**Visvesvaraya Technological University**

# Belagavi, Karnataka-590 018

A Mini Project Report on

**“PARKING MANAGEMENT SYSTEM”**

Mini Project Report submitted in partial fulfillment of the requirement for the award of the degree of

## Bachelor of Engineering in Information Science and Engineering

Submitted by

**AYUSH SHEKHAR**

**1MJ16IS013**

Under the esteemed guidance of **Dr. Mannar Mannan J Associate Professor**



**M V J College of Engineering, Bengaluru-67 Department of Information Science and Engineering 2018-19**



**M V J College of Engineering, Bengaluru-67 Department of Information Science and Engineering 2018-19**

# CERTIFICATE

Certified that the project work titled **“SEMINAR HALL MANAGEMENT SYSTEM”** is a bonafide work carried out **AYUSH SHEKHAR(1MJ16IS013**) in partial fulfillment for the award of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2018-19. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report. The report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the Bachelor of Engineering Degree.

|  |  |  |
| --- | --- | --- |
| **Signature of Guide** | **Signature of HOD** | **Signature of Principal** |
| **Dr.Mannar Mannan** | **Mrs. Sanchari Saha** | **Dr. Nagaraj Sitaram** |
| **Associate Professor** | **Associate Professor & HOD** | **Principal** |
| **Dept. of ISE**  **MVJCE** | **Dept. of ISE**  **MVJCE** | **MVJCE** |

##### Sl.No Examiners Signature with date 1.

**2.**

**M V J College of Engineering, Bengaluru-67 Department of Information Science and Engineering 2018-19**

# DECLARATION

I, **AYUSH SHEKHAR(1mj16is013**) hereby declare that the dissertation entitled, **“SEMINAR HALL MANAGEMENT SYSTEM”** has been completed and written by me under the supervision of Dr.Mannar J Mannan , Associate Professor, Department of Information Science and Engineering, MVJCE, Bangalore, in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Information Science and Engineering, of Visvesvaraya Technological University, Belagavi.The dissertation report is original and it has not been submitted for any other degree in any university.

Date:

Place: AYUSH SHEKHAR

(1MJ16IS013)

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of this project report would be incomplete without the mention of the people who made it possible, without whose constant guidance and encouragement, would have made efforts go in vain. I consider myself privileged to express gratitude and respect towards all those who guided me through the completion of this project.

I, acknowledge our beloved Principal **Dr. Nagaraj Sitaram**, who has always been and shall always be a source of motivation to us.

I am grateful to **Mrs. Sanchari Saha**, Associate Professor and HOD, Information Science and Engineering, for giving me the support and encouragement that was necessary for the completion of this project. In this context, I would also like to thank all the other staff members, both teaching and non- teaching, who have extended their timely help and eased our task.

I also convey thanks to our project guide **Dr.Mannar J Mannan**, Associate Professor, Department of Information Science and Engineering, for providing encouragement, constant support and guidance which was of great help to complete this project work successfully.

I would like to express my heart-felt gratitude to my parents and friends for their continued moral and material support throughout the course and in helping us finalize the project report.

AYUSH SHEKHAR

# ABSTRACT

The main aim of this project is to book seminar hall and auditorium of college. Seminar halls are booked and its booking info is stored in a physical file. Whenever any department has to book seminar hall, officials has to check the file and then allot the seminar hall to the department. When a seminar hall has to be booked, it’s available time-slot also has to be checked and for this we need to search the dates and it’s respective schedule which is very tidious.

Seminar hall booking management system is an application written in c++ language which will be integrated with CERP management system of any institute. It can be used by HODs, Vice-Principal, Principal, Chairman of college to book seminar hall and auditorium. It will be convenient to maintain seminar hall booking status. Users can check the seminar hall booking status and book required time-slot based on availability and it will also avoid collision of time-slots. Higher authority can modify the time-slots. It also has the feature to search the seminar hall booking status on a particular day.

