



# FILM MANAGEMENT SYSTEM



## A MINI PROJECT REPORT

*Submitted by*

**ABINAYA M**                   **(REG.NO: 9517202109002)**

**BHUVANIKA S**                   **(REG.NO: 9517202109011)**

**RAJAKUMARI S**                   **(REG.NO: 9517202109042)**

*In partial fulfillment for the award of the degree  
of*

**BACHELOR OF TECHNOLOGY**  
**IN**  
**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI**

**ANNA UNIVERSITY: CHENNAI 600 025**

**DECEMBER 2022**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**FILM MANAGEMENT SYSTEM**” is the bonafide work of “**Abinaya M (Reg.No:9517202109002), Bhuvanika S (Reg.No: 9517202109011), Rajakumari S (Reg.No: 9517202109042)**” who carried out the mini project work under my supervision.

### **SIGNATURE**

**Dr.A.Shenbagarajan,M.E.,Ph.D**

**Mrs.L.Prasika. M.E.,(Ph.D)**

Assistant Professor(SG)

Artificial Intelligence and Data Science

Mepco Schlenk Engineering College

Sivakasi – 626 005

Virudhunagar District.

### **SIGNATURE**

**Dr.J.Angela Jennifa Sujana, M.E.,Ph.D**

**Associate Professor(SG) & Head**

Artificial Intelligence and Data Science

Mepco Schlenk Engineering College,

Sivakasi – 626 005.

Virudhunagar District.

Submitted for the project viva-voce examination to be held on \_\_\_\_\_.

### **INTERNAL EXAMINER**

### **EXTERNAL EXAMINER**

## **TABLE OF CONTENTS**

<b>CH.NO</b>	<b>TITLE</b>	<b>PG.NO</b>
1	ABSTRACT	5
	LIST OF FIGURES	6
	INTRODUCTION	9
	1.1 Overview	9
	1.2 Characteristics	9
	1.3 Application	11
2	LITERATURE REVIEW	13
	2.1 Existing System	13
	2.2 Limitations of Existing System	13
	2.3 Proposed System	14
3	SYSTEM DESIGN	15
	3.1 Description	15
	3.2 ER Diagram	15
	3.3 Shematic Diagram	16
4	SYSTEM REQUIREMENT	17
	4.1 Hardware Requirement	17
	4.2 Software Requirement	17
	4.2.1 Software Characteristics	18
	4.2.2 Some Features of Idle	23
	4.2.3 Application of Idle	23
	4.2.4 Introduction to MySQL	24

4.2.5 Access in MySQL	24
4.2.6 Salient Features of MySQL	24
4.3 Module Description	27
<b>5 SYSTEM IMPLEMENTATION</b>	
5.1 Tabular Explanation	31
5.3 List of Tables	32
5.4 Normalisation	34
<b>6 RESULT</b>	35
<b>7 CONCLUSION AND FUTURE WORK</b>	47
<b>APPENDIX</b>	49
<b>REFERENCES</b>	64

## **ABSTRACT**

The film management automation system will storing of the booking records, booking information, customer records, customer information a lot easier. The booking and customer information in the database will automatically generate booking details when the customer booked for movie. The booking details can be retrieved promptly. And reports will be generated based on different criteria's.

The data is directly stored in the database in the hard disk of the Film Management System is application software designed to take advantage of today's technology and reduce or avoid the burden of storing data on paper and in files. This facilities booking details, customer details, and movie information. Booking information can be generated as needed. Each booking details has an option to store their data on one remote central database server. This will also allow stores to access information from other partner.

## **LIST OF FIGURES**

<b>FIG.NO</b>	<b>TITLE</b>	<b>PAGE NO</b>
3.2	ER Diagram	15
3.3	Schematic Diagram	16
6.1	Main Page	35
6.2	Movie List	36
6.3	Info	36
6.6	Facility	38
6.7	Booking Form	38
6.10	Facility Form	40

## **ACKNOWLEDGEMENT**

First and foremost we **praise and thank “The Almighty”**, the lord of all creations, who by his abundant grace has sustained us and helped us to work on this project successfully.

We really find unique pleasure and immense gratitude in thanking our respected management members, who is the backbone of our college.

A deep bouquet of thanks to respected Principal **Dr.S.Arivazhagan M.E.,Ph.D.**, for having provided the facilities required for our mini project.

We sincerely thank our Head of the Department **Dr. J. Angela Jennifa Sujana M.E.,Ph.D.**, Associate Professor(SG) & Head, Department of Artificial Intelligence and Data Science, for her guidance and support throughout the mini project .

We also thank our guide **Dr.A.Shenbagarajan.,M.E.,Ph.D.**, Assistant Professor(SG), **Mrs.L.Prasika., M.E(Ph.D)**, Assistant Professor Department of Artificial Intelligence and Data Science for their valuable guidance and it is great privilege to express our gratitude to them.

We extremely thank our project coordinator **Dr.A.Shenbagarajan.,M.E.,Ph.D.**, Assistant Professor(SG), **Mrs.L.Prasika., M.E(Ph.D)**, **Assistant Professor** Department of Artificial Intelligence and Data Science, who inspired us and supported us throughout the mini project.

We extend our heartfelt thanks and profound gratitude to all the faculty members of Artificial Intelligence and Data Science department for their kind help during our mini project work.

We also thank our parents and our friends who had been providing us with constant support during the course of the mini project work.

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1 OVERVIEW**

Film production is the process of film making which involves or includes various numbers of distinct levels. Film production takes place across the world with the wide range of social, political, and economic perspective by using diversity of cinematic techniques and advance technologies. Usually it involves a large number of human resources for their projects. And for creating and maintaining the quality it needs to have enormous efforts from very skilled people. And when it comes for communicating or coordinating the tasks like feedback from stakeholders across the world or across the time zone or any other things or tasks it becomes hard to look after or control the whole setup so it needs to have proper implementation of main elements of Project Management (Remidez, & Jones, 2012).

And for that we need to know what the main elements of Project Management are? As we all know that project management is the series of several important steps or activities which are interrelated and complex for attainment of the single goal. And the important aspect of booming project management is to see and consider the whole process interlinked. Completion of project may face constraints if it's not proper planned because without proper planning, risk management, WBS successful execution of film production cannot be possible (Remidez, & Jones, 2012). These elements are very necessary for the film production's successful execution.

### **1.2 CHARACTERISTICS**

Everything you need to know about the characteristics of management. An efficient management can make the employees to work efficiently and an inefficient management can break the employees and can bring industrial unrest in the business organization.

1. Management is Intangible
2. Management is Goal Oriented
3. Management is Pervasive
4. Management is Universal
5. Management is a Social Process
6. Management is a Group Activity
7. Management is a System of Authority
8. Management is Dynamic
9. Management is Transferable
10. Management is Contingent in Nature
11. Synthesis
12. Ever-Evolving Phenomenon
13. Multidisciplinary
14. Ephemeral
15. Management as a Career
16. Management as an Applied Science
17. Divorced from Proprietorship

18. Management is a Profession

19. Management is Decision-Making

20. Coordination is the Soul of Management

21. Management is a Science as well as an Art

22. Managing Results through Others.

### **1.3 APPLICATIONS OF PYTHON**

- Web Development
- Game Development
- Scientific and numeric applications
- AI and ml
- Desktop GUI
- Software development
- Enterprise level/business application
- Education programs and training courses
- Language development
- OS
- Web scraping applications
- Image processing and graphic design applications
- Audio and video applications
- Cad application
- Embedded application
- Console based application

- Data science and data visualization
- Network programming
- Database access

#### **1.4. APPLICATIONS OF DBMS**

- Railway reservation systems
- Library management system
- Banking
- Education sector
- Credit card exchanges
- Social media sites
- Broadcast communication
- Accounting and finance
- Online shopping
- Human resource management
- Manufacturing
- Airline reservation system
- Hotel management system
- Sales and marketing
- Industry
- Telecom
- Healthcare management system
- Agriculture management system
- Defense
- Personal cloud storage

# **CHAPTER 2**

## **LITERATURE REVIEW**

### **2.1 EXISTING SYSTEM**

As we know manual systems are quite tedious, time consuming and less efficient and accurate in comparison to the computerized system. So following are some disadvantages of the old system.

