

ABSTRACT

A reservation system named **STEGAN CINEMAS** will be able to provide faster and an alternative method of booking of seat for a movie. The admin will be able to create Stegan ID, delete Stegan ID (on violation of rules), and update the upcoming movies to the database.

Watching movies with family and friends in theaters is one of the best medium of entertainment after having a hectic schedule. But all this excitement vanishes after standing in hours in long queues to get tickets booked.

The system provides complete information regarding currently running movies on all the screens with details of show timings, available seats and fare charges of different classes. Seats can be reserved for different classes as well for same show and screen also. Tickets can be cancelled if needed.

The existing system has two ways of booking a ticket:

- One common method: Book tickets at ticket counter of respective screen and it is a hectic process where one should stand in long queue for hours.
- Tele-Booking: Booking tickets through phone calls. The introduction of this system kept user at comfort while booking tickets.

The primary objective of this system is to provide a three-dimensional way to book ticket(s) for a movie in Stegan Cinemas. It saves ones time and labor. This system is designed in a beautiful way such that one can cancel the booked ticket(s).

INTRODUCTION

This project is aimed at developing a ticket reservation system of Stegan Cinemas. This system will automate the reservation of tickets. Stegan Cinemas is one of the best opportunities for those who cannot afford enough time to get their tickets reserved standing in long queues. People can book tickets online at any time of day or night. Our reservation system also provides option to cancel the tickets which are reserved previously.

Keeping in view the customer's benefit, the system also has an additional functionality where the refund is available on cancellation of tickets. This is one of the functionalities where the previous systems were lacking with.

In this system there are two major modes. One is Stegan ID user mode the other one is Admin user mode. Admin plays a vital role in this system as he is the one who update the information of movies regularly. Stegan members can buy tickets, cancel tickets, and view running movies. The main advantage of the proposed system over existing system is, Users can book tickets according to their wish by selecting the seat number.

The system is designed in such a way that, it is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data, no formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly and can lead to error free, secure, reliable and fast management system.

This system will ultimately allow you for the better management of resources. And wastage of data is ultimately reduced in our system.

MODULE DESCRIPTION

In this System, functions have been used to display the menu and manipulate the data of the screen and also file handling is also performed to store the user ID and bus data. Here I have used many functions (mentioned below).

- **get_username();**

In this function the input part is coded. This function is used to get the username of a standard user ID when creating or deleting a standard user ID.

- **get_password();**

In this function the input part is coded. This function is used to get the password of a standard user ID when creating a deleting a standard user ID.

- **check_empty();**

This is a member function of class reserve. This function is used to check whether a particular seat of a particular bus is empty or not.

- **display();**

This is a member function of class reserve. This function is used to display the details of a particular bus.

- **disp_seat();**

This is a member function of class reserve. This function is used to display the seat map of a particular bus.

- **check_empticket();**

This is a member function of class reserve. This function is used to check the total number of empty tickets in a given bus.

- **vline(char);**

In this function the output part is coded. This function is used to display a character repeatedly in a given line.

- **main_menu(void);**

In this function both output and input part is coded. This function is used to display the main menu and select the options to perform the required operations.

- **admin_login(void);**

In this function both the input and output part is coded. This function is used to get the login details for the admin and interpret the login details and let the user to be authorized.

- **admin_mode(void);**

In this function both the input and output part is coded. This function displays all the operations that could be performed by an admin and the admin the choose the operations that he/she wants to do.

- **create_id(void);**

In this function both the input and output part is coded. This function is used to create a standard user ID which authorizes a standard end user to use the application.

- **delete_id(void);**

In this function both the input and output part is coded. This function is used to delete a standard user ID which removes the authentication of a standard end user to use the application.

- **install_movie(void);**

In this function both the input and output part is coded. This function is used to add the details of a particular movie/screen to the file.

- **standard_login(void);**

In this function both the input and output part is coded. This function is used to authenticate the end user for using the application.

- **standard_mode(void);**

In this function both the input and output part is coded. This function is used to display all the possible operations that a standard end user can perform. The user can select the operation to be performed with the help of this function.

- **book_tickets(void);**

In this function both the input and output part is coded. This function is used to book tickets in a particular with the help of bus number and the user ID.

- **cancel_ticket(void);**

In this function, user can delete the ticket(s) which he booked already.

- **main_menu(void);**

In this function, user can able to visit the main menu.

- **showall(void);**

In this function user can able to see each and every screen in the theatre.

- **deletescreen(void);**

In this function, admin can delete a particular (if needed).

- **showmovie(void);**

In this function, user can able to see what all the movies are screened currently.

EXISTING AND PROPOSED SYSTEM

Existing System:

The existing system has two ways of booking a ticket:

- One common method: Book tickets at ticket counter of respective screen and it is a hectic process where one should stand in long queue for hours.
- Tele-Booking: Booking tickets through phone calls. The introduction of this system kept user at comfort while booking tickets.

The main advantage of proposed system over the existing system is one can select their seat according to their wish.

Proposed System:

The proposed system reduces time and manual work(s). The system booking system will provide facility to book tickets for a movie online. A user can book tickets anytime and anywhere. The Online Cinema Ticket Booking System will provide detailed information so that a customer can know about the movie and based on the information customer will book the ticket. The user can search for the movies which will be going to release in the future, so they will have an option to book the ticket in advance. The Online Cinema Ticket Booking System will make the user experience much better than the existing system. The Cinema Ticket Booking System will provide a much better experience for booking of movie tickets. If seats are available, the customer can change the position of the seats while they will get an option to cancel of tickets.

SYSTEM SPECIFICATION

Hardware requirements:

Processor : Pentium(R) Dual core CPUE5400 @2.70GHz

RAM : 1.00 GB (2.00 GB Recommended)

Bit : 32-bit operating system

Hard disk : 900MB space required

Software requirements:

Software : Turbo C++ 3.8.0.1

Operating system : Windows 7

ABOUT C++

C++ is general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation. It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights.

Before the initial standardization in 1998, C++ was developed by Bjarne Stroustrup at Bell Labs since 1979, as an extension of the C language as he wanted an efficient and flexible language similar to C, which also provided high-level features for program organization. In 1979, Bjarne Stroustrup, a Danish computer scientist, began work on "C with Classes", the predecessor to C++.The motivation for creating a new language originated from Stroustrup experience in programming for his Ph.D. thesis.