# TABLE OF CONTENTS

1. Acknowledgement I
2. Abstract II
3. Table of Contents III
4. List of Figures V
5. List of Tables V

|  |  |  |
| --- | --- | --- |
| **Sl No.** | **Title** | **Page No.** |
| 1 | Introduction | 1 |
|  | 1.1 File Structure | 1 |
| 2 | Software Requirements Specification | 4 |
|  | 2.1 Requirements | 4 |
|  | 2.1.1 Software Requirements | 4 |
|  | 2.1.2 Software Features | 4 |
|  | 2.1.3 Hardware Requirements | 6 |
| 3 | System Design | 7 |
|  | 3.1 Description Of The Program | 8 |
|  | 3.1.1 Top level DFD(Data Flow Diagram) | 8 |
|  | 3.2Description Of The Component | 8 |
| 4 | System Analysis | 9 |
|  | 4.1 Various divisions in the project | 9 |
|  | 4.1.1 Admin Process | 9 |
| 5 | Implementation | 12 |
|  | 5.1 Module Implementation | 12 |
|  | 5.1.1 Insert Record | 13 |
|  | 5.1.2 Delete Record | 14 |
|  | 5.1.3 Search Record | 15 |

1. Snapshots 21

|  |  |  |
| --- | --- | --- |
| 6.1 | Snapshot 1 | 21 |
| 6.2 | Snapshot 2 | 22 |
| 6.3 | Snapshot 3 | 22 |
| 6.4 | Snapshot 4 | 23 |
| 6.5 | Snapshot 5 | 23 |

Conclusion And Future Work 19

References 20

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Description** Data | **Page No.** |
| 3.1. | Flow Diagram Snapshot | 8 |
| 6.1 | 1: Home Page | 21 |
| 6.2 | Snapshot 2: Arrival of Vehicle | 22 |
| 6.3 | Snapshot 3: Display page | 22 |
| 6.4 | Snapshot 4: Record History | 23 |
| 6.5 | Snapshot 5: Departure of Vehicle | 23 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 1.1 | Operations Of File Structure | 2 |
| 1.2 | Different modes of class | 2 |
| 1.3 | Functions used in this project | 3 |
| 3.1 | Symbol Used In Data Diagram | 7 |

## Chapter 1

PARKING MANAGEMENT SYSTEM

**INTRODUCTION**

The project on “**SEMINAR HALL BOOKING MANAGEMENT SYSTEM**” is developed to manage the seminar hall booking of any college or institute. It is written in c++ language which will be integrated with CERP management system of any institute. It can be used by HODs, Vice-Principal, Principal, Chairman of college to book seminar hall and auditorium. It will be convenient to maintain seminar hall booking status. Users can check the seminar hall booking status and book required time-slot based on availability and it will also avoid collision of time-slots. Higher authority can modify the time-slots. It also has the feature to search the seminar hall booking status on a particular day

#### 1.1 File Structures:

A computer file is a computer resource for recording data discretely in a computer storage device. Just as words can be written to paper, so can information be written to a computer file. There are different types of computer files, designed for different purposes. A file may be designed to store a picture, a written message, a video, a computer program, or a wide variety of other kinds of data. Some types of files can store several types of information at once. By using computer programs, a person can open, read, change, and close a computer file. Computer files may be reopened, modified, and copied an arbitrary number of times. Typically, files are organised in a file system, which keeps track of where the files are located on disk and enables user access.

##### Input or Output with files:

C++ provides the following classes to perform output and input of characters to/from files: ofstream**:** Stream class to write on files.

ifstream**:** Stream class to read from files.

fstream**:** Stream class to both read and write from/to files.

Dept. Of ISE, MVJCE 2018-19 Page 1

These classes are derived directly or indirectly from the classes’ istream and ostream. We have already used objects whose types were these classes: cin is an object of class istream and cout is an object of class ostream.

##### Open File:

In order to open a file with a stream object we use its member function. open;open(filename,mode) , Where filename is a string representing the name of the file to be opened, and mode is an optional parameter with a combination of the following flags:

|  |  |
| --- | --- |
| ios::in | Open for input operations. |
| ios::out | Open for output operations. |
| ios::binary | Open in binary mode. |
| ios::app | All output operations are performed at the end of the file, appending the content to the current content of the file. |
| ios::trunc | If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one. |

##### Table 1.1 Operations Of File Structure.

Each of the open member functions of classes’ ofstream, ifstream and fstream has a default mode that is used if the file is opened without a second argument:

|  |  |
| --- | --- |
| **Class** | **default mode parameter** |
| ofstream | ios::out |
| Ifstream | ios::in |
| Fstream | ios::in | ios::out |

##### Table 1.2 Different modes of class.

* + - **Closing file:**

When we are finished with our input and output operations on a file we shall close it so that the operating system is notified and its resources become available again. For that, we call the stream's member function close. This member function takes flushes the associated buffers and closes the file:Myfile.close()