- Time consuming
- Less accurate
- Less efficient
- Lot of paper work
- Slow data processing
- Not userfriendly environment
- Difficult to keep old records

### **2.2 LIMITATIONS OF EXISTING SYSTEM**

1. Management techniques and policies should be adjusted according to specific circumstances. One principle may be good for one enterprise, but it may not be suitable for another enterprise. Likewise, a technique may be extremely useful in England, but it may be unworkable in India.

2. Principles of management are not static in nature. The concepts about management changes with the development of science and technology. New ideas are innovated, new products being put on the market, new likes and dislikes are developing every year. So what was successful in 2015 may lead to failure in 2016. Thus, a great deal of adjustment is to be done to cope with the changing times.

3. Management is concerned with human element in an organization. Different groups and different persons even in the same group, behave differently under different circumstances. This human aspect of management provides the greater challenge to its scientific treatment.

### **2.3 PROPOSED SYSTEM**

- The main objective of this system is to keep records of the complete ticket booking.
- It support for Film Management and designing system helps you record and track materials on the basis of both quantity and value.
- It improves cash flow, visibility, and decision making.
- For warehouse management, you can track quantity and value of all your materials, perform ticket booking, and optimize your warehouse resources.

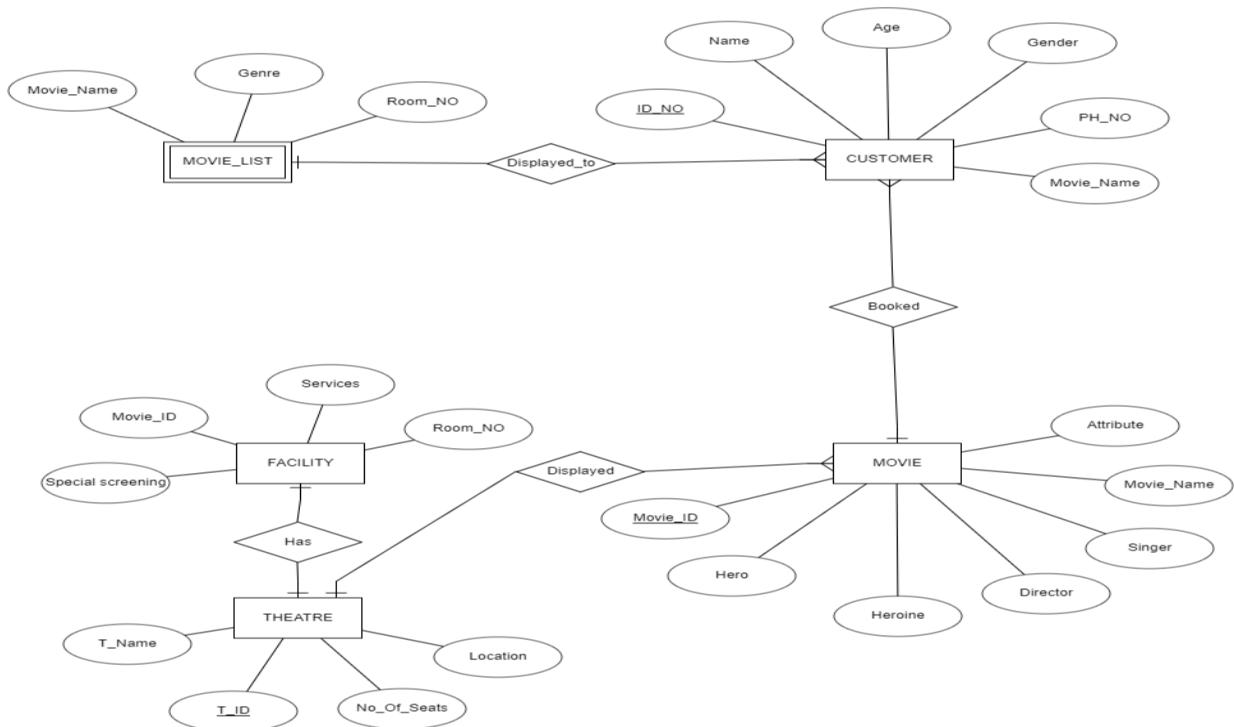
# CHAPTER 3

## SYSTEM DESIGN

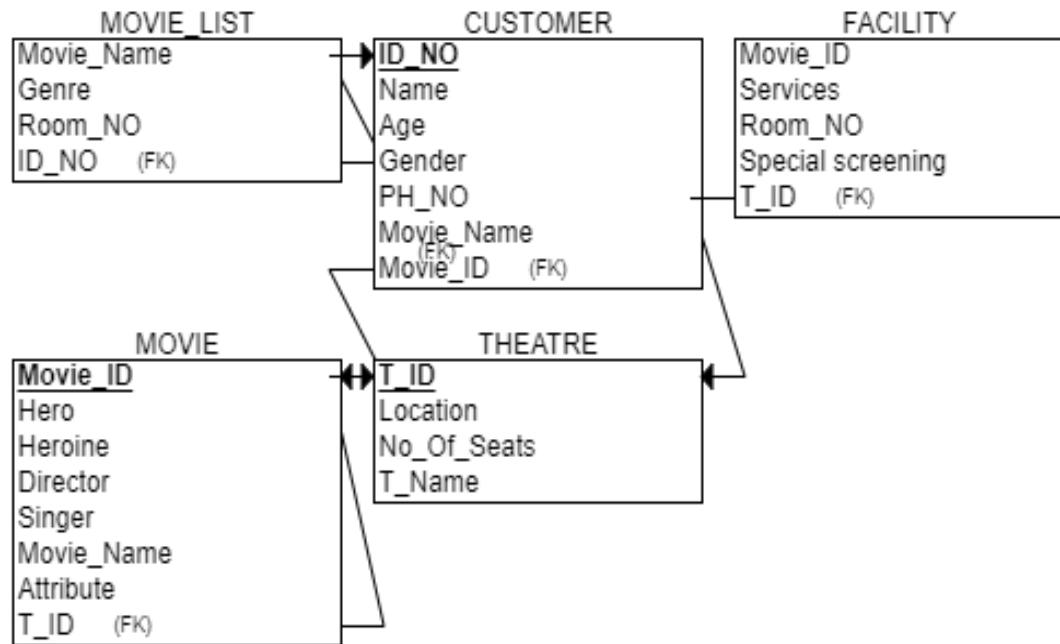
### 3.1 DESCRIPTION

**Movie Ticket Booking Management System** is an application that can automate the way of booking the movie tickets with great ease. The movie is one of the leisure time things that everyone enjoys watching. But due to the heavy work schedule, people may not find time to go to the theatres and watch a movie.

