcpy(Books[4][9],"The Twilight Saga: Stephenie Meyer");

strcpy(Books[5][0],"The Time Machine: H.G. Wells");

strcpy(Books[5][1],"Golden Sun: Pierce Brown");

strcpy(Books[5][2],"1984: George Orwell");

strcpy(Books[5][3],"Twenty Thousand Leagues Under The Sea: Jules

Verne");

strcpy(Books[5][4],"Brave New World: Aldous Huxley");

strcpy(Books[5][5],"Dune: Frank Herbert");

strcpy(Books[5][6],"The Forever War: Joe Haldeman");

strcpy(Books[5][7],"The Book Of The New Son: Gene Wolfe");

strcpy(Books[5][8],"Hyperion: Dan Simmons");

strcpy(Books[5][9],"Neuromamcer: William Gibson");

strcpy(Books[6][0],"The Perks Of Being A Wallflower: Stephen Chbosky");

strcpy(Books[6][1],"The Giver: Lois Lowry");

strcpy(Books[6][2],"The Divergent: Veronica Roth");

strcpy(Books[6][3],"The Maze Runner: James Dashner");

strcpy(Books[6][4],"The Martian: Andy Weir");

strcpy(Books[6][5],"Fahrenheit 451: Ray Bradbury");

strcpy(Books[6][6],"Ender's Game: Orson Scott Card");

strcpy(Books[6][7],"The Princess Bride: William Goldman");

strcpy(Books[6][8],"The Fault In Our Stars: John Green");

strcpy(Books[6][9],"The Monk Who Sold His Ferrari: Robin Sharma");

strcpy(Books[7][0],"Clarissa: Samuel Richardson");

strcpy(Books[7][1],"Emma: Jane Austen");

strcpy(Books[7][2],"The Danish Girl: David Ebershoff");

strcpy(Books[7][3],"A Midsummer Night's Dream: William Shakespeare");

strcpy(Books[7][4],"The Curious Case Of Benjamin Button: F. Scott

Fitzgerald");

strcpy(Books[7][5],"Far From The Madding Crowd: Thomas Hardy");

strcpy(Books[7][6],"Anna Karenina: Leo Tolstoy");

strcpy(Books[7][7],"The Portrait Of An Artist As A Young Man: James

Joyce");

strcpy(Books[7][8],"The Pursuit Of Love: Nancy Mitford");

strcpy(Books[7][9],"Crimson Peak: Edith M. Cushing");

strcpy(Books[8][0],"The Diary Of A Young Girl: Anne Frank");

strcpy(Books[8][1],"A Brief History Of Time: Stephen Hawking");

strcpy(Books[8][2],"Silent Spring: Rachel Carson");

strcpy(Books[8][3],"Homage To Catalonia: George Orwell");

strcpy(Books[8][4],"On The Origin Of Species: Charles Darwin");

strcpy(Books[8][5],"The Sixth Extinction: Elizabeth Kolbert");

strcpy(Books[8][6],"The Republic: Plato");

strcpy(Books[8][7],"Principia Mathematica: Issac Newton");

strcpy(Books[8][8],"Dialogue Concerning The Two Chief World Systems:

Galileo Galilei");

strcpy(Books[8][9],"The Second World War: Winston Churchill");

strcpy(Books[9][0],"Ulysses: James Joyce");

strcpy(Books[9][1],"Don Quixote: Miguel De Cervantes");

strcpy(Books[9][2],"To Kill A Mockingbird: Harper Lee");

strcpy(Books[9][3],"Alchemist: Paulo Coelho");

strcpy(Books[9][4],"Dracula: Bram Stoker");

strcpy(Books[9][5],"Dune: Frank Herbert");

strcpy(Books[9][6],"The Divergent: Veronica Roth");

strcpy(Books[9][7],"Animal Farm: George Orwell");

strcpy(Books[9][8],"Anna Karenina: Leo Tolstoy");

strcpy(Books[9][9],"Dialogue Concerning The Two Chief World Systems:

Galileo Galilei");

strcpy(Movies[0][0],"The Shawshank Redemption (1994)");

strcpy(Movies[0][1],"The Godfather (1972)");

strcpy(Movies[0][2],"12 Angry Men (1957)");

strcpy(Movies[0][3],"Schindler's List (1993)");

strcpy(Movies[0][4],"Pulp Fiction (1994)");

strcpy(Movies[0][5],"The Good, The Bad And The Ugly (1966)");

strcpy(Movies[0][6],"Forrest Gump (1994)");

strcpy(Movies[0][7],"Saving Private Ryan (1998)");

strcpy(Movies[0][8],"The Green Mile (1999)");

strcpy(Movies[0][9],"Gladiator (2000)");

strcpy(Movies[1][0],"The Lord Of The Rings Trilogy (2001-2003)");

strcpy(Movies[1][1],"The Treasure Of The Sierra Madre (1948)");

strcpy(Movies[1][2],"Harry Potter And The Deathly Hallow - Part II

(2011)");

strcpy(Movies[1][3],"Indiana Jones and the Last Crusade (1989)");

strcpy(Movies[1][4],"The Wizard Of Oz (1939)");

strcpy(Movies[1][5],"Spirited Away (2001)");

strcpy(Movies[1][6],"Inside Out (2015)");

strcpy(Movies[1][7],"Nausicaä Of The Valley Of The Wind (1984)");

strcpy(Movies[1][8],"Monsters, Inc. (2001)");

strcpy(Movies[1][9],"Toy Story 3 (2010)");

strcpy(Movies[2][0],"The Silence Of The Lambs (1991)");

strcpy(Movies[2][1],"Alien (1979)");

strcpy(Movies[2][2],"The Shining (1980)");

strcpy(Movies[2][3],"The Thing (1982)");

strcpy(Movies[2][4],"Jaws (1975)");

strcpy(Movies[2][5],"Sinister (2012)");

strcpy(Movies[2][6],"Saw(2004)");

strcpy(Movies[2][7],"The Prestige (2006)");

strcpy(Movies[2][8],"The Illusionist (2005)");

strcpy(Movies[2][9],"Now You See Me (2012)");

strcpy(Movies[3][0],"Fight Club (1999)");

strcpy(Movies[3][1],"One Flew Over the Cuckoo's Nest (1975)");