* Functions used in the project:

Function: Description:

|  |  |
| --- | --- |
| Write() | Used to write the booking status to a file as a record. |
| Modify() | Used to modify the seminar hall booking status. |
| Search() | Used to search the seminar hall booking status of a particular day. |
| File.open() | Used to open the file. |
| File.close() | Used to close the file. |
| Eof() | Denotes the end of the file. |
| Display() | Used to display the seminar hall booking status. |

##### Table 1.3 Functions used in this project

**Chapter 2**

**SYSTEM REQUIREMENT SPECIFICATION**

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Characteristics of good SRS:

* + Complete.
  + Consistent.
  + Feasible.
  + Modifiable.
  + Unambiguous.
  + Testable.
  1. **Requirements**

##### Software Requirements:

Operating System : UBUNTU / WINDOWS

Programming Language : C++

Tool Used : SUBLIME TEXT EDITOR

##### Software Features:

* + - * **C++**

C++ is a 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 a its design highlights. C++ has also been found useful in many other contexts, with key

strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes). C++ is a compiled language, with implementations of it available on many platforms. Many vendors provide C++ compilers, including the Free Software Foundation, Microsoft, Intel, and IBM.

##### SUBLIME TEXT

Sublime Text is a proprietary cross-platform source code editor with a Python application programming interface (API). It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses.

##### LINUX TERMINAL

The **Linux Terminal** is a [system console](https://en.wikipedia.org/wiki/System_console) internal to the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) (a system console is the device which receives all kernel messages and warnings and which allows logins in single user mode).[[2]](https://en.wikipedia.org/wiki/Linux_console#cite_note-2) The Linux console provides a way for the kernel and other processes to send text output to the user, and to receive text input from the user. The user typically enters text with a [computer keyboard](https://en.wikipedia.org/wiki/Computer_keyboard) and reads the output text on a [computer monitor](https://en.wikipedia.org/wiki/Computer_monitor). The Linux kernel supports [virtual consoles](https://en.wikipedia.org/wiki/Virtual_console) - consoles that are logically separate, but which access the same physical keyboard and display.[[3]](https://en.wikipedia.org/wiki/Linux_console#cite_note-3) The Linux console (and Linux virtual consoles) are implemented by the VT subsystem of the Linux kernel, and do not rely on any [user space](https://en.wikipedia.org/wiki/User_space) software.[[4]](https://en.wikipedia.org/wiki/Linux_console#cite_note-deprecating-4) This is in contrast to a [terminal emulator](https://en.wikipedia.org/wiki/Terminal_emulator), which is a user space process that emulates a terminal, and is typically used in a graphical display environment.

The Linux console was one of the first features of the kernel and was originally written by [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds) in 1991. There are two main implementations: [framebuffer](https://en.wikipedia.org/wiki/Linux_framebuffer" \o "Linux framebuffer) and [text mode](https://en.wikipedia.org/wiki/Text_mode). The framebuffer implementation is the default in modern [Linux distributions](https://en.wikipedia.org/wiki/Linux_distributions), and together with [kernel mode setting](https://en.wikipedia.org/wiki/Kernel_mode_setting), provides kernel-level support for display hardware and features such as showing graphics while the system is booting.[[6]](https://en.wikipedia.org/wiki/Linux_console#cite_note-fbcon.txt-6) The legacy text mode implementation was used in [PC-compatible](https://en.wikipedia.org/wiki/PC-compatible) systems with [CGA](https://en.wikipedia.org/wiki/Color_Graphics_Adapter), [EGA](https://en.wikipedia.org/wiki/Enhanced_Graphics_Adapter), [MDA](https://en.wikipedia.org/wiki/IBM_Monochrome_Display_Adapter) and [VGA](https://en.wikipedia.org/wiki/Video_Graphics_Array) graphics cards. Non-[x86](https://en.wikipedia.org/wiki/X86) architectures used framebuffer mode because their graphics cards did not implement text mode.[[6]](https://en.wikipedia.org/wiki/Linux_console#cite_note-fbcon.txt-6) The Linux console uses fixed-size [bitmap](https://en.wikipedia.org/wiki/Bitmap_fonts), [monospace fonts](https://en.wikipedia.org/wiki/Monospace_font" \o "Monospace font), usually defaulting to 8x16 pixels per character.[[6]](https://en.wikipedia.org/wiki/Linux_console#cite_note-fbcon.txt-6)

The Linux console is an optional kernel feature, and most [embedded Linux](https://en.wikipedia.org/wiki/Embedded_Linux) systems do not enable it. These systems typically provide an alternative user interface (e.g. web based), or boot immediately into a [graphical user interface](https://en.wikipedia.org/wiki/Graphical_user_interface) and use this as the primary means of interacting with the user. Other implementations of the Linux console include the [Braille](https://en.wikipedia.org/wiki/Braille) console to support [refreshable Braille displays](https://en.wikipedia.org/wiki/Refreshable_braille_display) and the [serial port](https://en.wikipedia.org/wiki/Serial_port) console.