### 3.2 ER DIAGRAM



### 3.3 SCHEMATIC DIAGRAM



## **CHAPTER 4**

### **SYSTEM REQUIREMENT**

#### **4.1 HARDWARE REQUIREMENTS**

- Processor required: Minimum 2.0 GHz
- Hard-Disk space required: 80 GB of available hard-disk space
- RAM required: Minimum 1 GB
- Display: 1024×768 or higher-resolution display
- Other Hardware: DVD-ROM Drive

#### **4.2 SOFTWARE REQUIREMENTS**

Front End/Language : Idle

Back End/Language : MySQL

## **4.2.1 SOFTWARE CHARACTERISTICS**

### **TOOLS, PLATFORM/LANGUAGES USED**

#### **FRONT END**

##### **Python - IDLE**

IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for Python. The Python installer for Windows contains the IDLE module by default.

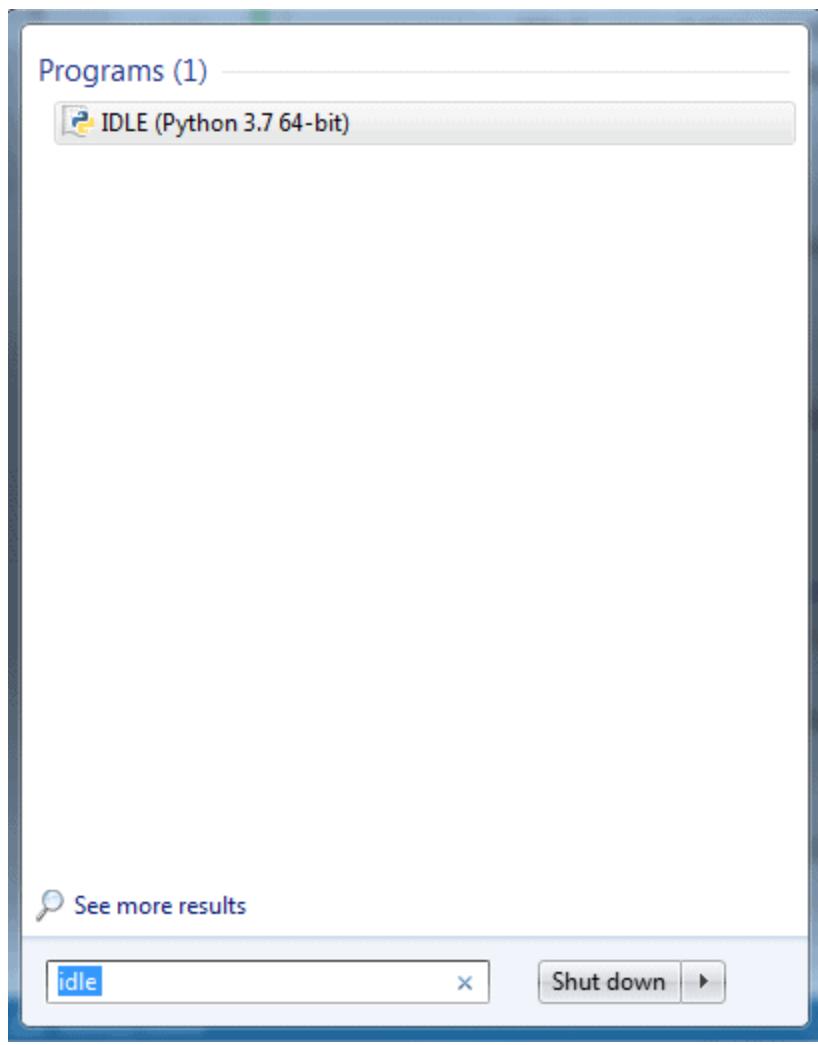
IDLE is not available by default in Python distributions for Linux. It needs to be installed using the respective package managers. Execute the following command to install IDLE on Ubuntu:

```
$ sudo apt-get install idle
```

IDLE can be used to execute a single statement just like Python Shell and also to create, modify, and execute Python scripts. IDLE provides a fully-featured text editor to create Python script that includes features like syntax highlighting, autocompletion, and smart indent. It also has a debugger with stepping and breakpoints features.

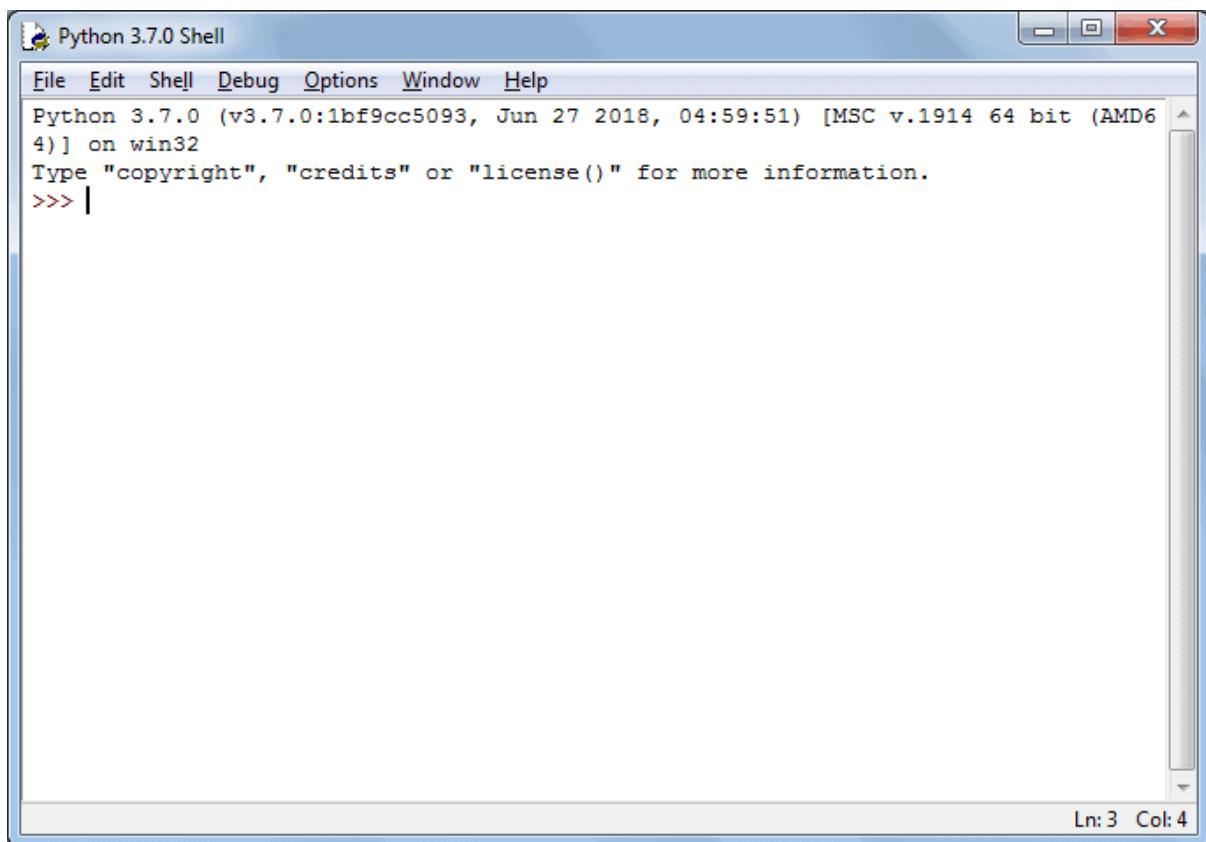
To start an IDLE interactive shell, search for the IDLE icon in the start menu and double click on it