strcpy(Movies[3][2],"Se7en (1995)");

strcpy(Movies[3][3],"Psycho (1960)");

strcpy(Movies[3][4],"American History X (1998)");

strcpy(Movies[3][5],"Vertigo (1958)");

strcpy(Movies[3][6],"Eternal Sunshine of the Spotless Mind (2004)");

strcpy(Movies[3][7],"L.A. Confidential (1997)");

strcpy(Movies[3][8],"V for Vendetta (2005)");

strcpy(Movies[3][9],"Donnie Darko (2001)");

strcpy(Movies[4][0],"Goodfellas (1990)");

strcpy(Movies[4][1],"It's a Wonderful Life (1946)");

strcpy(Movies[4][2],"Life Is Beautiful (1997)");

strcpy(Movies[4][3],"Casablanca (1942)");

strcpy(Movies[4][4],"City Lights (1931)");

strcpy(Movies[4][5],"Shakespeare In Love (1998)");

strcpy(Movies[4][6],"Back to the Future (1985)");

strcpy(Movies[4][7],"Singin' in the Rain (1952)");

strcpy(Movies[4][8],"Some Like It Hot (1959)");

strcpy(Movies[4][9],"Before Sunrise (1995)");

strcpy(Movies[5][0],"The Dark Knight (2008)");

strcpy(Movies[5][1],"Inception (2010)");

strcpy(Movies[5][2],"The Matrix (1999)");

strcpy(Movies[5][3],"Star Wars: Episode V - The Empire Strikes Back

(1980)");

strcpy(Movies[5][4],"Interstellar (2014)");

strcpy(Movies[5][5],"Terminator 2: Judgment Day (1991)");

strcpy(Movies[5][6],"2001: A Space Odyssey (1968)");

strcpy(Movies[5][7],"Metropolis (1927)");

strcpy(Movies[5][8],"Blade Runner (1982)");

strcpy(Movies[5][9],"Twelve Monkeys (1995)");

strcpy(Movies[6][0],"Divergent (2014)");

strcpy(Movies[6][1],"Gone Girl (2014)");

strcpy(Movies[6][2],"3 Idiots (2009)");

strcpy(Movies[6][3],"The Maze Runner (2014)");

strcpy(Movies[6][4],"The Fault In Our Stars (2014)");

strcpy(Movies[6][5],"The Spectacular Now (2013)");

strcpy(Movies[6][6],"Predestination");

strcpy(Movies[6][7],"Submarine (2010)");

strcpy(Movies[6][8],"Project Almanac (2014)");

strcpy(Movies[6][9],"Big Hero 6 (2014)");

strcpy(Movies[7][0],"Letters To Juliet (2010)");

strcpy(Movies[7][1],"Good Will Hunting (1997)");

strcpy(Movies[7][2],"Dead Poets Society (1989)");

strcpy(Movies[7][3],"About Time (2013)");

strcpy(Movies[7][4],"A Walk To Remember (2002)");

strcpy(Movies[7][5],"Boyhood (2015)");

strcpy(Movies[7][6],"Inkheart (2008)");

strcpy(Movies[7][7],"The Perks Of Being A Wallflower (2012)");

strcpy(Movies[7][8],"Still Alice (2014)");

strcpy(Movies[7][9],"The Devil Wears Prada (2006)");

strcpy(Movies[8][0],"The Imitation Game (2014)");

strcpy(Movies[8][1],"Catch Me If You Can (2002)");

strcpy(Movies[8][2],"The Social Network (2010)");

strcpy(Movies[8][3],"Spotlight (2015)");

strcpy(Movies[8][4],"A Beautiful Mind (2001)");

strcpy(Movies[8][5],"Bridge Of Spies (2015)");

strcpy(Movies[8][6],"Pain & Gain (2013)");

strcpy(Movies[8][7],"Steve Jobs (2015)");

strcpy(Movies[8][8],"The Theory Of Everything (2014)");

strcpy(Movies[8][9],"The Wolf Of Wall Street (2013)");

strcpy(Movies[9][0],"The Shawshank Redemption (1994)");

strcpy(Movies[9][1],"The Green Mile (1999)");

strcpy(Movies[9][2],"Forrest Gump (1994)");

strcpy(Movies[9][3],"The Prestige (2006)");

strcpy(Movies[9][4],"Intersteller (2014)");

strcpy(Movies[9][5],"The Perks Of Being A Wallflower (2012)");

strcpy(Movies[9][6],"Fight Club (1999)");

strcpy(Movies[9][7],"Boyhood (2014)");

strcpy(Movies[9][8],"The Lord Of The Rings Trilogy (2001-2003)");

strcpy(Movies[9][9],"The Imitation Game (2014)");

strcpy(Genre[0],"Classic");

strcpy(Genre[1],"Adventure/Fantasy/Children");

strcpy(Genre[2],"Horror/Cult");

strcpy(Genre[3],"Suspense/Thriller/Crime");

strcpy(Genre[4],"Comedy/Romance/Allegory");

strcpy(Genre[5],"Science Fiction");

strcpy(Genre[6],"Young Adult");

strcpy(Genre[7],"Feel Good/Drama");

strcpy(Genre[8],"Non-Fiction");

strcpy(Genre[9],"Best Of Emerald");

}

class user

{

public:

char name[50],pass[18],status[25],bks[100][100],movs[100][100];

int EmeraldPoints,totmov,totbks;

void showdata()

{

time\_t gen;

unsigned int seedval;

seedval=(unsigned) time(&gen);

srand(seedval);

cout<<"Name:"<<name<<endl;

cout<<"Status:"<<status<<endl;

cout<<"Emerald Points:"<<EmeraldPoints<<endl;

int Number=(rand()%5)+1;

switch(Number)

{

case 1: cout<<"The Emerald Blues is pleasured in your

presence, "<<name<<"."<<endl;

break;

case 2: cout<<"We're always by your side,

"<<name<<"."<<endl;

break;

case 3: cout<<"Art is all you desire,

"<<name<<"."<<endl;

break;

case 4: cout<<"Seek and ye' shall find,

"<<name<<"."<<endl;

break;

case 5: cout<<"Feel infinite in the world of words and arts,

"<<name<<"."<<endl;

break;