Stroustrup found that Simula had features that were very helpful for large software development, but the language was too slow for practical use, while BCPL was fast but too low-level to be suitable for large software development. When Stroustrup started working in AT&T Bell Labs, he had the problem of analyzing the UNIX kernel with respect to distributed computing. Remembering his Ph.D. experience, Stroustrup set out to enhance the C language with Simula-like features.

C was chosen because it was general-purpose, fast, portable and widely used. As well as C and Simula's influences, other languages also influenced C++, In 1983, "C with Classes" was renamed to "C++" ("++" being the increment operator in C), adding new features that included virtual functions, function name and operator overloading, references, constants, type-safe free-store

memory allocation (new/delete), improved type checking, and BCPL style single-line comments with two forward slashes (//).

Furthermore, it included the development of a standalone compiler for C++, Cfront. In 1985, the first edition of The C++ Programming Language was released, which became the definitive reference for the language, as there was not yet an official standard. The first commercial implementation of C++ was released in October of the same year.

In 1989, C++ 2.0 was released, followed by the updated second edition of The C++ Programming Language in 1991. New features in 2.0 included multiple inheritance, abstract classes, static member functions, const member functions, and protected members.

In 1990, The Annotated C++ Reference Manual was published. Adding numerous new features, enlarging the standard library further, and providing more facilities to C++ programmers. After a minor C++14 update released in December 2014, various new additions are planned for 2017 and 2020. The C++ language has two main components: a direct mapping of hardware features provided primarily by the C subset and zero-overhead abstractions based on those mappings.

Stroustrup describes C++ as "a light-weight abstraction programming language [designed] for building and using efficient and elegant abstractions"; and "offering both hardware access and abstraction is the basis of C++. Doing it efficiently is what distinguishes it from other languages" C++ is often considered to be a superset of C, but this is not strictly true. Most C code can easily be made to compile correctly in C++, but there are a few differences that cause some valid C code to be invalid or behave differently in C++.

SOURCE CODE

```
#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<string.h>

#include<process.h>

#include<fstream.h>

#include<dos.h>

#include<iomanip.h>

char c=254;

void vline(char);//Function Prototype

class ID //Class for USER ID

{

    public:

        char user_name[20];

        char password[20];

        char get_username() //Member function to get username

        {

            cout<<"\n\n\t Enter the user name of the ID: ";

            gets(user_name);

            return user_name[20];

        }

        char get_password() //Member function to get password

        {

            cout<<"\n\n\t Enter the password of the ID: ";
```

```

        gets(password);
        return password[20];
    }
}st_login;

class reserve //Class for the details of the bus
{
public:
    int screen;
    char movie[30];
    char hero[20];
    char heroin[20];
    char showtime[5];
    char runtime[5];
    char seat[10][10];
    char user_id[10][10][20];
    int cost;
    void display(); //Member function to display bus details
    void disp_seat(); //Member function to display seat details
    int check_empticket(); //Member function to check number of empty tickets
in bus
    int check_empty(int i,int j) //Member function to check whether the seat is
empty
    {
        if(seat[i][j]==c)
        {
            return 1;

```

```

    }
else
{
    return 0;
}
}
};

void reserve::display()
{
    clrscr();
    vline('.');
    cout<<"\n\n\t\t\t The details of Screen";
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\nScreen number: ";
    cout<<screen;
    cout<<"\n\n\tMovie's name: ";
    cout<<movie;
    cout<<"\n\n\tStarring: "<<hero<<" and "<<hero;
    cout<<"\n\n\tShow time: ";
    for(int x=0;x<5;x++)cout<<showtime[x];
    cout<<"\n\n\tRun time: ";
    for(x=0;x<5;x++)cout<<runtime[x];
    cout<<"\n\n\tCost: ";
    cout<<cost;
    cout<<"\n\n\n\n\t\t Press any key to continue to seat map";

```

```

    getch();
}
void reserve::disp_seat()
{
    int s=0,p=0,c=0;
    for(int i=0;i<10;i++)
    {
        cout<<"\n";
        if(i==4 || i==7)cout<<endl;
        if(i==0)cout<<" ";
        for(int j=0;j<10;j++)
        {
            s++;
            if(j==3 || j==7)cout<<"    ";
            if((j==3 || j==7)&& i==0)cout<<" ";
            cout<<s<<".";
            cout.width(3);
            cout.fill(' ');
            cout.width(3);
            cout.fill(' ');
            cout<<seat[i][j];
            p++;
        }
    }

    for(i=0;i<10;i++)for(int x=0;x<10;x++)if(seat[i][x]=='R')c++;
    cout<<endl<<endl;
}

```

```

    vline('_');cout<<"\t\t\t\t\t SCREEN THIS WAY";cout<<endl; vline('_');
    cout<<"\t\t\t\t\tThere are "<<100-c<<" empty seats in screen"<<screen;
}

int reserve::check_empticket()
{
    int e=0;
    for(int i=0;i<10;i++)
    {
        for(int j=0;j<10;j++)
        {
            if(seat[i][j]==c)
                e++;
        }
    }
    return e;
}

//Function prototypes
void main_menu(void);
void admin_login(void);
void admin_mode(void);
void create_id(void);
void delete_id(void);
void standard_login(void);
void standard_mode(void);
void book_tickets(void);
void install_movie(void);

```

```

void delete_screen(void);
void show_movie(void);
void show_seats(reserve);
void cancel_ticket(void);
void show_all(void);
void vline(char ch) //Function to insert line
{
    for(int i=1;i<80;i++)
        cout<<ch;
}
void start_screen() //Start Screen Function
{
    clrscr();
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\t\t\t Welcome to Stegan Cinemas";
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t Press any key to continue";
    getch();
}
void main_menu() //Main menu function
{
    clrscr();
    int choice1;
    cout<<"\n\n";

```



```

vline('.');
cout<<"\n\n\t\t\tWELCOME TO STEGAN CINEMAS";
cout<<"\n\n";
vline('.');
cout<<"\n\n\t\t\t Choose the login mode";
cout<<"\n\n\t\t\t 1.ADMIN Mode";
cout<<"\n\n\t\t\t 2.Login";
cout<<"\n\n\t\t\t 3.Exit";
cout<<"\n\n\n\t Enter your choice (1 , 2 or 3): ";
cin>>choice1;
switch(choice1)
{
    case 1:admin_login();
        break;
    case 2:standard_login();
        break;
    case 3:cout<<"\n\n\n\n\t\t\t Press any key to exit";
        getch();
        exit(0);
        break;
    default:cout<<"\n\n\t\t\t Enter a valid choice";
        cout<<"\n\n\t\t\t Press any key to continue";
        getch();
        main_menu();
        getch();
        break;
}