Python IDLE



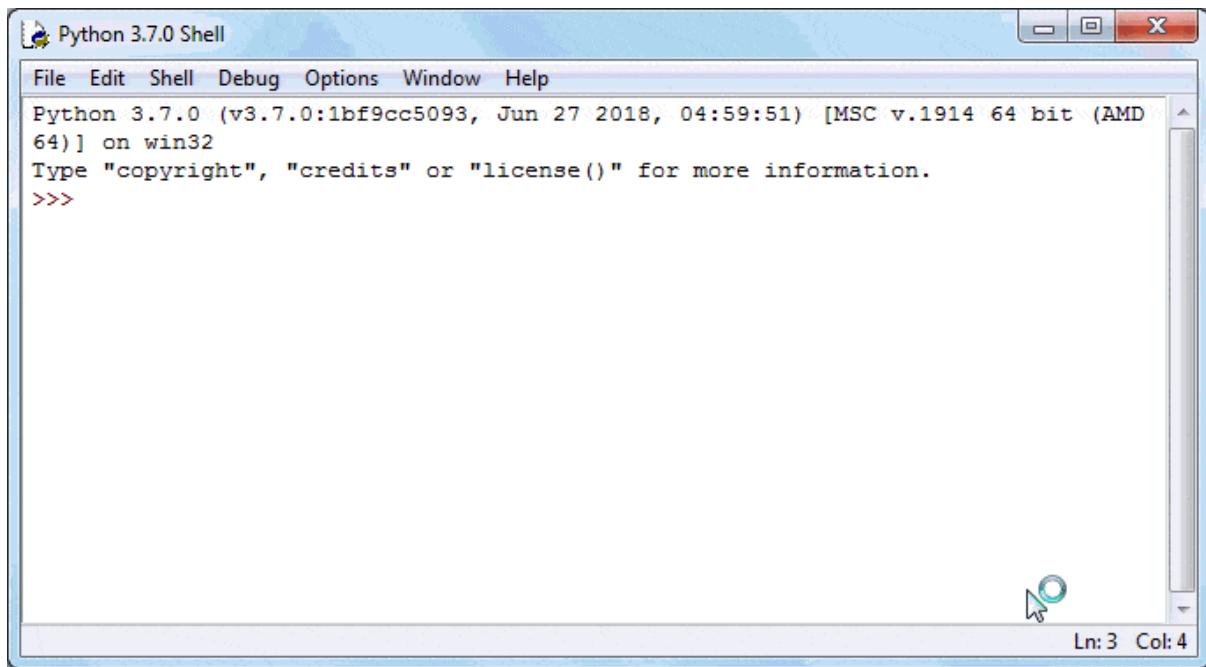
Python IDLE

This will open IDLE, where you can write and execute the Python scripts, as shown below.



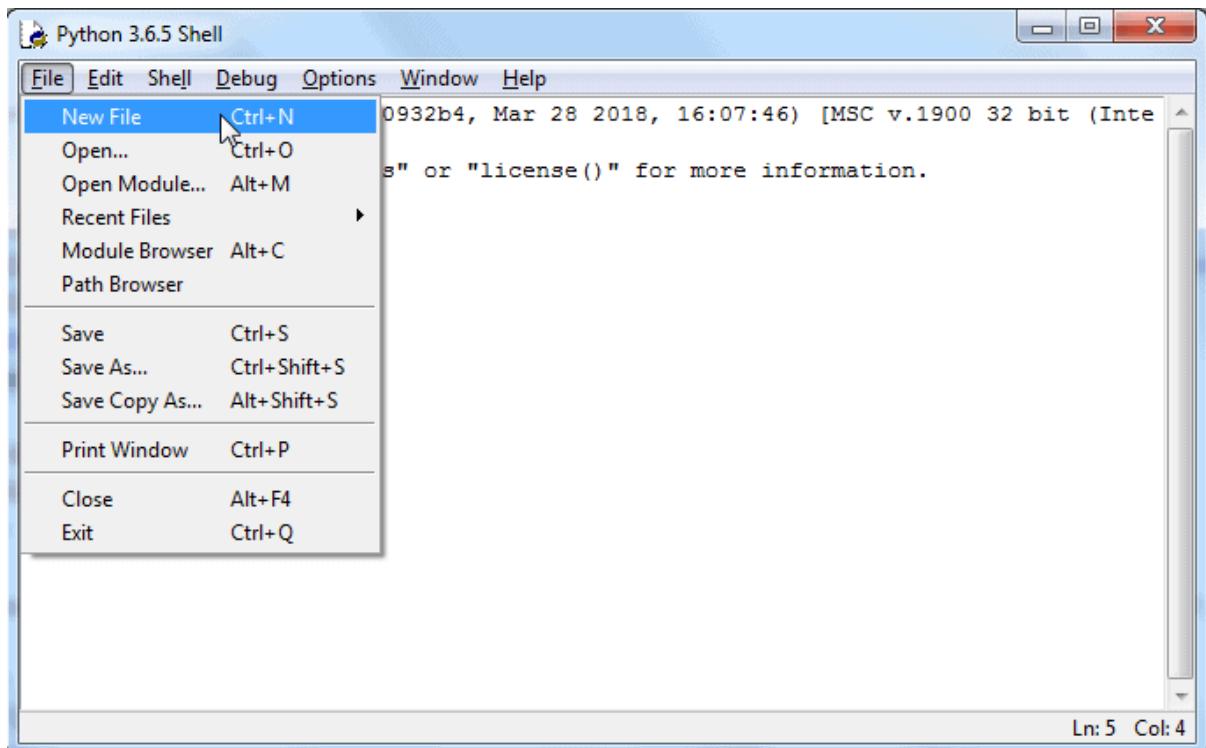
Python IDLE

You can execute Python statements same as in Python Shell as shown below.