##### Hardware Requirements:

* Processor : INTEL dual core and higher
* MEMORY : 2GB and higher
* HDD : Free space of 30 GB
* Platform : Windows / Ubuntu

## Chapter 3

### SYSTEM DESIGN

Design of the new system begins by elaborating the statement of requirements in terms of more detailed objectives. The main aim of design process is to produce a model or representation of the system, which can be used later to bind the system. The produced model is called design of the system. A system design is a top down approach to minimize complexity and make a problem manageable by subdividing it into smaller segments.

The most changing phase of the system development of life cycle is system design. It refers to the technical specification that will be applied in implementing the candidate system. The potential objects are thoroughly analyzed. Class hierarchies are to check whether the system is behaving the way it has to. There after the classes are individually tested and subsequently they are integrated from the overall system.

##### SOME OF THE SYMBOLS USED IN DATAFLOW DIAGRAM

|  |  |
| --- | --- |
| Processes | The process shows a transformation or manipulation of dataflow with in a system. A process transforms in coming data  flow into outgoing data flow. |
| File | A file is a holding place for information within the system it is represented by an open ended narrow  rectangle. |
| External entity | External entities are outside the system but they either supply input data into the system or use the system output. External  entities |

are represented by rectangle.

##### Table 3.1 Symbols used in DFD.

* 1. **DESCRIPTION OF THE PROGRAM**
     1. **DFD (Data Flow Diagram):**

DFD shows the functional component in the software package. Each component shown in the top level DFD is described in the subsections of 3.1

SEMINAR HALL BOOKING

MANAGEMENT SYSTEM

DISPLAY THE STATUS OF SEMINAR HALL

ARRIVAL OF VEHICLES

SEMINAR HALL

DEPARTURE OF VEHICLES

CHECK RECORDS HISTORY

##### Fig 3.1 Data Flow Diagram

* + 1. **Description of the components:**

**Description**: Seminar hall booking management system is an application written in c++ language which will be integrated with CERP management system of any institute. It can be used by HODs, Vice-Principal, Principal, Chairman of college to book seminar hall and auditorium. It will be convenient to maintain seminar hall booking status. Users can check the seminar hall booking status and book required time-slot based on availability and it will also avoid collision of time-slots. Higher authority can modify the time-slots. It also has the feature to search the seminar hall booking status on a particular day

## Chapter 4

### SYSTEM ANALYSIS

Analysis is the process of breaking a complex topic or substance into smaller parts to gain a better understanding of it. Analysts in the field of engineering look at requirements, structures, mechanisms, and systems dimensions. Analysis is an exploratory activity.

Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements.

#### Various divisions in the project

##### USERS

* + There are multiple users :- HOD of every department, Vice-Principal, Principal, Chairman.
* **SEMINAR HALL** 
  + Users have option to select the seminar hall or auditorium they want to access.
* **DISPLAY** 
  + This option displays all the bookings related to a particular seminar hall.
* **WRITE**
  + This option lets user book a seminar hall. If there is availability then only seminar hall can be booked.
* **MODIFY**
  + This option lets users modify the seminar hall bookings done by them and even higher ups can modify the seminar hall
* **SEARCH**
  + This option let user to search the bookings of a seminar hall on a particular day.

#### Feasibility Study

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development .Feasibility and risk analysis and related in many ways. If a project risk is great and feasibility of producing software is reduced. During the feasibility analysis in this project has been discussed below in the abovementioned topics.

#### Operational Feasibility

Feasibility of the working of the system after the installation in the organization as mentioned in the feasibility analysis.

#### Technical Feasibility

Technical feasibility is frequently the most difficult area to ensure this stage. It is essential that the process of analysis and definition to be conducted parallel to an assessment of the technical feasibility. The consideration that is normally associated with technical feasibility includes the resources availability of the Organization where the project is to be developed and implemented. By taking these facts into consideration before developing the resource availability at Retail Outlet of Hindustan Petroleum was observed. As very limited resources are required for this project hence this project is considered feasible for development.