```

```

    }
}

void admin_login() //Function for admin login page
{
    clrscr();
    char admin_name[8],admin_pass[10],try_choice;
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\t\t\t ADMIN Login";
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\n\t\t Enter the admin name: ";
    gets(admin_name);
    cout<<"\n\n\t\t Enter the password: ";
    gets(admin_pass);
    if(strcmp("Surya",admin_name)==0 && strcmp("lm10",admin_pass)==0)
    {
        cout<<"\n\n\n\n\t\t\t Press any key to continue";getch();admin_mode();
    }
    else
    {
        cout<<"\n\n\n\n\t\t\t Incorrect Password";
        label1:
        cout<<"\n\n\t\t\t Do you want to try again(y/n):";
        cin>>try_choice;
        switch(try_choice)

```

```

{
    case 'y':admin_login();
        break;

    case 'n':cout<<"\n\n\t\t Press any key to return back to main menu";
        getch();
        main_menu();
        break;

    case 'Y':admin_login();
        break;

    case 'N':main_menu();
        break;

    default :cout<<"\n\n\t\t Enter a valid input";
        goto label1;
}
}

getch();
}

void admin_mode() //Admin Page
{
    clrscr();
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\t\t\t\t ADMIN MENU";
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\t 1.To create a Stegan ID.";

```

```

cout<<"\n\n\t 2.To delete a existing Stegan ID.";
cout<<"\n\n\t 3.Add a Movie to the database.";
cout<<"\n\n\t 4.Delete a Movie.";
cout<<"\n\n\t 5.Return to main menu.";
cout<<"\n\n\t 6.Exit.";
cout<<"\n\n\t Enter your choice:";
int choice2;
cin>>choice2;
switch(choice2)
{
case 1:create_id();
        break;
case 2:delete_id();
        break;
case 3:install_movie();
        break;
case 4:delete_screen();
        break;
case 5:cout<<"\n\n\t\t Press any key to return back to main menu";
        getch();
        main_menu();
        break;
case 6:cout<<"\n\n\t\t\t Press any key to exit";
        getch();
        exit(0);
        break;
}

```

```

default:cout<<"\n\n\t\t\t Enter a valid choice";

        cout<<"\n\n\t\t\t Press any key to continue";

        getch();

        admin_mode();

        break;

    }

    getch();

}

void create_id() //Function to create a user ID

{

    char choice='n';

    do

    {

        label3:

        fstream login("user.dat",ios::in|ios::out|ios::app|ios::binary);

        clrscr();

        ID user,confirm,temp;

        cout<<"\n\n";

        vline('.');

        cout<<"\n\n\t\t\t Create a Stegan ID";

        cout<<"\n\n";

        vline('.');

        user.get_username();

        cout<<"\n";

        cout<<"\t\t\t Retype the user name: ";

        gets(confirm.user_name);

```

```

if(strcmp(user.user_name,confirm.user_name)!=0)
{
    cout<<"\n\n\n\n\n\t\tThe usernames that you have entered doesn't match";
    cout<<"\n\n\n\n\n\t\t Press any key to try again";
    getch();
    goto label3;
}

fstream z("user.dat",ios::in|ios::binary);
while(!z.eof())
{
    z.read((char*)&temp,sizeof(temp));
    if(strcmp(temp.user_name,user.user_name)==0)
    {
        cout<<"\n\n\t This Stegan ID already taken";
        cout<<"\n\n\n\n\n\t\t\t Press any key to try again";
        getch();
        z.close();
        goto label3;
    }
}

z.close();
user.get_password();
cout<<"\n";
cout<<"\t\t\t Retype the password: ";
gets(confirm.password);
if(strcmp(user.password,confirm.password)!=0)

```

```

{
    cout<<"\n\n\n\n\n\t\tThe passwords that you have entered doesn't match";
    cout<<"\n\n\n\n\n\t\t Press any key to try again";
    getch();
    goto label3;
}

login.write((char*)&user,sizeof(user));

cout<<"\n\n\n\t\t Your ID has been created successfully";

login.close();

cout<<"\n\n\t\t Do you want to create another ID: ";
cin>>choice;
}

while(choice=='y' || choice=='Y');

cout<<"\n\n\t\t Press any key to return to main menu";
getch();

admin_mode();
}

void delete_id() //Function to delete a user ID
{
    char found='n';
    label4:
    clrscr();
    ID del_user,temp_del,confirm,temp2;
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\t\t Delete a Stegan user ID";

```

```

cout<<"\n\n";
vline('.');
del_user.get_username();
cout<<"\n";
cout<<"\t\t\t\t\t Retype the user name: ";
gets(confirm.user_name);
if(strcmp(del_user.user_name,confirm.user_name)!=0)
{
    cout<<"\n\n\n\n\n\n\t\t\t\t\tThe usernames that you have entered doesn't match";
    cout<<"\n\n\n\n\n\n\t\t\t\t\t Press any key to try again";
    getch();
    goto label4;
}
fstream y("user.dat",ios::in|ios::binary);
while(!y.eof())
{
    y.read((char*)&temp2,sizeof(temp2));
    if(strcmp(temp2.user_name,del_user.user_name)==0)
    {
        found='y';
        break;
    }
}
y.close();
if(found=='n')
{

```



```

cout<<"\n\n\t The entered stegan ID not found";
cout<<"\n\n\t\t Returning to main menu"; getch();
admin_mode();
}

fstream del_org("user.dat",ios::in|ios::out|ios::binary);
fstream del_temp("temp.dat",ios::in|ios::out|ios::app|ios::binary);
while(!del_org.eof())
{
    del_org.read((char*)&temp_del,sizeof(temp_del));
    if(strcmp(del_user.user_name,temp_del.user_name)!=0)
    {
        del_temp.write((char*)&temp_del,sizeof(temp_del));
    }
}

del_org.close();
del_temp.close();
remove("user.dat");
rename("temp.dat","user.dat");
cout<<"\n\n\n\t\t Your ID has been deleted successfully";
cout<<"\n\n\t\t Press any key to return to main menu";
getch();
admin_mode();
}

void install_movie()
{
    clrscr();