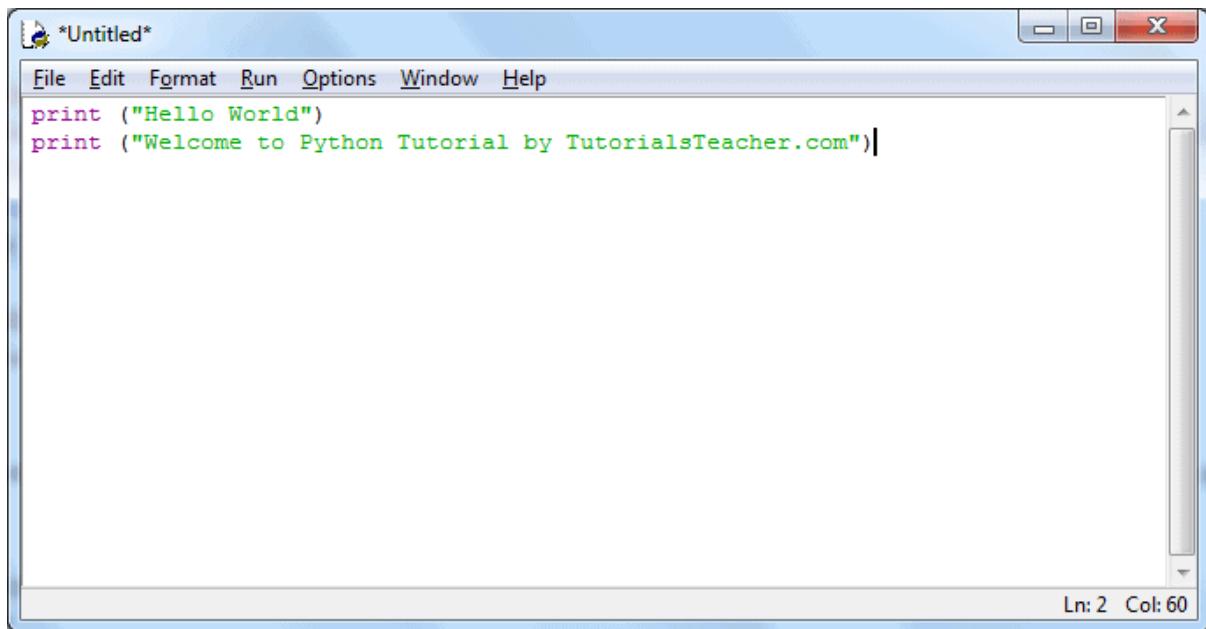


Python IDLE

To execute a Python script, create a new file by selecting File -> New File from the menu.



Enter multiple statements and save the file with extension .py using File -> Save. For example, save the following code as hello.py.



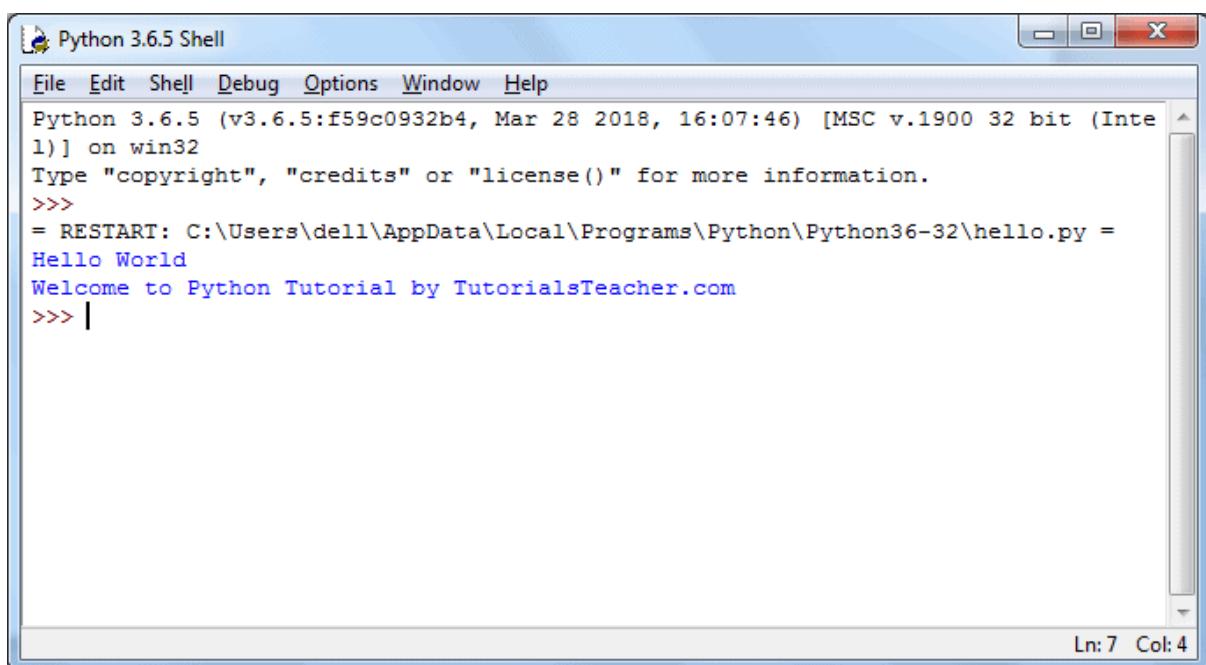
The screenshot shows the Python 3.6.5 IDLE editor window. The title bar says "Python 3.6.5 Shell". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The main code area contains the following Python code:

```
print ("Hello World")
print ("Welcome to Python Tutorial by TutorialsTeacher.com")|
```

The status bar at the bottom right indicates "Ln: 2 Col: 60".

Python Script in IDLE

Now, press F5 to run the script in the editor window. The IDLE shell will show the output.



The screenshot shows the Python 3.6.5 IDLE shell window. The title bar says "Python 3.6.5 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell area displays the following output:

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 16:07:46) [MSC v.1900 32 bit (Inte
1) ] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\dell\AppData\Local\Programs\Python\Python36-32\hello.py =
Hello World
Welcome to Python Tutorial by TutorialsTeacher.com
>>> |
```

The status bar at the bottom right indicates "Ln: 7 Col: 4".

Python Script Execution Result in IDLE

Thus, it is easy to write, test and run Python scripts in IDLE.

## **4.2.2 Some Features of Idle**

- Coded in 100% pure Python, using the tkinter GUI toolkit
- Cross-platform: works mostly the same on Windows, Unix, and macOS
- Python shell window (interactive interpreter) with colorizing of code input, output, and error messages
- Multi-window text editor with multiple undo, Python colorizing, smart indent, call tips, auto completion, and other features
- Search within any window, replace within editor windows, and search through multiple files (grep)
- Debugger with persistent breakpoints, stepping, and viewing of global and local namespaces
- Configuration, browsers, and other dialogs

## **4.2.3 APPLICATIONS OF IDLE**

- Forms-Windows that you create for user interface
- Controls-Graphical features drawn on forms to allow user interaction (text boxes, labels, scroll bars, command buttons, etc.) (Forms and Controls are objects.)
- Properties-Every characteristic of a form or control is specified by a property. Example properties includes names, captions, size, color, position, and contents. Idle applies default properties.
- Methods-Built-in procedure that can be invoked to impart some action to a particular object.
- Event Procedures-Code related to some object. This is the code that is executed when a certain event occurs.
- General Procedures-Code not related to objects. This code must be invoked by the application.
- Modules-Collection of general procedures, variable declarations, and constant definitions used by application.

#### **4.2.4 INTRODUCTION TO MySQL AS A BACKEND TOOL**

MySQL is the world's most popular opensource database. With its proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.

Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

#### **4.2.5 ACCESS IN MYSQL**

1. Log into your Linux web server via Secure Shell.
2. Open the MySQL client program on the server in the /usr/bin directory.
3. Type in the following syntax to access your database:

```
$ mysql -h {hostname} -u username -p {database name}
```

Password: {your password}

hostname: the name of the MySQL server that you are assigned to, for example, mysql4.safesecureweb.com

database name: the name of your MySQL database

password: the password you use to access your MySQL database

#### **4.2.6 SALIENT FEATURES OF MySQL**

##### **Relational Database Management System (RDBMS)**

MySQL is a relational database management system. This database language is based on the SQL queries to access and manage the records of the table.

## **Easy to use**

MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.

## **It is secure**

MySQL consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.

## **Client/ Server Architecture**

MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.

## **Free to download**

MySQL is free to use so that we can download it from MySQL official website without any cost.

## **It is scalable**

MySQL supports multi-threading that makes it easily scalable. It can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, we can increase this number to a theoretical limit of 8 TB of data.

## **Speed**

MySQL is considered one of the very fast database languages, backed by a large number of the benchmark test.

## **High Flexibility**

MySQL supports a large number of embedded applications, which makes MySQL very flexible.

## **Compatible on many operating systems**

MySQL is compatible to run on many operating systems, like Novell NetWare, Windows\*, Linux\*, many varieties of UNIX\* (such as Sun\* Solaris\*, AIX, and DEC\* UNIX), OS/2, FreeBSD\*, and others. MySQL also provides a facility that the clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

## **Allows roll-back**

MySQL allows transactions to be rolled back, commit, and crash recovery.

## **Memory efficiency**

Its efficiency is high because it has a very low memory leakage problem.

## **High Performance**

MySQL is faster, more reliable, and cheaper because of its unique storage engine architecture. It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory.

## **High Productivity**

MySQL uses Triggers, Stored procedures, and views that allow the developer to give higher productivity.

## **Platform Independent**

It can download, install, and execute on most of the available operating systems.

## **Partitioning**

This feature improves the performance and provides fast management of the large database.

## **GUI Support**

MySQL provides a unified visual database graphical user interface tool named "**MySQL Workbench**" to work with database architects, developers, and Database Administrators. MySQL Workbench provides SQL development, data modeling, data migration, and comprehensive administration tools for server configuration, user administration, backup, and many more. MySQL has a fully GUI supports from MySQL Server version 5.6 and higher.

## **Dual Password Support**

MySQL version 8.0 provides support for dual passwords: one is the current password, and another is a secondary password, which allows us to transition to the new password.

## **4.3 MODULE DESCRIPTION**

### **Product:**

In this project we use this module to add booking details. We use booking detail in bill of material process and also use in movie details.

### **Movie Detail:**

In this project we use this module to view the movie details. We use ticket availability details and ticket sales details also.

## **Bill of Material:**

In this project we use this module to ticket product to customer. We use bill of material process in sale the ticket to customer and available movies to the customer also.

## **Payment Detail:**

In this project we use this module to view the payment details. We use use ticket payment details of ticket details are show in this form.

## **Security:**

In this project we use this module to change the user password. We use this security as a tool in this form

# CHAPTER 5

## 5.1 TABULAR EXPLANATION

Booking (idno , name, age, gender, phone, movie\_name)

```
MySQL 8.0 Command Line Cli  ×  + | ▾
5 rows in set (0.01 sec)

mysql> desc booking;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| idno       | varchar(20) | NO   | PRI  | NULL    |       |
| name        | varchar(20) | YES  |       | NULL    |       |
| age         | varchar(20) | YES  |       | NULL    |       |
| gender      | varchar(20) | YES  |       | NULL    |       |
| phone       | varchar(20) | YES  |       | NULL    |       |
| movie_name  | varchar(20) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Movie(movie\_id, hero, heroine, director, singer, movie\_name, genre)

```
0 rows in set (0.00 sec)

mysql> desc movie;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| movie_id   | varchar(20) | NO   | PRI  | NULL    |       |
| hero        | varchar(20) | YES  |       | NULL    |       |
| heroine     | varchar(20) | YES  |       | NULL    |       |
| director    | varchar(20) | YES  |       | NULL    |       |
| singer      | varchar(20) | YES  |       | NULL    |       |
| movie_name  | varchar(20) | YES  |       | NULL    |       |
| genre       | varchar(20) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Showdetails(show\_time, show\_date, facility, snacks)