}

}

void calcEP()

{

EmeraldPoints=(totbks\*1000+totmov\*100)/4;

}

void calcStatus()

{

if(EmeraldPoints>=10000)

strcpy(status,"Saint");

else if(EmeraldPoints>=1000 and EmeraldPoints<10000)

strcpy(status,"Conformist");

else

strcpy(status,"Apprentice");

}

};

class userlist

{

public:

int no;

char users[100][50];

};

void showbooklist();

void showmovielist();

void chkuserlist();

void showtitle();

void showacc();

void showch();

void showerr();

void showem();

void showin();

char gotoEM();

char gotomain();

int crtusr();

void setdefault(user &);

void namecutter(char name[]);

void Nameit(char\*);

int openacc();

void delay(float);

int main()

{

chkuserlist();

Emerald Data;

Data.Database();

mpage:

system("cls");

system("Color 50");

int i,ch,ch1,ch2;

char ch3;

for (i=1;i<=5;i++)

{

system("cls");

delay(0.1);

cout<<"Welcome to The Emerald Blue Database of Fine

Arts..."<<endl<<endl;

delay(0.1);

}

cout<<"A place to find the ideas and 'emeralds' you

desire"<<endl<<endl;

cout<<"1.Sneak Peak: Read and Watch Without An

Account."<<endl;

cout<<"2.Sign In Into the World of Art and Literature."<<endl;

cout<<"3.Create An Account."<<endl;

cout<<"4.What's New?"<<endl;

cout<<"5.About."<<endl;

cout<<"6.Exit."<<endl;

showch();

cin>>ch;

switch(ch)

{

case 1:

{

sp:

system("Color 40");

showtitle();

cout<<"Sneak Peak:"<<endl<<endl;

cout<<"Here you can watch the list of books and

movies present in "

<<"the database. However, to access them, you

must sign in."<<endl<<endl;

cout<<"1.The list of books."<<endl;

cout<<"2.The list of movies."<<endl;

showch();

cin>>ch1;

switch(ch1)

{

case 1:

{

gen1page:

system("Color F4");

showtitle();

int i;

cout<<"Books:"<<endl;

cout<<"Select Genre:"<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Genre[i]<<endl;

}

showch();

cin>>ch2;

switch(ch2)

{

case 1:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[0]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[0][i]<<endl;

}

}

break;

case 2:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[1]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[1][i]<<endl;

}

}

break;

case 3:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[2]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[2][i]<<endl;

}

}

break;

case 4:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[3]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[3][i]<<endl;

}

}

break;

case 5:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[4]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[4][i]<<endl;

}

}

break;

case 6:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[5]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[5][i]<<endl;

}

}

break;

case 7:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[6]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[6][i]<<endl;

}

}

break;

case 8:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[7]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[7][i]<<endl;

}

}

break;

case 9:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[8]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[8][i]<<endl;

}

}

break;

case 10:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[9]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Books[9][i]<<endl;

}

}

break;

default:

{

goto gen1page;

showerr();

}

break;

}

}

break;

case 2:

{

gen2page:

system("Color F4");

showtitle();

int i;

cout<<"Movies:"<<endl;

cout<<"Select Genre:"<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Genre[i]<<endl;

}

showch();

cin>>ch2;

switch(ch2)

{

case 1:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[0]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[0][i]<<endl;

}

}

break;

case 2:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[1]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[1][i]<<endl;

}

}

break;

case 3:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[2]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[2][i]<<endl;

}

}

break;

case 4:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[3]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[3][i]<<endl;

}

}

break;

case 5:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[4]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[4][i]<<endl;

}

}

break;

case 6:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[5]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[5][i]<<endl;

}

}

break;

case 7:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[6]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[6][i]<<endl;

}

}

break;

case 8:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[7]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[7][i]<<endl;

}

}

break;

case 9:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[8]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[8][i]<<endl;

}

}

break;

case 10:

{

showtitle();

int i;

cout<<"Genre:"<<Data.Genre[9]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Data.Movies[9][i]<<endl;

}

}

break;

default:

{

goto gen2page;

showerr();

}

break;

}

}

break;

default:

{

goto sp;

showerr();

}

break;

}

ch3=gotomain();

if(ch3=='y' or ch3=='Y')

goto mpage;

}

break;

case 2:

{

int flag=0;

flag=openacc();

if(flag==0)

{

ch3=gotomain();

if(ch3=='y' or ch3=='Y')

goto mpage;

}

}

break;

case 3:

{

int flag=0;

flag=crtusr();

if(flag==0)

{

ch3=gotomain();

if(ch3=='y' or ch3=='Y')

goto mpage;

}

}

break;

case 4:

{

system("Color E4");

showtitle();

cout<<"What's New?:"<<endl<<endl;

cout<<"Find out here what's new is going on around

the world of art "

<<"and literature, and your own Emerald

Blues."<<endl<<endl;

time\_t gen;

unsigned int seedval;

seedval=(unsigned) time(&gen);

srand(seedval);

int Number=(rand()%5)+1;

switch(Number)

{

case 1: cout<<"John Green Is The New Teen Icon

For Young Adult -

2014!"<<endl<<endl;

case 2: cout<<"One Of The Best Classic Young

Adult: The Outsiders by S.E. Hinton

- 2017!"<<endl<<endl;

case 3: cout<<"Watch The Shawshank Redemption

(1994), Streaming At Emerald

Blues!"<<endl<<endl;

case 4: cout<<"The Trailer Of Christopher Nolan's

Much Awaited Movie: Dunkirk Has

Been Out - 2016!"<<endl<<endl;

case 5: cout<<"The Academy Awards Are Arriving

Yet Again, In February 2017. Keep

Waiting For The Fun It Will

Bring."<<endl<<endl;

break;

}

ch3=gotomain();

if(ch3=='y' or ch3=='Y')

goto mpage;

}

break;

case 5:

{

system("Color 34");

showtitle();

cout<<"About:"<<endl<<endl;

cout<<"The Emerald Blues is a database, aiming to

provide people with ideas, with words, "

<<"with a precious asset called knowledge. The

database is comprised of fine arts of "

<<"all time, from books to movies, it is a place to

meet the finest people, came and "

<<"lost in time, to learn from them, going through

their books, passing through their"

<<" movies. Basically, its place to find the ideas

and 'emeralds' you desire. So, let"

<<" yourself sign in into the world of ideas and

witness a platform, one that ranks "

<<"you on the basis of your reading and

watching skills, on your desire of learning. "

<<"There are three statuses, which will describe

you as a user, based on your Emerald"

<<" Points or EPs. The database is created by

Jatin of class -"

<<" XII who're working on a bigger project,

linking this database to the web server. "

<<" You'll get to know more about Emerald Blues

with frequent use. Thank you friend..."