```

```

reserve ins;

cout<<"\n\n";

vline('.');

cout<<"\n\n\t\t\t Add details of the screen";

cout<<"\n\n";

vline('.');

cout<<"\n\n\n\tEnter Screen Number: ";

cin>>ins.screen;

cout<<"\tEnter    Movie's Name: ";

gets(ins.movie);

cout<<"\n\tEnter Hero's name: ";

gets(ins.hero);

cout<<"\n\tEnter Heroin's name: ";

gets(ins.heroin);

cout<<"\n\tEnter Show Time ";

gets(ins.showtime);

cout<<"\tEnter Run Time: ";

gets(ins.runtime);

cout<<"\n\tCost: ";

cin>>ins.cost;

for(int i=0; i<10;i++)
{
    for(int j=0;j<10;j++)
    {
        ins.seat[i][j]=c;
    }
}

```

```

}

fstream inst("scrdata.dat",ios::app|ios::binary);
inst.write((char*)&ins,sizeof(ins));
cout<<"\n\n\t#Added successfully.";
inst.close();
getch();
admin_mode();
}

void delete_screen() //Function to delete a user ID
{
char found='n';
label11:
clrscr();
reserve del_screen,temp_del,confirm,temp2;
cout<<"\n\n";
vline('.');
cout<<"\n\n\t\t\t Delete a Stegan screen";
cout<<"\n\n";
vline('.');
cout<<"\n\n\tEnter the screen no:";
cin>>del_screen.screen;
fstream q("scrdata.dat",ios::in|ios::binary);
while(!q.eof())
{
q.read((char*)&temp2,sizeof(temp2));
if(del_screen.screen==temp2.screen)

```

```

{
    found='y';
    break;
}
}
q.close();
if(found=='n')
{
    cout<<"\n\n\t The entered stegan screen not found";
    cout<<"\n\n\t\t Returning to main menu"; getch();
    admin_mode();
}
fstream del_org("scrdata.dat",ios::in|ios::out|ios::binary);
fstream del_temp("temp2.dat",ios::in|ios::out|ios::app|ios::binary);
while(!del_org.eof())
{
    del_org.read((char*)&temp_del,sizeof(temp_del));
    if(del_screen.screen!=temp_del.screen)
    {
        del_temp.write((char*)&temp_del,sizeof(temp_del));
    }
}
del_org.close();
del_temp.close();
remove("scrdata.dat");
rename("temp2.dat","scrdata.dat");

```

```

    cout<<"\n\n\n\t\t Your stegan screen has been deleted successfully";
    cout<<"\n\n\t\t Press any key to return to main menu";
    getch();
    admin_mode();
}

void standard_login() //Function for end user login
{
    ID temp_user;
    fstream std_login;
    std_login.open("user.dat",ios::out|ios::in|ios::binary);
    clrscr();
    cout<<"\n\n\t\t\t\t\tLogin";
    st_login.get_username();
    st_login.get_password();
    char found='n';
    while(!std_login.eof())
    {
        std_login.read((char*)&temp_user,sizeof(temp_user));
        if(strcmp(temp_user.user_name,st_login.user_name)==0)
        {
            if(strcmp(temp_user.password,st_login.password)==0)
            {
                found='y';
                break;
            }
        }
    }
}

```

```

}
if(found=='n')
{
cout<<"\n\n\n\n\n\n\n\n\n\n\t\t Either your name or password is incorrect";
label2:
cout<<"\n\n\t\t\t Do you want to try again(y/n):";
char try_choice1;
cin>>try_choice1;
switch(try_choice1)
{
case 'y':standard_login();
        break;
case 'n':main_menu();
        break;
case 'Y':standard_login();
        break;
case 'N':main_menu();
        break;
default :cout<<"\n\n\t\t Enter a valid input";
        goto label2;
}
}
std_login.close();
if(found=='y')
{
cout<<"\n\n\n\n\n\n\n\n\n\n\t\t Press any key to continue";

```

```

    getch();
    standard_mode();
}
}

void standard_mode()
{
    clrscr();
    cout<<"\n\n\t\t\t Welcome To Stegan Cinemas";
    cout<<"\n\n\n\n\t 1.Book Tickets.";
    cout<<"\n\n\t 2.Movies Running";
    cout<<"\n\n\t 3.Cancel your Tickets";
    cout<<"\n\n\t 4.Main Menu";
    cout<<"\n\n\t 5.Instuctions";
    cout<<"\n\n\t 6.Screens Available";
    cout<<"\n\n\t 7.Exit";
    cout<<"\n\n\n\n\t Enter your choice:";
    int choice3;
    cin>>choice3;
    switch(choice3)
    {
        case 1:book_tickets();
            break;
        case 2:show_movie();
            break;
        case 3:cancel_ticket();
            break;
    }
}

```

```

case 4:main_menu();
    break;
case 5:clrscr(); cout<<"\n\n\n\t\tINSTRUCTIONS";
    cout<<"\n\n\n\n\t #DONT CARRY BABIES WITH YOU";
    cout<<"\n\t #ARRIVE TO CINEMA ON TIME";
    cout<<"\n\t #DRUNKEN PEOPLE ARE NOT ALLOWED";
    cout<<"\n\t #SMOKING IS STRICTLY PROHIBITED";
    cout<<"\n\t #EXCHANGE THE TICKET IN TIME";
    cout<<"\n\t #MUST STAND UP FOR THE NATIONAL ANTHEM";
    cout<<"\n\t #SHOULD NOT BRING ANY EATABLES";
    cout<<"\n\t #MANAGEMENT IS NOT RESPONSIBLE FOR ANY LOSS
OF VALUBLES";
    cout<<"\n\n\n\n\n\n\n\n\n\n\n\t\t THANK YOU FOR VISITING";
break;
case 7:exit(0);
    break;
case 6: show_all(); break;
default:cout<<"\n\n\n\t\t\t Enter a valid choice";
    cout<<"\n\n\n\t\t\t Press any key to continue";
    getch();
    standard_mode();
}
getch();
}
void book_tickets() //Function to book tickets
{