```
mysql> desc showdetails;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| show_time | varchar(20) | YES |   | NULL    |       |
| show_date | date     | YES |   | NULL    |       |
| facility  | varchar(20) | YES |   | NULL    |       |
| snacks    | varchar(20) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Facility(service available, room\_no)

```
mysql> desc facility;
+-----+-----+-----+-----+-----+
| Field      | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| service_available | varchar(20) | YES |   | NULL    |       |
| room_no     | varchar(20) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## 5.2 LIST OF TABLES

### MOVIE\_LIST

Field Name	Data Type	Constraints
Movie_Name	VARCHAR(20)	NOT NULL
Genre	VARCHAR(20)	NOT NULL
Room_NO	VARCHAR(20)	NOT NULL

## **BOOKING**

Field Name	Data Type	Constraints
ID_NO	VARCHAR(20)	PRI
Name	VARCHAR(20)	NOT NULL
Age	VARCHAR(20)	NOT NULL
Gender	VARCHAR(20)	NOT NULL
PH_NO	VARCHAR(20)	NOT NULL
Movie_Name	VARCHAR(20)	NOT NULL

## **THEATRE**

Field Name	Data Type	Constraints
Movie_ID	VARCHAR(20)	NOT NULL
Services	VARCHAR(20)	NOT NULL
Room_NO	VARCHAR(20)	NOT NULL

## **SHOW\_DETAILS**

Field Name	Data Type	Constraints
Show time	VARCHAR(20)	NOT NULL
Show date	VARCHAR(20)	NOT NULL
Facility	VARCHAR(20)	NOT NULL
Snacks	VARCHAR(20)	NOT NULL

## MOVIE

Field Name	Data Type	Constraints
Movie_ID	VARCHAR(20)	PRI
Hero	VARCHAR(20)	NOT NULL
Heroine	VARCHAR(20)	NOT NULL
Director	VARCHAR(20)	NOT NULL
Singer	VARCHAR(20)	NOT NULL
Movie_Name	VARCHAR(20)	NOT NULL
Genre	VARCHAR(20)	NOT NULL

## 5.3 NORMALISATION

### 1NF:

The first step in confirming 1NF is modifying multivalued columns to make sure that each column in a table does not take more than one entry.

idno	name	age	gender	phone	movie_name
1	bhu	18	Female	9876543210	anjaan
2	abi	17	Female	9765432180	Master