<<endl;

ch3=gotomain();

if(ch3=='y' or ch3=='Y')

goto mpage;

}

break;

case 6:

{

cout<<"\aThank you for using Emerald

Blues...\a"<<endl;

exit(0);

}

break;

default:

{

goto mpage;

showerr();

}

break;

}

}

void chkuserlist()

{

ifstream usrlst;

usrlst.open("Userlist.DAT",ios::in);

if(!usrlst.good())

{

ofstream fo("Userlist.DAT",ios::out);

userlist list;

list.no=0;

fo.write((char \*) &list,sizeof(list));

fo.close();

}

}

int crtusr()

{

showtitle();

cout<<"Create An Account:"<<endl<<endl;

char name[50],reply;

ifstream fi;

user u; userlist list;

fi.open("Userlist.DAT");

fi.read((char \*) &list,sizeof(list));

fi.close();

if(list.no==99)

{

cout<<"You can't create any account."<<endl;

cout<<"Since the account list is full, "

<<"so, an account is need to be deleted."

<<"We're extremely sorry for the inconvenience."<<endl<<endl;

return 0;

}

cin.get();

cout<<"Enter your name (Max 49 Characters):"<<endl;

page1:

cin.getline(name,50);

Nameit(name);

if(name[0]=='\0'||name[0]==' ')

{

showacc();

cout<<"\aPlease enter a valid name:"<<endl;

goto page1;

}

for(int i=0;name[i]!='\0';i++)

{

if(name[i]=='\\'||name[i]=='/'||name[i]==':'||name[i]=='\*'||name[i]=='?'||name[i]=='\"'||name[i]=='<'||name[i]=='>'||name[i]=='|')

{

showacc();

cout<<"\aPlease enter a valid name:"<<endl;

goto page1;

}

}

fi.open(name);

if(fi.good())

{

showacc();

cout<<"\aThe name you have entered is already present!"<<endl;

cout<<"Please enter another name:"<<endl;

goto page1;

}

fi.close();

strcpy(u.name,name);

page2:

showacc();

cout<<"Do you want to add a password (Y/N)?";

cin>>reply;

if(reply=='y' or reply=='Y')

{

cin.get();

showacc();

cout<<"Enter your password (Max 17 Characters):"<<endl;

cin.getline(u.pass,18);

}

else if(reply=='n' or reply=='N')

{

strcpy(u.pass,"\*\*\*\*\*\*\*\*");

}

else

{

showerr();

goto page2;

}

setdefault(u);

ofstream fo(name);

fo.write((char \*) &u,sizeof(u));

fo.close();

strcpy(list.users[list.no],name);

list.no++;

fo.open("Userlist.DAT");

fo.write((char \*) &list,sizeof(list));

fo.close();

showacc();

cout<<"\aCongratulations!\a"<<endl<<endl;

cout<<"You have created your account of name:\t'"<<name<<"'"<<endl;

return 0;

}

void setdefault(user &usr)

{

int i,j;

for(i=0;i<100;i++)

{

for(j=0;j<100;j++)

{

usr.bks[i][j]='\0';

usr.movs[i][j]='\0';

}

}

usr.totmov=0;

usr.totbks=0;

usr.calcEP();

usr.calcStatus();

}

void showerr()

{

cout<<"You have entered an invalid choice, please enter

again."<<endl;

}

void showtitle()

{

system("cls");

cout<<"The Emerald Blue Database of Fine Arts"<<endl<<endl;

}

void showacc()

{

system("cls");

showtitle();

cout<<"Create An Account:"<<endl<<endl;

}

void showin()

{

system("cls");

showtitle();

cout<<"Sign In Into the World of Art and Literature:"<<endl<<endl;

}

int openacc()

{

user u;

userlist list;

char name[50],password[20],reply,choice;

int sel;

ofstream fo;

page3:

cin.get();

showin();

cout<<"Enter your account name:"<<endl;

cin.getline(name,50);

Nameit(name);

ifstream fi(name);

if(!fi.good())

{ pagename:

showin();

cout<<"\aInvalid account name!"<<endl<<"Enter again (Y/N)?"<<endl;

cin>>reply;

if(reply=='y'||reply=='Y')

goto page3;

else if(reply=='n'||reply=='N')

return 0;

else

goto pagename;

}

fi.read((char \*) &u,sizeof(u));

fi.close();

fi.open("Userlist.DAT");

fi.read((char \*) &list,sizeof(list));

fi.close();

int flag=0;

for(int s=0;s<list.no;s++)

{

if(!strcmp(name,list.users[s]))

{ flag=1; break; }

}

if(flag==0)

{

strcpy(list.users[list.no],name);

list.no++;

fo.open("Userlist.DAT",ios::out);

fo.write((char \*) &list,sizeof(list));

fo.close();

}

if(strcmp(u.pass,"\*\*\*\*\*\*\*\*")==0)

goto page5;

page4:

cout<<"Press enter to continue..."<<endl;

cin.get();

showin();

cout<<"Enter your password:"<<endl;

cin.getline(password,18);

if(strcmp(u.pass,password)!=0)

{

pagepass:

cout<<"\aWrong password!"<<endl<<"Enter again (Y/N)?"<<endl;

cin>>reply;

if(reply=='y'||reply=='Y')

goto page4;

else if(reply=='N'||reply=='n')

return 0;

else

goto pagepass;