```



```

reserve book_ticket,temp,temp1;
clrscr();
vline('.');
cout<<"\n\n\t\t\t\t Book Tickets\n\n";
vline('.');
cout<<"\n\n\t Enter the Screen number: ";
cin>>temp1.screen;
int n,temp_seat;
fstream ticket("scrdata.dat",ios::in|ios::out|ios::binary|ios::app);
ticket.seekg(0);
while(!ticket.eof())
{
    ticket.read((char*)&book_ticket,sizeof(book_ticket));
    if(book_ticket.screen==temp1.screen)
    {
        if(book_ticket.check_empticket()!=0)
        {
            cout<<"\n\n\tEnter the number of tickets to be booked: ";
            cin>>n;
            cout<<"\n\n";
            top:
            for(int i=1;i<=n;i++)
            {
                cout<<"\n\t Enter the seat "<<i<<": ";
                cin>>temp_seat;
                if(temp_seat>100)

```

```

{
    cout<<"\n\tThere are only 100 seats available in this screen.";
    goto top;
}

if(book_ticket.check_empty((temp_seat/10),(temp_seat%10)-1)==1)
{
    book_ticket.seat[temp_seat/10][(temp_seat%10)-1]='R';
    strcpy(book_ticket.user_id[temp_seat/10][(temp_seat%10)-1],st_login.user_name);
}
else
{
    cout<<"\n\t Already reserved !!";
    goto top;
}
}

fstream del_temp("temp.dat",ios::in|ios::out|ios::app|ios::binary);
ticket.seekg(0);
del_temp.seekp(0);
while(!ticket.eof())
{
    ticket.read((char*)&temp,sizeof(temp));
    if(book_ticket.screen!=temp.screen)
    {
        del_temp.write((char*)&temp,sizeof(temp));
    }
}

```

```

    }

    del_temp.close();
    ticket.close();
    remove("scrdata.dat");
    rename("temp.dat","scrdata.dat");
    fstream twrite("scrdata.dat",ios::app|ios::binary);
    twrite.write((char*)&book_ticket,sizeof(book_ticket));
    twrite.close();

    cout<<"\n\n\t\t Your tickets have been booked successfully";
    cout<<"\n\n\t The cost for the ticket is "<<book_ticket.cost;
    cout<<"\n\t Please pay the amount before you enter the cinema and get the
ticket";

    cout<<"\n\t Please carry each of the ticket for verification";
    cout<<"\n\n\t\t Press any key to return to main menu";

    break;
}

else
{
    cout<<"No tickets are available in this screen"; }
}
}

float a=n;

float m=book_ticket.cost;

clrscr();

vline('-');

cout<<"\t\t\t\t Stegan Cinemas"<<endl;

```



```

cout<<"\n\n";
vline('.');
cout<<"\n\n\n\t Enter the Screen number: ";
cin>>temp.screen;
fstream disp("scredata.dat",ios::in|ios::out|ios::binary);
while(!disp.eof())
{
    disp.read((char*)&search,sizeof(search));
    if(search.screen==temp.screen)
    {
        search.display();
        show_seats(search);
        break;
    }
}
disp.close();
cout<<"\n\n\n\t\t Press any key to return to main menu";
getch();
standard_mode();
}

void show_seats(reserve seat)
{
    clrscr();
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\t\t\t The details of the seats of the screen";

```

```

cout<<"\n\n";
vline('.');
cout<<"\n";
seat.disp_seat();
}
void cancel_ticket()
{
    label6:
    clrscr();
    reserve del,temp,temp2,temp3;
    char choice='n';
    int seats;
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n\t\t\t Cancel the booked tickets";
    cout<<"\n\n";
    vline('.');
    cout<<"\n\n";
    do
    {
        cout<<"\t Enter the Screen number: ";
        cin>>temp.screen;
        fstream scr("scrdata.dat",ios::binary|ios::in);
        while(!scr.eof())
        {
            scr.read((char*)&del,sizeof(del));

```

```

if(del.screen==temp.screen)
{
    cout<<"\n\n\t Enter the seat number to be cancelled: ";
    cin>>seats;
    if(strcmp(del.user_id[seats/10][(seats%10)-1],st_login.user_name)!=0)
    {
        cout<<"\n\n\t\t The seat is not booked from your ID";
        cout<<"\n\n\t Do yo want to try again(Y/N): " ;
        cin>>choice;
        if(choice=='y' || choice=='Y')
            goto label6;
        else
        {
            cout<<"\n\n\n\n\t\t Press any key to return to main menu";
            getch();
            standard_mode();
        }
    }
    if(del.seat[seats/10][(seats%10)-1]!=c)
    {
        del.seat[seats/10][(seats%10)-1]=c;
        fstream temp1("temp1.dat",ios::app|ios::binary);
        scr.seekg(0);
        while(!scr.eof())
        {
            scr.read((char*)&temp2,sizeof(temp2));

```

```

if(del.screen!=temp2.screen)
{
    temp1.write((char*)&temp2,sizeof(temp2));
}
}
temp1.write((char*)&del,sizeof(del));
temp1.close();
scr.close();
remove("scrdata.dat");
rename("temp1.dat","scrdata.dat");
cout<<"\n\n\t Do yo want cancel one more ticket(Y/N): " ;
cin>>choice;
break;
}
else
{
    cout<<"\n\n\t The selected seat havent booked by anyone";
    cout<<"\n\n\t Do yo want to try again(Y/N): " ;
    cin>>choice;
    if(choice=='y' || choice=='Y')
        goto label6;
    else
    {
        cout<<"\n\n\n\n\t\t Press any key to return to main menu";
        getch();
        standard_mode(); }
}