#### Economic Feasibility

An evaluation of development cost is weighted against the ultimate income or benefits derived from the developed system. There was no need of extra hardware and software for development of this project. Hence this project has economically justified for development in this organization.

#### Motivational Feasibility

An evaluation of the probability that the organization is sufficient motivation to support the development and implementation of the application with necessary user participation, resources, training etc. The interest and support shown by the organization during the system study do not seem that the new system developed to have efficient support from the organization.

#### Schedule Feasibility

An evaluation of the time needed for the development of this project. The time schedule required for the development of this project is very important, since more development time effects machine time, costs and delays in the development of the other systems. So the project should be complete within affixed schedule time as far as the organization is concerned.

## Chapter 5

### IMPLEMENTATION

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In other words, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment. Many implementations may exist for a given specification or standard.

#### Module implementation

Pseudo code is an informal high-level description of the operating principle of a computer program or other algorithm. The purpose of using pseudo code is that it is easier for people to understand than conventional programming language code, and that it is an efficient and environment-independent description of the key principles of an algorithm. It is commonly used in textbooks and scientific publications that are documenting various algorithms, and also in planning of computer program development, for sketching out the structure of the program before the actual coding takes place.

Below code shows the pseudo code for this project.

* + - Write data

void seminar1::writedata(int a,char file\_name[30])

{

fstream fo;

int j;

record r;

int tot\_slots\_count=0;

r.id=a;

int count=0;

record s;

char date\_list[100][15];

char time\_list[100][15];

int starttime[20],endtime[20];

int i=0;

fo.open(file\_name,ios::in|ios::binary);

if(fo)

{

while(1){

fo.read((char\*)&s,sizeof(s));

if(fo.eof())

{

break;

}

strcpy(date\_list[i],s.date);

strcpy(time\_list[i],s.time);

sscanf(time\_list[i],"%d-%d",&starttime[i],&endtime[i]);

i++;

}

}

fo.close();

cout<<"\nEnter Date (DD/MM/YYYY):";

cin>>r.date;

cout<<"\ntime slots taken on this dates are:";

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

{

if(strcmp(date\_list[j],r.date)==0)

{

cout<<"\n"<<j+1<<"."<<"\t"<<time\_list[j]<<"\n";

tot\_slots\_count++;

}

}

if(tot\_slots\_count==0)

{

cout<<"\t no time slots booked";

}

int flag=0;

while(flag==0)

{

cout<<"\nEnter Time Slot (From - To):\t";

cin>>r.time;

int starttime1,endtime1;

sscanf(r.time,"%d-%d",&starttime1,&endtime1);

if(starttime1<8 || starttime1>16 || endtime1>16 || endtime1<8 || starttime1==endtime1 || endtime1<starttime1)

{

cout<<"thats not working time. try again \n";

}

else if(tot\_slots\_count==0)

{

flag=1;

break;

}

else{

count=0;

for(int k=0;k<i;k++)

{

if((starttime[k]<starttime1 && endtime[k]<endtime1) ||(starttime[k]>starttime1 && endtime[k]>endtime1)||(endtime1<=starttime[k] && starttime1<starttime[k])||(starttime1>=endtime[k] && endtime1>endtime[k]))

{

count=count+1;

}

else

{

count=count-1;

}

}

if(count>0)

{

flag=1;

}

else{

cout<<"\nseminar hall will be busy during this time slot\n";

}

}

}

cout<<"\nEnter Branch:";

cin>>r.branch;

cout<<"\nEnter Discription:";

cin>>r.discription;

fstream f;

f.open(file\_name,ios::app|ios::binary);

f.write((char\*)&r,sizeof(r));

cout.flush();

f.close();

}

* + - Display

void seminar1::display(char file\_name[30])

{

fstream f;

record r;

char date\_list[100][15];

char time\_list[100][15];

int i=0;

int c=1;

f.open(file\_name,ios::in|ios::binary);

if(!f){

cout<<"no records found";

}

else{

cout << '|' << setw(4) << "s.no" << '|'<< setw(15) << "date" << '|'<< setw(10) << "time" << '|'<< setw(10) << "branch" << '|'<< setw(15) << "description" << '|'<< setw(16) << "id" << '|' << endl;

while(1){

f.read((char\*)&r,sizeof(r));

if(!f.eof())