}

page5:

showem();

system("Color 30");

cout<<"\aWelcome "<<name<<endl<<endl;

u.showdata();

cout<<endl;

cout<<"1.My Status."<<endl;

cout<<"2.Emerald Points."<<endl;

cout<<"3.No. and Names Of Books Read, Movies Watched."<<endl;

cout<<"4.Read A Book."<<endl;

cout<<"5.Watch A Movie."<<endl;

cout<<"6.Add/Modify Your Password."<<endl;

cout<<"7.Remove Your Password."<<endl;

cout<<"8.Delete Your Account."<<endl;

cout<<"9.Log Out."<<endl;

showch();

cin>>sel;

switch(sel)

{

case 1:

{

showem();

system("Color 34");

cout<<"My Status:"<<endl<<endl;

cout<<name<<", you're a "<<u.status<<" by

status."<<endl;

if(strcmp(u.status,"Apprentice")==0)

cout<<"Earn "<<(1000-u.EmeraldPoints)<<" points to

achieve the 'Conformist' status :)"<<endl;

else if(strcmp(u.status,"Conformist")==0)

cout<<"Earn "<<(10000-u.EmeraldPoints)<<" points to

achieve the 'Saint' status :)"<<endl;

else if(strcmp(u.status,"Saint")==0)

cout<<"We're enlightened in your presence :)"<<endl;

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

break;

case 2:

{

showem();

system("Color 34");

cout<<"Emerald Points:"<<endl<<endl;

cout<<name<<", you've "<<u.EmeraldPoints<<" Emerald

Points. Read/Watch more to score more EPs :)"<<endl;

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

break;

case 3:

{

Emerald Dat;

Dat.Database();

showem();

system("Color 34");

int i,j,k,flag=0,flag2=0;

char gen[25];

cout<<"No. and Names Of Books Read, Movies

Watched:"<<endl<<endl;

cout<<"No. of books read:"<<u.totbks<<endl<<endl;

cout<<"Books read:"<<endl;

for(i=0;i<u.totbks;i++)

{

for(j=0;j<10;j++)

{

for(k=0;k<10;k++)

{

if(strcmp(u.bks[i],Dat.Books[j][k])==0)

{

strcpy(gen,Dat.Genre[j]);

flag=1;

break;

}

}

if(flag==1)

break;

}

cout<<(i+1)<<"."<<u.bks[i]<<", Genre:"<<gen<<endl;

}

cout<<"No. of movies watched:"<<u.totmov<<endl<<endl;

cout<<"Movies watched:"<<endl;

for(i=0;i<u.totmov;i++)

{

for(j=0;j<10;j++)

{

for(k=0;k<10;k++)

{

if(strcmp(u.movs[i],Dat.Movies[j][k])==0)

{

strcpy(gen,Dat.Genre[j]);

flag2=1;

break;

}

}

if(flag2==1)

break;

}

cout<<(i+1)<<"."<<u.movs[i]<<",

Genre:"<<gen<<endl;

}

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

break;

case 4:

{

Emerald Dat;

Dat.Database();

bkspage:

system("Color 34");

showem();

int ch,ch1,i;

cout<<"Read A Book:"<<endl<<endl;

cout<<"Select Genre:"<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Genre[i]<<endl;

}

showch();

cin>>ch;

switch(ch)

{

case 1:

{

showem();

cout<<"Genre:"<<Dat.Genre[0]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[0][i]<<endl;

}

}

break;

case 2:

{

showem();

cout<<"Genre:"<<Dat.Genre[1]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[1][i]<<endl;

}

}

break;

case 3:

{

showem();

cout<<"Genre:"<<Dat.Genre[2]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[2][i]<<endl;

}

}

break;

case 4:

{

showem();

cout<<"Genre:"<<Dat.Genre[3]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[3][i]<<endl;

}

}

break;

case 5:

{

showem();

cout<<"Genre:"<<Dat.Genre[4]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[4][i]<<endl;

}

}

break;

case 6:

{

showem();

cout<<"Genre:"<<Dat.Genre[5]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[5][i]<<endl;

}

}

break;

case 7:

{

showem();

cout<<"Genre:"<<Dat.Genre[6]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[6][i]<<endl;

}

}

break;

case 8:

{

showem();

cout<<"Genre:"<<Dat.Genre[7]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[7][i]<<endl;

}

}

break;

case 9:

{

showem();

cout<<"Genre:"<<Dat.Genre[8]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[8][i]<<endl;

}

}

break;

case 10:

{

showem();

cout<<"Genre:"<<Dat.Genre[9]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Books[9][i]<<endl;

}

}

break;

default:

{

goto bkspage;

showerr();

}

break;

}

showch();

cin>>ch1;

if(ch1>=1 and ch1<=10)

{

u.totbks++;

strcpy(u.bks[(u.totbks-1)],Dat.Books[(ch-1)][(ch1-1)]);

u.calcEP();

u.calcStatus();

fo.open(name,ios::out);

fo.write((char \*) &u,sizeof(u));

fo.close();

delay(1.0);

cout<<"\a\nCongratulations, "<<name<<", you've read

"<<u.bks[(u.totbks-1)]<<" :)"<<endl<<endl;

cout<<"\aYou've earned 1000 Emerald Points!"<<endl;

}

else

{

showerr();

goto bkspage;

}

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

break;

case 5:

{

Emerald Dat;

Dat.Database();

movpage:

system("Color 34");

showem();

int ch,ch1,i;

cout<<"Watch A Movie:"<<endl<<endl;

cout<<"Select Genre:"<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Genre[i]<<endl;

}

showch();

cin>>ch;

switch(ch)

{

case 1:

{

showem();

cout<<"Genre:"<<Dat.Genre[0]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[0][i]<<endl;

}

}

break;

case 2:

{

showem();

cout<<"Genre:"<<Dat.Genre[1]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[1][i]<<endl;

}

}

break;

case 3:

{

showem();

cout<<"Genre:"<<Dat.Genre[2]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[2][i]<<endl;

}

}

break;

case 4:

{

showem();

cout<<"Genre:"<<Dat.Genre[3]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[3][i]<<endl;

}

}

break;

case 5:

{

showem();

cout<<"Genre:"<<Dat.Genre[4]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[4][i]<<endl;

}

}

break;

case 6:

{

showem();

cout<<"Genre:"<<Dat.Genre[5]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[5][i]<<endl;

}

}

break;

case 7:

{

showem();

cout<<"Genre:"<<Dat.Genre[6]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[6][i]<<endl;

}

}

break;

case 8:

{

showem();

cout<<"Genre:"<<Dat.Genre[7]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[7][i]<<endl;

}

}

break;

case 9:

{

showem();

cout<<"Genre:"<<Dat.Genre[8]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[8][i]<<endl;

}

}

break;

case 10:

{

showem();

cout<<"Genre:"<<Dat.Genre[9]<<endl;

for(i=0;i<10;i++)

{

cout<<(i+1)<<"."<<Dat.Movies[9][i]<<endl;

}

}

break;

default:

{

goto movpage;

showerr();

}

break;

}

showch();

cin>>ch1;

if(ch1>=1 and ch1<=10)

{

u.totmov++;

strcpy(u.movs[(u.totmov-1)],Dat.Movies[(ch-1)][(ch1-1)]);

u.calcEP();

u.calcStatus();

fo.open(name,ios::out);

fo.write((char \*) &u,sizeof(u));

fo.close();

delay(1.0);

cout<<"\a\nCongratulations, "<<name<<", you've watched

"<<u.movs[(u.totmov-1)]<<" :)"<<endl<<endl;

cout<<"\aYou've earned 100 Emerald Points!"<<endl;

}

else

{

showerr();

goto movpage;

}

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

break;

case 6:

{

cout<<"Press enter to continue..."<<endl;

cin.get();

showem();

system("Color 34");

cout<<"Add/Modify Your Password:"<<endl<<endl;

cout<<"\nEnter new password(max 19 characters):\n ";

cin.getline(password,18);

strcpy(u.pass,password);

showem();

cout<<"\aPassword modified succesfully!\a"<<endl;

fo.open(name);

fo.write((char \*) &u,sizeof(u));

fo.close();

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

break;

case 7:

{

pagermvp:

showem();

cout<<"Remove Your Password:"<<endl<<endl;

system("Color 34");

cout<<"Remove password (Y/N)?"<<endl;

cin>>reply;

if(reply=='y'||reply=='Y')

{

strcpy(u.pass,"\*\*\*\*\*\*\*\*");

fo.open(name);

fo.write((char \*) &u,sizeof(u));fo.close();system("cls");

cout<<"\aPassword removed successfully!\a"<<endl;

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

}

else if(reply=='n'||reply=='N')

{

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

else

{

showerr();

goto pagermvp;

}

}

break;

case 8:

{

pagedelacc:

showem();

system("Color 34");

cout<<"Delete Your Account:"<<endl<<endl;

cout<<"Do you want to delete your account (Y/N)?";

cin>>reply;

if(reply=='y'||reply=='Y')

{

showem();

remove(name);

cout<<"\aYour account is deleted successfully!\a"<<endl;

namecutter(name);

return 0;

}

else if(reply=='n'||reply=='N')

{

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

else

{

showerr();

goto pagedelacc;

}

}

case 9:

{

pagelogout:

showem();

system("Color 34");

cout<<"Log Out:"<<endl<<endl;

cout<<"\nDo you want to log out from your account

(Y/N)?"<<endl;

cin>>reply;

if(reply=='y'||reply=='Y')

return 0;

else if(reply=='n'||reply=='N')

{

choice=gotoEM();

if(choice=='y' or choice=='Y')

goto page5;

else

return 0;

}

else

goto pagelogout;

}

break;

default:

{

goto page5;

showerr();

}

break;

}

return 0;

}

void showch()

{

cout<<"\nSelect from the given options:";

}

void showem()

{

system("cls");

showtitle();

cout<<"Welcome to The World of Art and Literature"<<endl<<endl;

}

char gotomain()

{

char ch;

cout<<"\nWant to go to Main Menu (Y/N)?"<<endl;

showch();

cin>>ch;

return ch;

}

char gotoEM()

{

char ch;

cout<<"\nWant to go to Account Menu (Y/N)?"<<endl;

showch();

cin>>ch;

return ch;

}

void Nameit(char \*a)

{

for(int i=0;a[i]!='\0';i++)

{

if(i==0)

a[i]=toupper(a[i]);

else

a[i]=tolower(a[i]);

}

}

void namecutter(char name[])

{

userlist u;

char temp[50];

int i;

ifstream fi("Userlist.DAT");

fi.read((char \*) &u,sizeof(u));

fi.close();

for(i=0;i<(u.no-1);i++)

{

if(strcmp(u.users[i],name)==0)

{

strcpy(temp,u.users[i]);

strcpy(u.users[i],u.users[(i+1)]);

strcpy(u.users[(i+1)],temp);

}

}

u.no--;

ofstream fo("Userlist.DAT");

fo.write((char \*) &u,sizeof(u));

fo.close();

}

void delay(float secs)

{

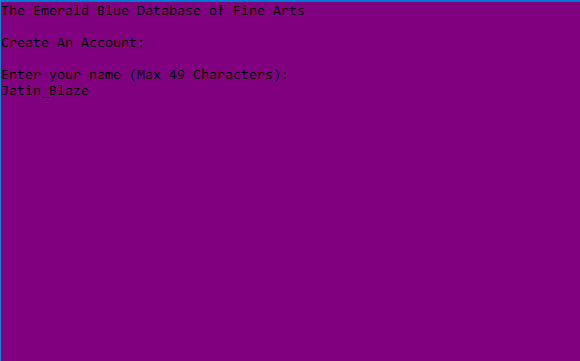
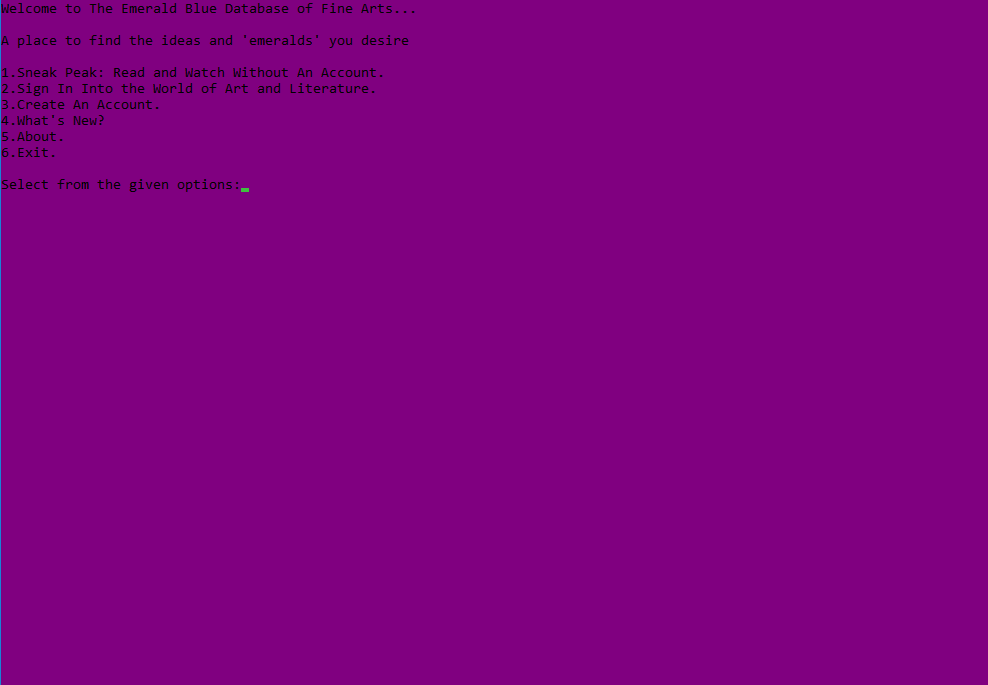
clock\_t timedelay=secs\*CLOCKS\_PER\_SEC;