```



```

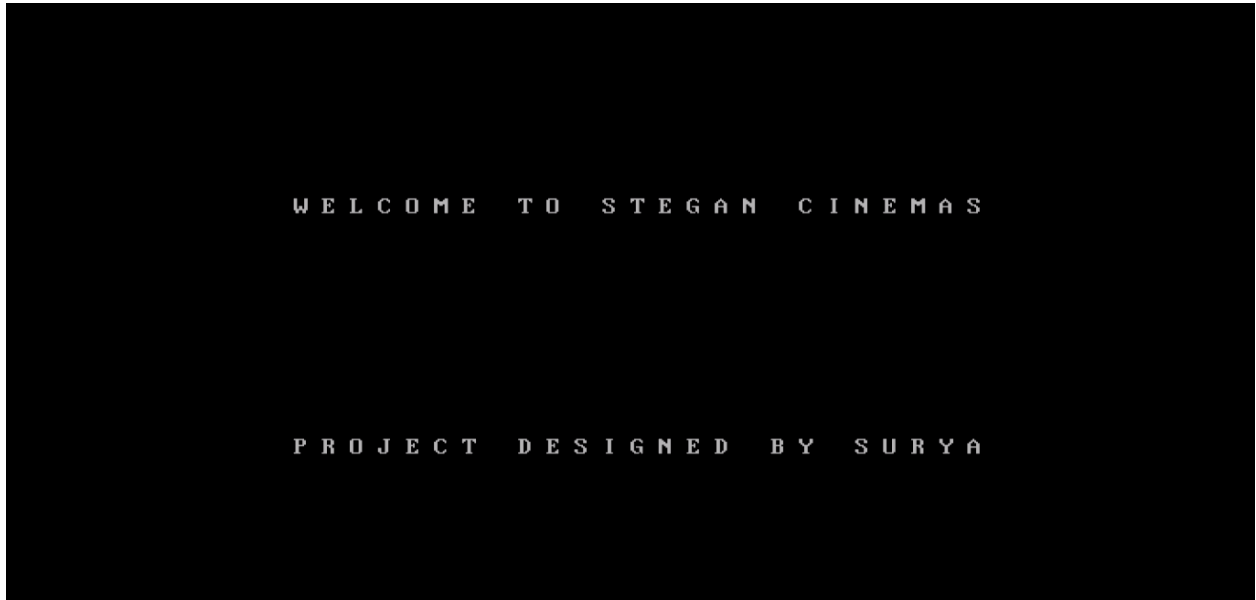
vline('.');
cout<<"\n\n\t\t Screens Available";
cout<<"\n\n";
vline('.');
cout<<"\n\n";
fstream t("scrdata.dat",ios::in|ios::binary);
t.read((char*)&temp,sizeof(temp));
while(t.read((char*)&temp,sizeof(temp)))
{
    cout<<"Screen "<<temp.screen<<": "<<temp.movie; cout<<endl<<endl;
}
t.close();
cout<<"\n\n\n\n\t\t Press any key to return to main menu";
    getch();
    standard_mode();
}

void main()
{clrscr();
start_screen();
main_menu();
}

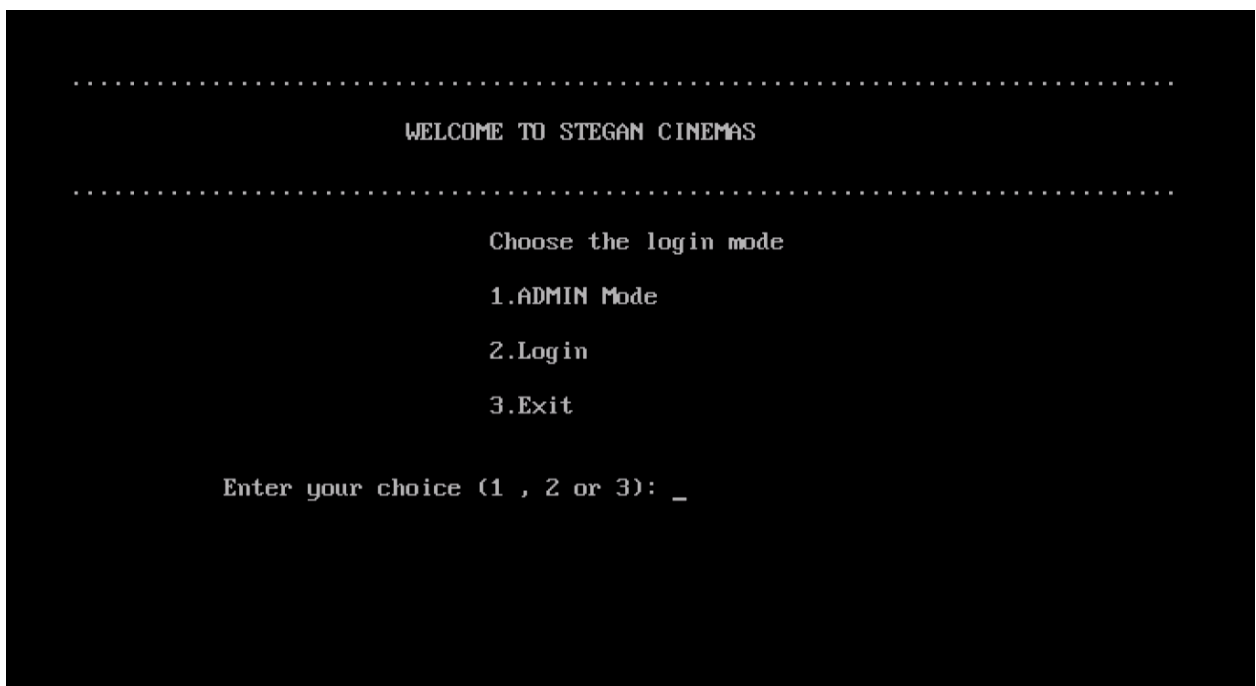
```

OUTPUT

Start Screen:



Main Menu:



Admin login Successful:

```
.....  
                                ADMIN Login  
.....  
  
Enter the admin name: Surya  
  
Enter the password: lm10  
  
Press any key to continue
```

Admin login Unsuccessful:

```
.....  
                                ADMIN Login  
.....  
  
Enter the admin name: Surya  
  
Enter the password: cr?  
  
Incorrect Password  
  
Do you want to try again(y/n):_
```

Admin Mode Screen:

```
.....  
                                ADMIN MENU  
.....  
  
    1.To create a Stegan ID.  
    2.To delete a existing Stegan ID.  
    3.Add a Movie to the database.  
    4.Delete a movie in Stegan Cinemas.  
    5.Return to main menu.  
    6.Exit  
  
Enter your choice:
```

Create User ID (Name Incorrect):

```
.....  
                                Create a Stegan ID  
.....  
  
Enter the user name of the ID: siddhu  
      Retype the user name: sidu  
  
  
The usernames that you have entered doesn't match  
  
Press any key to try again
```