{ cout.flush();

cout << '|' << setw(4) << c++ << '|'<< setw(15) << r.date << '|'<< setw(10) << r.time << '|'<< setw(10) << r.branch << '|'<< setw(15) << r.discription << '|'<< setw(16) << official(r.id) << '|' << endl;

strcpy(date\_list[i],r.date);

strcpy(time\_list[i],r.time);

i++;

}

else{

break;

}

}

f.close();

}

}

* + - Search

void seminar1::search(char file\_name[30])

{

fstream f;

record r;

char date[15];

cout<<"\n\t\t\t\tenter date you want to search :\t";

cin>>date;

char date\_list[100][15];

char time\_list[100][15];

char description[100][100];

char branch[100][10];

int id[100];

int i=0;

int c=1;

f.open(file\_name,ios::in|ios::binary);

if(!f){

cout<<"no records found";

}

else{

while(1){

f.read((char\*)&r,sizeof(r));

if(!f.eof())

{ if(strcmp(r.date,date)==0)

{

strcpy(date\_list[i],r.date);

strcpy(time\_list[i],r.time);

strcpy(branch[i],r.branch);

strcpy(description[i],r.discription);

id[i]=r.id;

i++;

}

}

else{

break;

}

}

if(i==0)

{

cout<<"\n\t\t\t\tno booking on this date";

}

else

{

cout << '|' << setw(4) << "s.no" << '|'<< setw(15) << "date" << '|'<< setw(10) << "time" << '|'<< setw(10) << "branch" << '|'<< setw(15) << "description" << '|'<< setw(16) << "id" << '|' << endl;

for(int k=0;k<i;k++)

{

cout << '|' << setw(4) << c++ << '|'<< setw(15) << date\_list[k] << '|'<< setw(10) <<time\_list[k] << '|'<< setw(10) << branch[k] << '|'<< setw(15) << description[k] << '|'<< setw(16) << official(id[k]) << '|' << endl;

}

}

f.close();

}

}

* Main()

int main()

{

int ch;

int hall;

int type;

char pass[10];

char enterpass[10];

char chairman[10]="chairman";

char vp[10]="vp";

char principal[10]="principal";

cout<<"\n\t\t\t1. HOD\n";

cout<<"\n\t\t\t2. VICE-PRINCIPAT\n";

cout<<"\n\t\t\t3. PRINCIPAL\n";

cout<<"\n\t\t\t4. CHAIRMAN\n";

cout<<"\n";

cout<<"\n\t\t\t"<<"enter your choice:\t";

cin>>type;

if(type==1)

{

cout<<"\n\t\t\t1. HOD ISE\n";

cout<<"\n\t\t\t2. HOD CSE\n";

cout<<"\n\t\t\t3. HOD MECH\n";

cout<<"\n\t\t\t4. HOD CIV\n";

cout<<"\n\t\t\t5. HOD AERO\n";

cout<<"\n";

cout<<"\n\t\t\t"<<"enter your choice:\t";

cin>>ch;

if(ch==1)

{

strcpy(pass,"hodise");

}

else if(ch==2)

{

strcpy(pass,"hodcse");

}

else if(ch==3)

{

strcpy(pass,"hodmech");

}

else if(ch==4)

{

strcpy(pass,"hodciv");

}

else if(ch==5)

{

strcpy(pass,"hodaero");

}

cout<<"\n\t\t\tenter password:\t";

cin>>enterpass;

int id=(type\*10)+ch;

while(strcmp(enterpass,pass)!=0)

{

cout<<"\n\t\t\tenter correct password:\t";

cin>>enterpass;

}

login(id);

}

else if(type==2)

{

cout<<"\n\t\t\tenter password:\t";

cin>>enterpass;

while(strcmp(enterpass,vp)!=0)

{

cout<<"\n\t\t\tenter correct password:\t";

cin>>enterpass;

}

login(type);

}

else if(type==3){

cout<<"\n\t\t\tenter password:\t";

cin>>enterpass;

while(strcmp(enterpass,principal)!=0)

{

cout<<"\n\t\t\tenter correct password:\t";

cin>>enterpass;

}

login(type);

}

else if(type==4){

cout<<"\n\t\t\tenter password:\t";

cin>>enterpass;

while(strcmp(enterpass,chairman)!=0)

{

cout<<"\n\t\t\tenter correct password:\t";

cin>>enterpass;