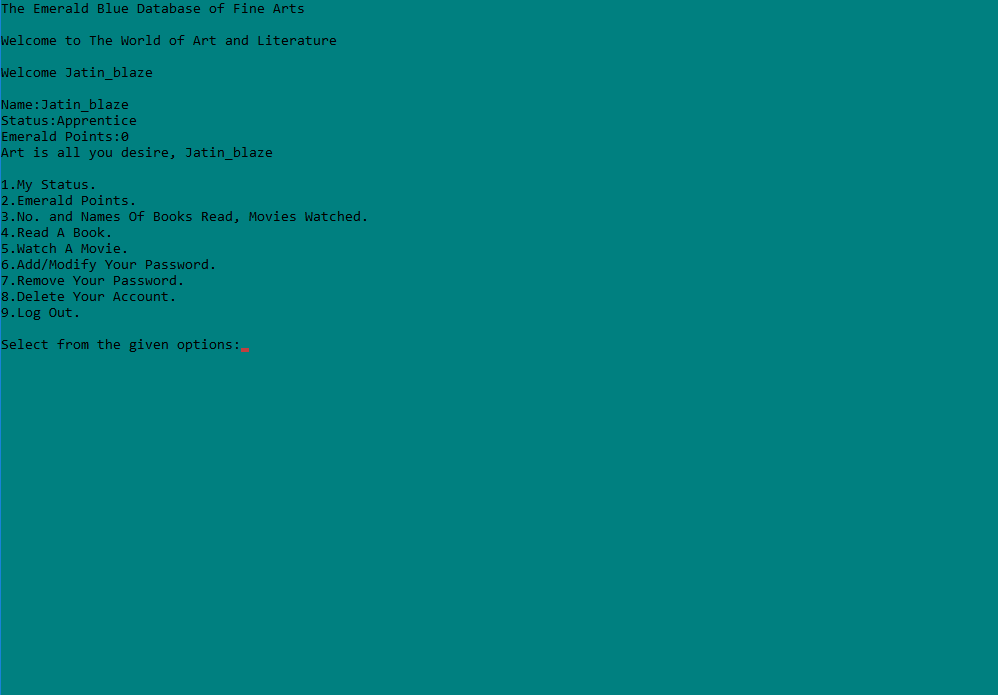
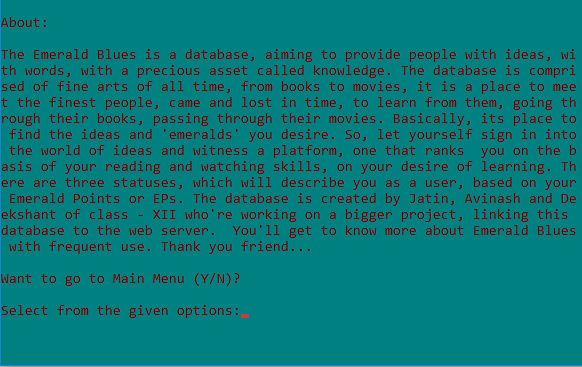
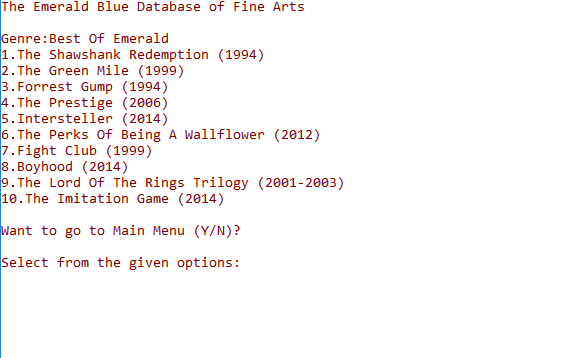
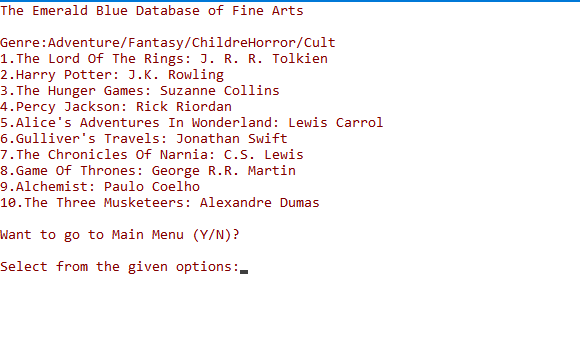
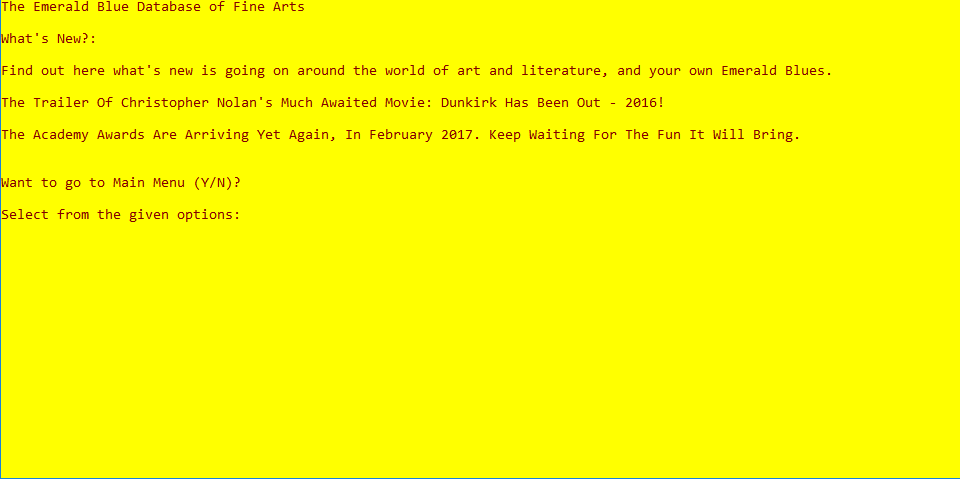
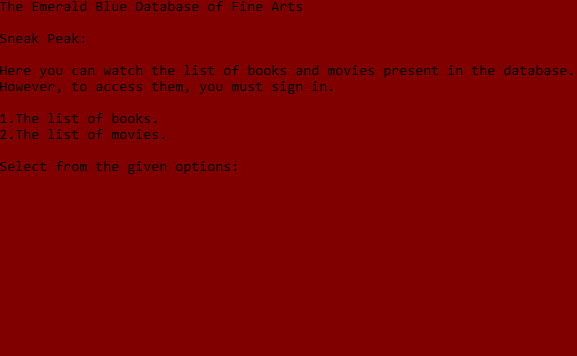
clock\_t start=clock();