Create User ID (Password Incorrect):

```
.....  
Create a Stegan ID  
.....  
Enter the user name of the ID: siddhu  
Retype the user name: siddhu  
  
Enter the password of the ID: kdb17  
Retype the password: pp06  
  
The passwords that you have entered doesn't match  
  
Press any key to try again
```

Create User ID (ID already exists):

```
.....  
Create a Stegan ID  
.....  
Enter the user name of the ID: siddhu  
Retype the user name: siddhu  
  
This Stegan ID already taken  
  
Press any key to try again
```

Create User ID (Confirmation Page):

```
*****
Create a standard user ID
*****

Your ID has been created successfully

Do you want to create another ID: _
```

Delete User ID (ID not found):

```
.....
Delete a Stegan user ID
.....

Enter the user name of the ID: lloris
Retype the user name: lloris

The entered stegan ID not found

Returning to main menu
```

Delete User ID (Confirmation Page):

```
*****
Delete a standard user ID
*****

Your ID has been deleted successfully

Do you want to delete another ID: _
```

Add a Screen to the database:

```
.....
Add details of the screen
.....

Enter Screen Number: 3
Enter Movie's Name: Jil Jung Juk

Enter Hero's name: Siddharth

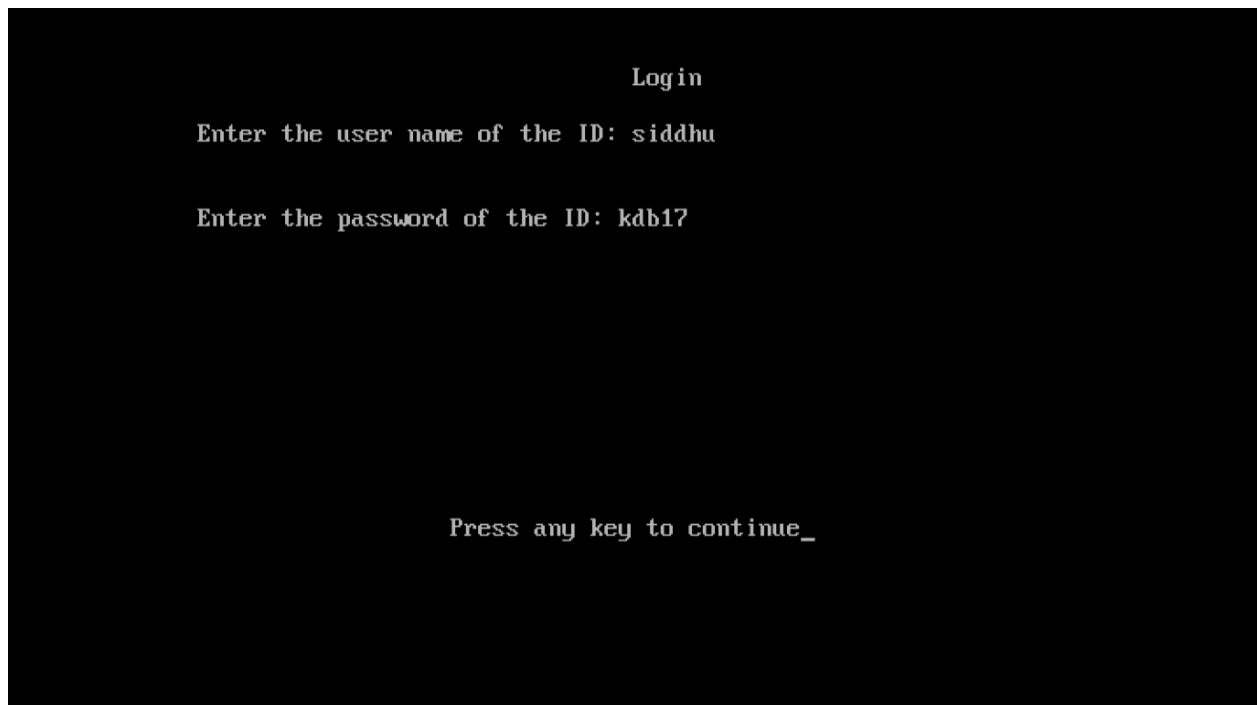
Enter Heroin's name: --none--

Enter Show Time 07:00
Enter Run Time: 02:28

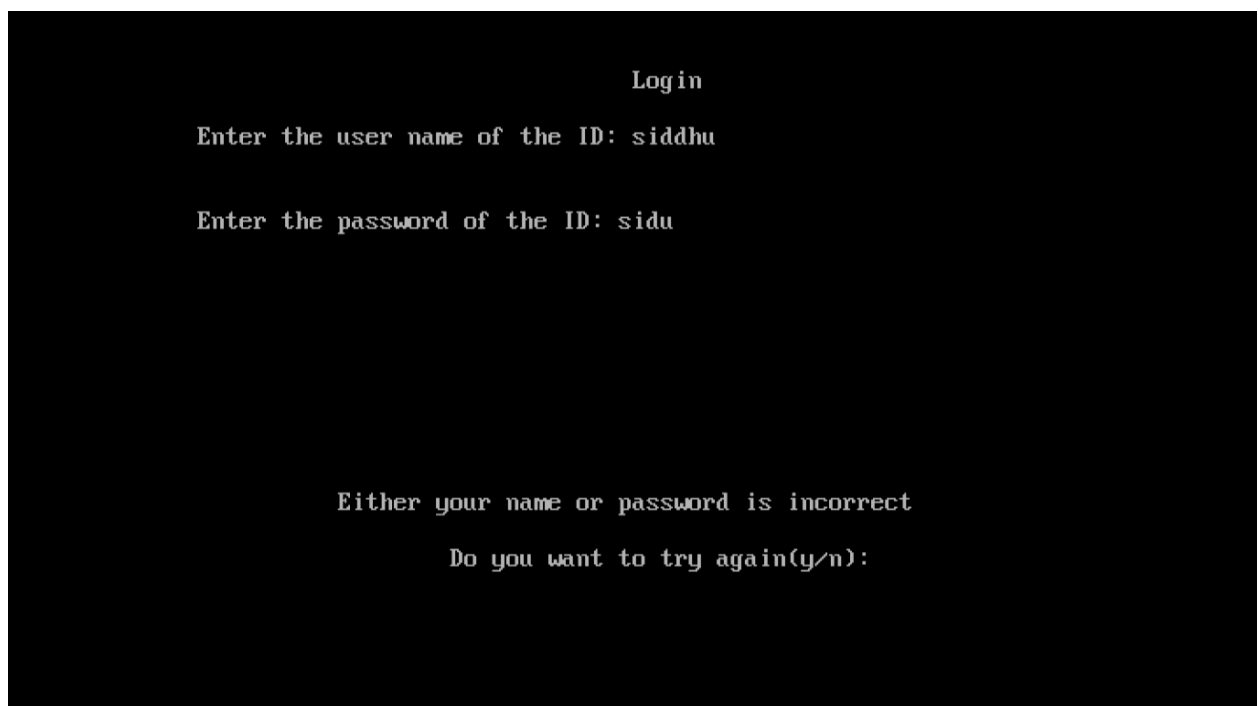
Cost: 150

#Added successfully.
```