}

login(type);

}

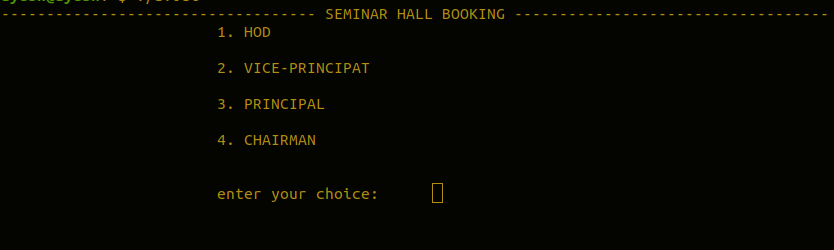
}

## Chapter 6

### SNAPSHOTS AND RESULT DISCUSSION

In this chapter we will present some of the snapshots of this project.

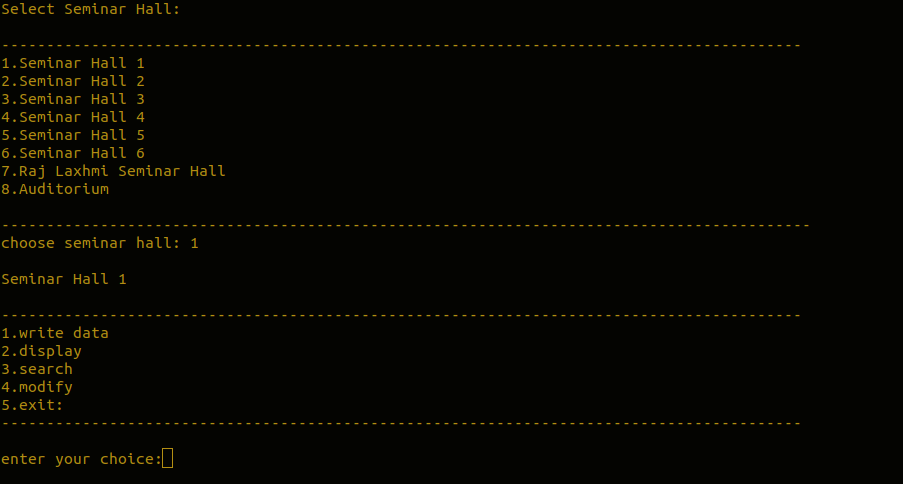
#### Snapshot 1

****

##### Fig 6.1 Index Page

Fig 6.1 This is the index page for seminar hall booking management system

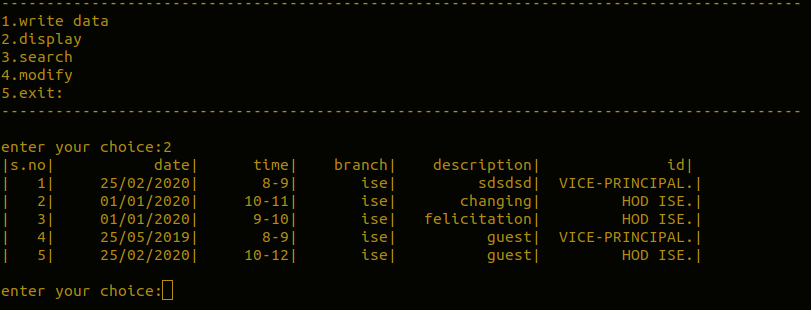
#### Snapshot 2

****

##### Fig 6.2 Seminar hall Selection

Fig 6.2 Shows the menu for selecting seminar hall and its corresponding option

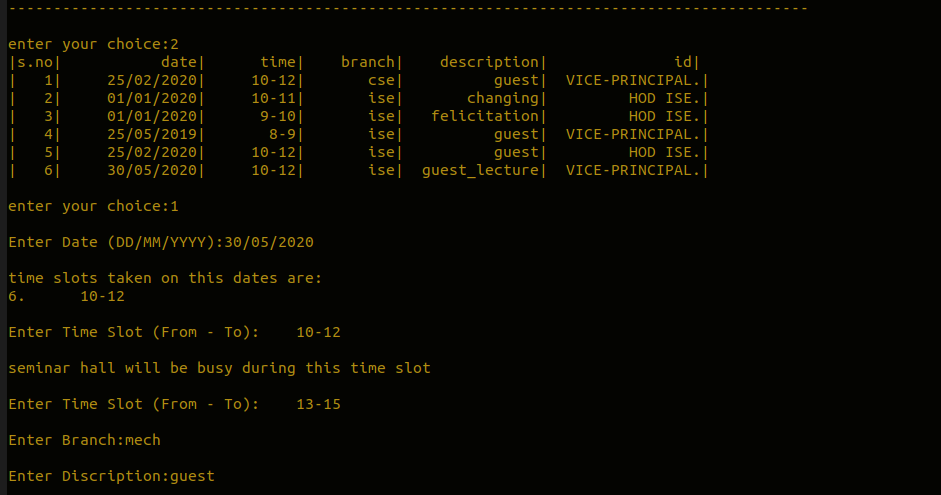
#### Snapshot 3

****

##### Fig 6.3 Display Page

Fig 6.3 Shows the page that displays the bookings related to a particular seminar hall

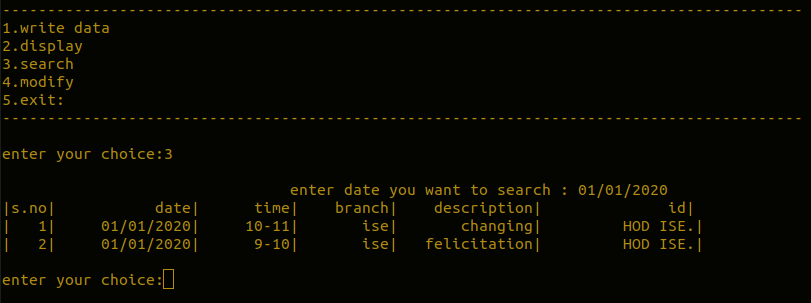
#### Snapshot 4

****

##### Fig 6.4 Booking Seminar hall

Fig 6.4: Shows the page to book seminar hall based on availability of time slots.

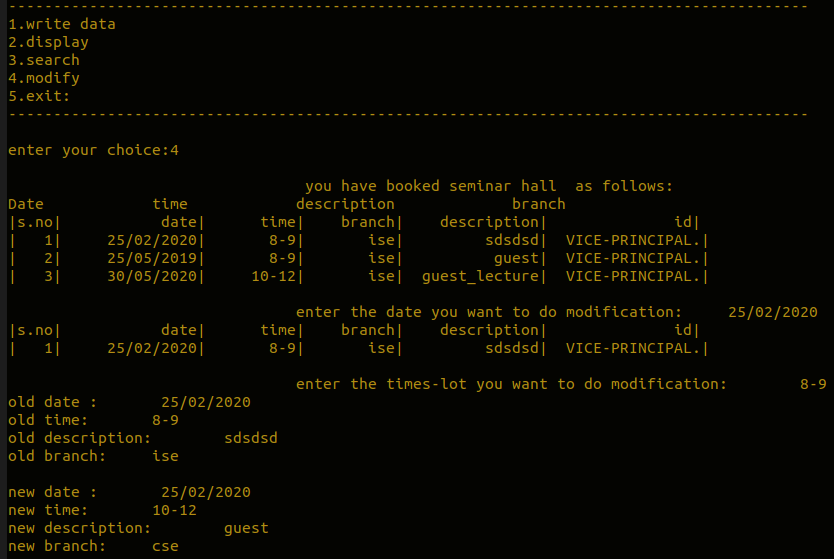
#### Snapshot 5

****

##### Fig 6.5 Search

Fig 6.5: Shows the page to search for the bookings of a particular date

#### Snapshot 6

****

##### Fig 6.6 Modify

Fig 6.5: Shows the page to modify the seminar hall booking status

### CONCLUSION AND FUTURE WORK

With the theoretical inclination of our syllabus it becomes very essential to take the utmost advantage of any opportunity of gaining practical experience that comes along. The construction of this Minor Project **“**SEMINAR HALL BOOKING MANAGEMENT SYSTEM” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development: The planning that goes into implementing a project. The importance of proper planning and an organized methodology. The project also provided us the opportunity of interacting with our teachers and to gain from their vast experience.

There is always room for improvement, and the software we created can also be improved. This is especially because we had to create it within a limited time. With more time, the software can be improved to include security and different types of users. The project can be improved in various ways like addition of features of taking approvals of higher authorities before final booking.

## REFERENCES

1. <https://www.tutorialspoint.com/>
2. https://www.geeksforgeeks.org/
3. [http://www.stackoveflow.com](http://www.stackoveflow.com/)