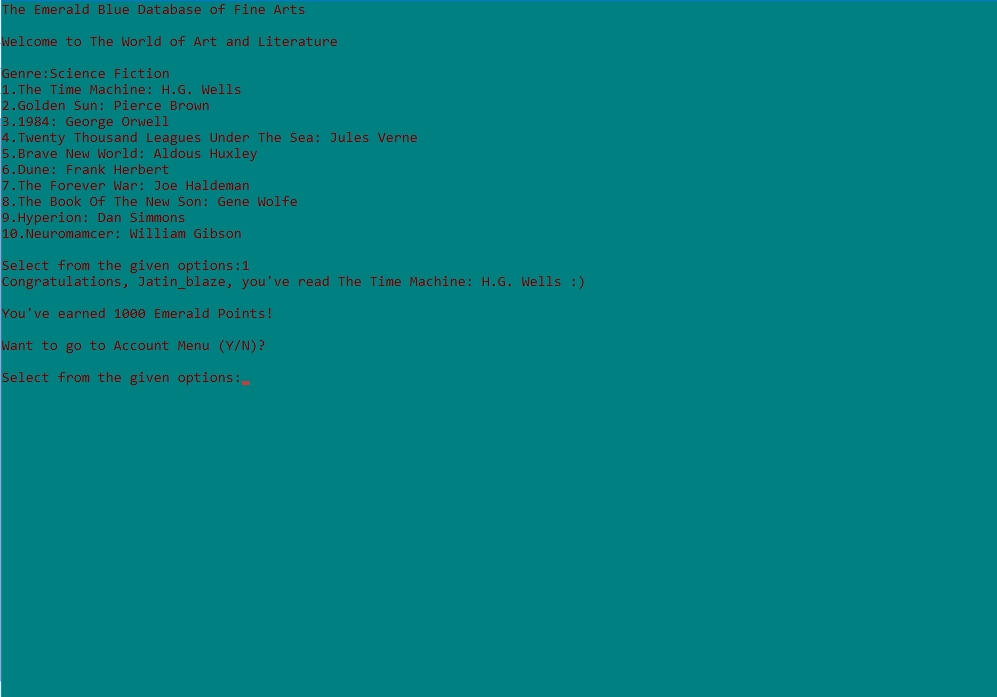
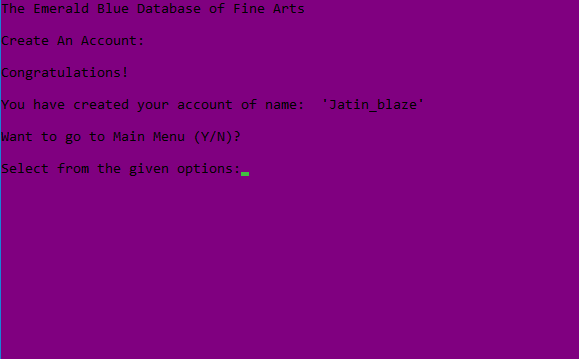
while((clock()-start)<timedelay);

}

end\_

**SNAPSHOTS:**

****

****

**ANALYSIS:**

The application certainly has some striking feature over the manual system. The time is gone when people used to sit in front of their screens and the pages of a never ending text. The database will provide a user with a whole new experience, into the world of art and literature, into the world the database defines as ‘Emerald Blues’. The algorithm used to score users with EPs is based on a simple maths trick. No doubt the Virtual Database Management System will be helpful for people interested in ‘learning’. It’s the ever increasing desire of people to make life simple with advancements in technology and programming that seeded the idea of starting with this splendid project. We’re glad that we have made this idea came true.

**FUTURESCOPE:**

There is no doubt that there always remains some scope of improvement. The important thing is the system developed should be flexible to accommodate any future enhancements. This system can be used to provide some enhancements without rewriting the existing code.

The application still can have the following enhancements:

* It can be connected to a dynamic or actual hard disk so as to make the experience reach new heights.
* It can be linked to a web server so as to connect the users around the globe to a single network of the database.
* New books and movies or even more could be added to the database in the near future. Also, new or better graphics and visuals could be added.

**BIBLIOGRAPHY:**

* **BOOKS:**
* C++ for Class XII – Part I and II: Sumita Arora
* C++ Primer Plus: Stephen Prata
* Programming: Bjarne Stroustrup
* **WEBSITES:**
* Wikipedia.com
* Google.com
* Twitter.com