Stegan User login Successful:



Stegan User login Unsuccessful:



Stegan User Mode Menu:

```

                                Welcome To Stegan Cinemas

1.Book Tickets.
2.Movies Running
3.Cancel your Tickets
4.Main Menu
5.Instructions
6.Screens Available
7.Exit

Enter your choice:_
```

Booking Tickets:

```

                                Book Tickets
.....

Enter the Screen number: 1

Enter the number of tickets to be booked: 3

Enter the seat 1: 17
Enter the seat 2: 18
Enter the seat 3: 19

Your tickets have been booked successfully

The cost for the ticket is: 3X130=390
Please pay the amount before you enter the cinema and get the ticket
Please carry each of the ticket for verification

Press any key to return to main menu
```

Ticket Bill:

Stegan Cinemas		
BILL		
Ticket(s)	Cost	Total
4	160	640
Goods and service taxes(18%)		
CGST	SGST	IGST
115.2	115.2	230.4
		Grand Total: 870.4

Screen Details

The details of Screen	
Screen number: 3	
Movie's name: Jil Jung Juk	
Starring: Siddharth and --none--	
Show time: 07:00	
Run time: 02:28	
Cost: 150	
Press any key to continue to seat map	

Show Screen Details (Seat Map):

```

                                The details of the seats of the screen
.....

1.■ 2.■ 3.■          4.■ 5.■ 6.■ 7.■          8.■ 9.■ 10.■
11.■ 12.■ 13.R      14.R 15.R 16.■ 17.R      18.R 19.R 20.■
21.■ 22.■ 23.■      24.■ 25.■ 26.■ 27.■      28.■ 29.■ 30.■
31.■ 32.■ 33.■      34.R 35.R 36.■ 37.■      38.■ 39.■ 40.■

41.■ 42.■ 43.■      44.■ 45.■ 46.■ 47.■      48.■ 49.■ 50.■
51.■ 52.■ 53.■      54.■ 55.■ 56.■ 57.■      58.■ 59.■ 60.■
61.■ 62.■ 63.■      64.■ 65.■ 66.■ 67.■      68.■ 69.■ 70.■

71.■ 72.■ 73.■      74.■ 75.■ 76.■ 77.■      78.■ 79.■ 80.■
81.■ 82.■ 83.■      84.■ 85.■ 86.■ 87.■      88.■ 89.■ 90.■
91.■ 92.■ 93.■      94.■ 95.■ 96.■ 97.■      98.■ 99.■ 100.■

-----
                                SCREEN THIS WAY
-----
                        There are 92 empty seats in screen1

                                Press any key to return to main menu

```

Cancel Tickets (Details Page):

```

.....

                                Cancel the booked tickets
.....

Enter the Screen number: 1

Enter the seat number to be cancelled: 17

Do you want cancel one more ticket(Y/N):

```

Cancel Tickets (Reserved by other):

```
.....  
Cancel the booked tickets  
.....  
Enter the Screen number: 1  
  
Enter the seat number to be cancelled: 55  
  
The seat is not booked from your ID  
Do you want to try again(Y/N): _
```

Instructions:

```
INSTRUCTIONS  
  
#DONT CARRY BABIES WITH YOU  
#ARRIVE TO CINEMA ON TIME  
#DRUNKEN PEOPLE ARE NOT ALLOWED  
#SMOKING IS STRICTLY PROHIBITED  
#EXCHANGE THE TICKET IN TIME  
#MUST STAND UP FOR THE NATIONAL ANTHEM  
#SHOULD NOT BRING ANY EATABLES  
#MANAGEMENT IS NOT RESPONSIBLE FOR ANY LOSS OF VALUBLES  
  
THANK YOU FOR VISITING_
```

Screens Available:



CONCLUSION

The accurate knowledge of the theatre data and management allow the user to expand and grow his business. Cinema Reservation System is the perfect tool to assist in business management. Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adapted. The package shall prove to be a powerful package in satisfying the entire requirement in school. The objective of software planning is to provide a framework that enables the manager to make reasonable estimate made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points.

- ✓ A description of the background and context of the project and its relation to work already done in the area.
- ✓ Made statement of the aims and objectives of the project.
- ✓ The description of purpose, scope and applicability.
- ✓ We define the problem on which we are working in the project.
- ✓ We describe the requirement specification of the system and the actions that can be done on these things.
- ✓ We understand the problem domain and produce a model of the system, which describe operations that can be performed on the system.
- ✓ We included features and operation in detail, including screen layouts.
- ✓ Finally, the system is implemented and tested according to test cases.

FUTURE ENHANCEMENT

The system can be further enhanced in future by including the following features:

- By making the entire process online which helps the viewers to search movie tickets and do transactions.
- By implementing the facility for access for more than one admin where more than one admin can login and modify the details of the company.
- By implementing the feature of premium Stegan ID where the users can unlock many features.

The new system is designed such that these enhancements can be integrated with the current modules easily with less integration work. The enhanced features make project more interactive, more user friendly and can fulfill each users need in the best way possible.

BIBLIOGRAPHY

Books:

- ✓ Computer Science with C++ for class XII by Sumita Arora
- ✓ Computer Science with C++ for class XI by Sumita Arora
- ✓ CBSE All in One Computer Science CBSE Class 12 (Old edition).

Websites:

- ✓ www.google.com
- ✓ www.codeproject.com
- ✓ www.geeksforgeeks.org
- ✓ www.stackoverflow.com
- ✓ www.zetcode.com
- ✓ www.programiz.com