3	Raj	17	Male	9486491087	96
4	Rajakumari	17	Female	8903621576	love today
5	dharshini	16	Female	0987654321	NGK

The table is in 1NF. It has not multivalued columns.

## 2NF:

- The relation (table) must be in ***First Normal Form (1NF)***.
- Every **non-key attribute** must **fully depend** on the **key attribute** (i.e., the primary key).
- +-----+-----+-----+-----+-----+
- | idno | name | age | gender | phone | movie\_name |
- +-----+-----+-----+-----+-----+
- | 1 | bhu | 18 | Female | 9876543210 | anjaan |
- | 2 | abi | 17 | Female | 9765432180 | Master |
- | 3 | Raj | 17 | Male | 9486491087 | 96 |
- | 4 | Rajakumari | 17 | Female | 8903621576 | love today |
- | 5 | dharshini | 16 | Female | 0987654321 | NGK |

This table is in 2NF. It has only one primary key (idno) and non-key attribute fully depend on the key attribute.

## 3NF:

*The* relation is in 2NF. It should not include some transitive partial dependency.

(idno->name)

(idno->phno)

(idno->name->age)

(idno->name>movie name)

It is in 3NF. It should not include transitive dependency.

## **BCNF:**

A relation is in BCNF, X is a super key for every functional dependency (FD).

idno   name   age   gender   phone   movie_name
1   bhu   18   Female   9876543210   anjaan
2   abi   17   Female   9765432180   Master
3   Raj   17   Male   9486491087   96
4   Rajakumari   17   Female   8903621576   love today
5   dharshini   16   Female   0987654321   NGK

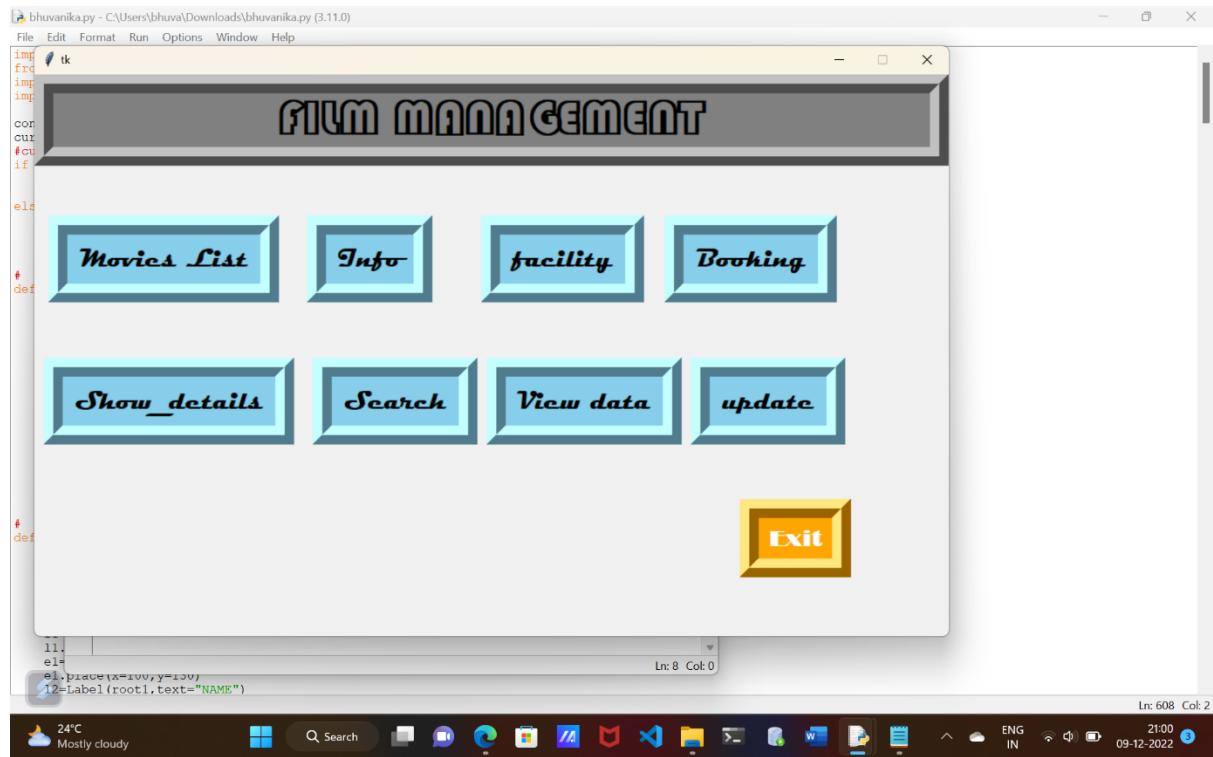
Here idno is a primary key. All other non-prime attributes depends only on a idno. So, idno is a super key.

It is in BCNF.

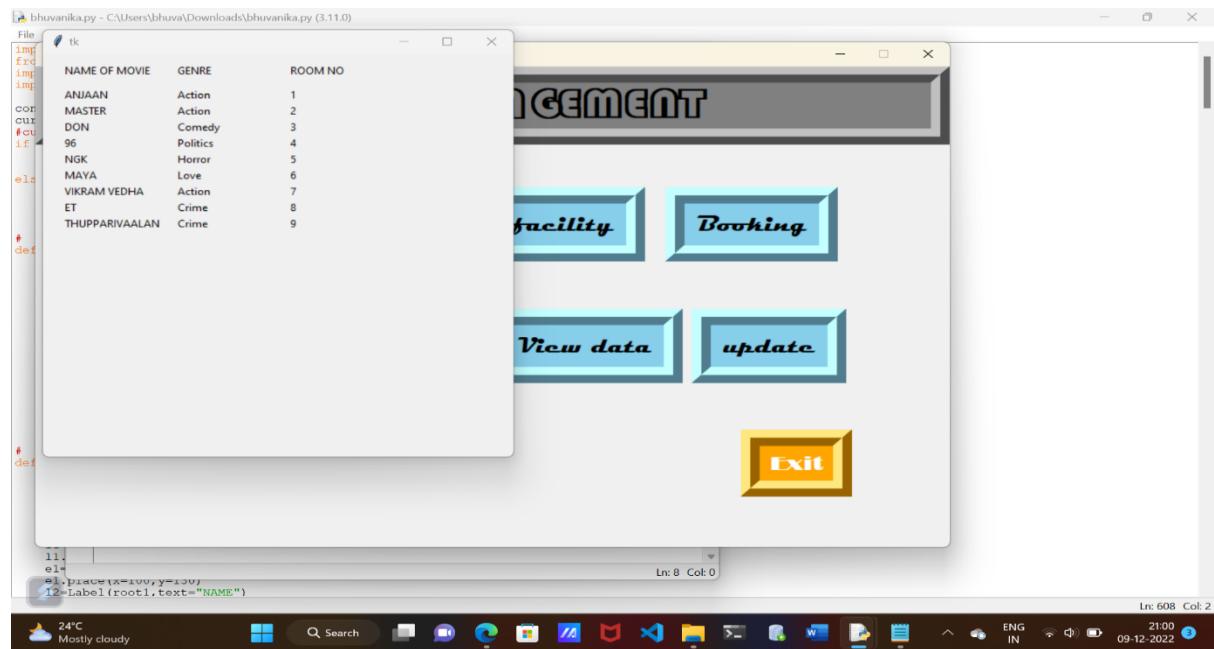
CHAPTER 6

## RESULT

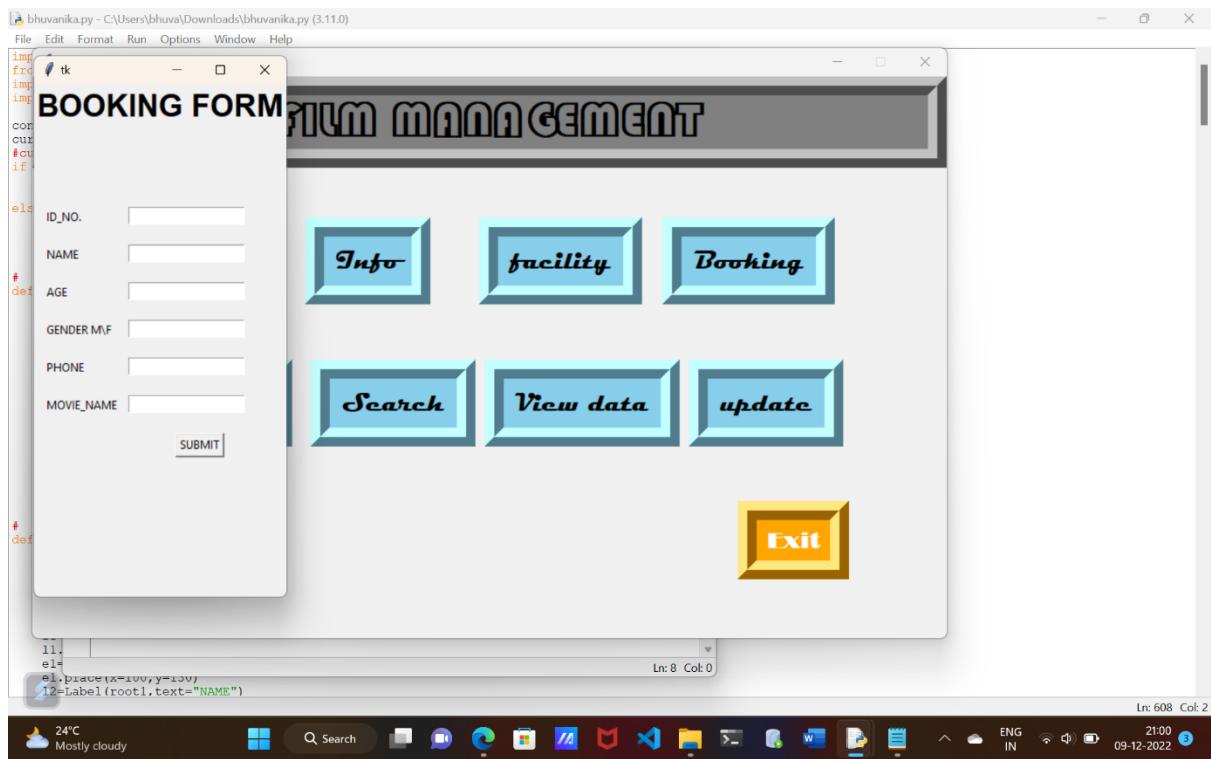
## 6.1 MAIN PAGE



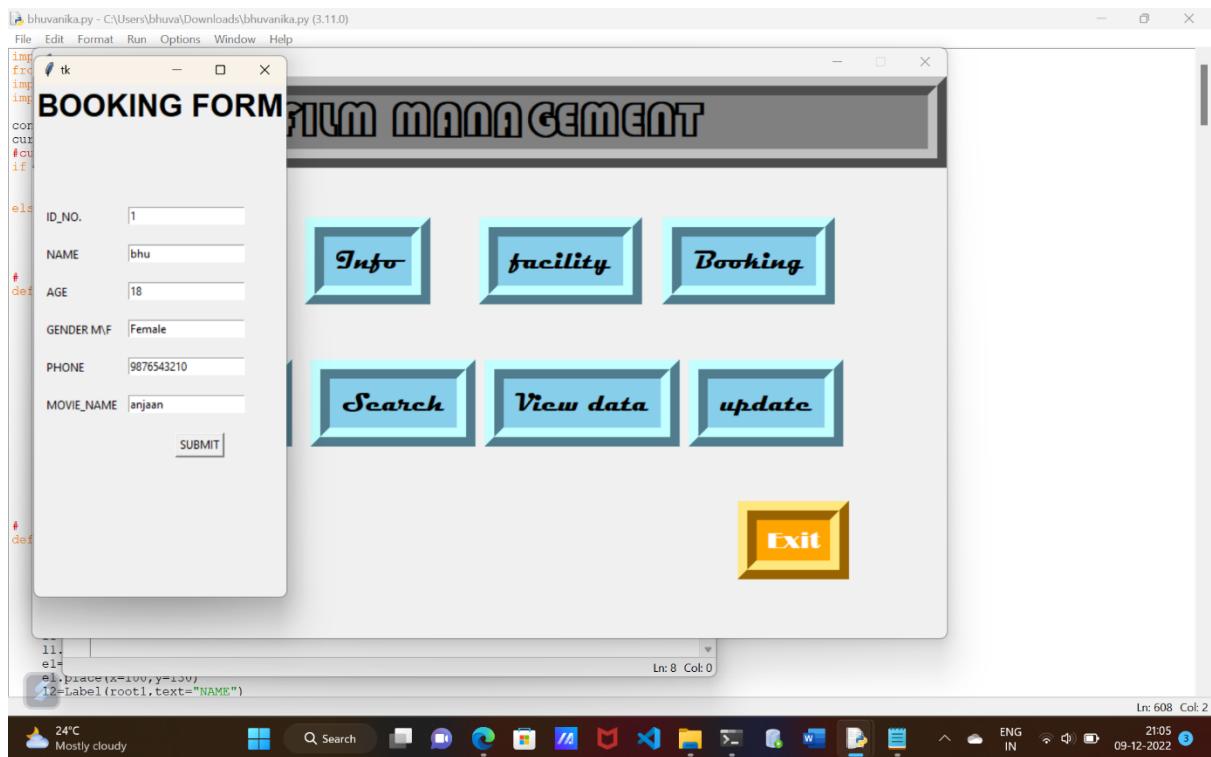
## 6.2 MOVIE LIST



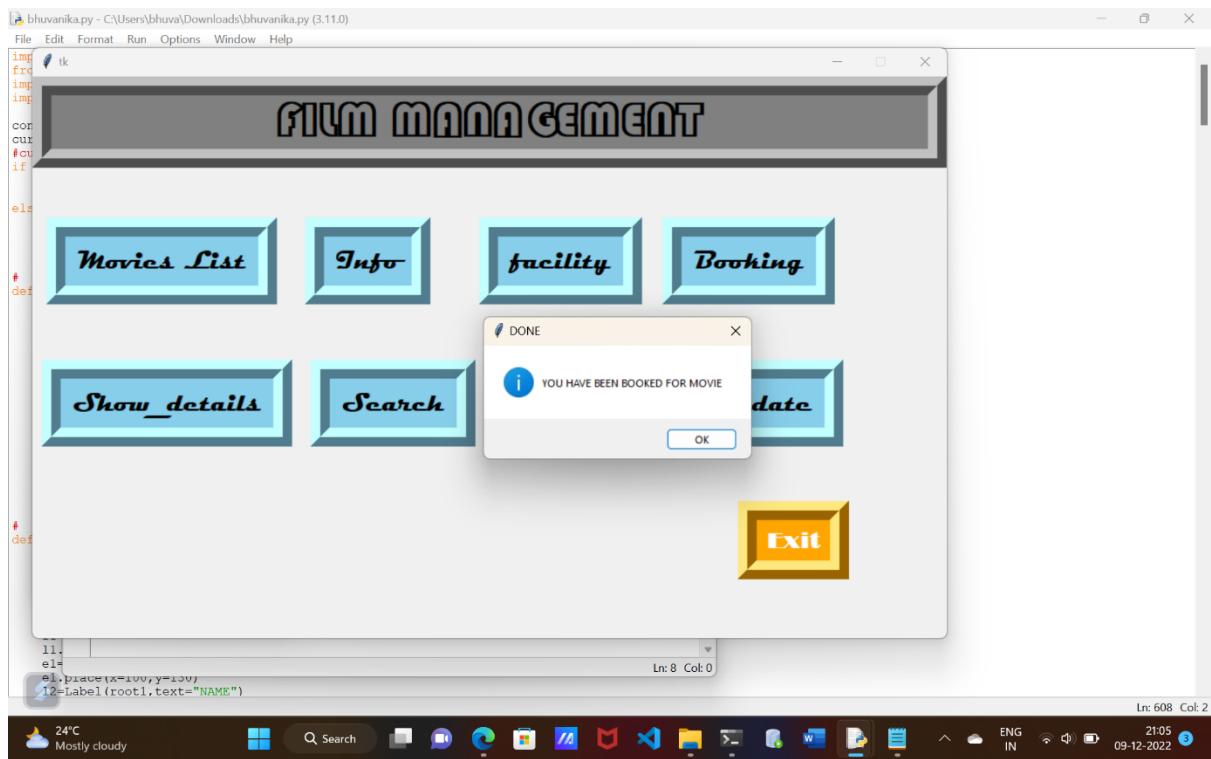
## 6.3 INFO



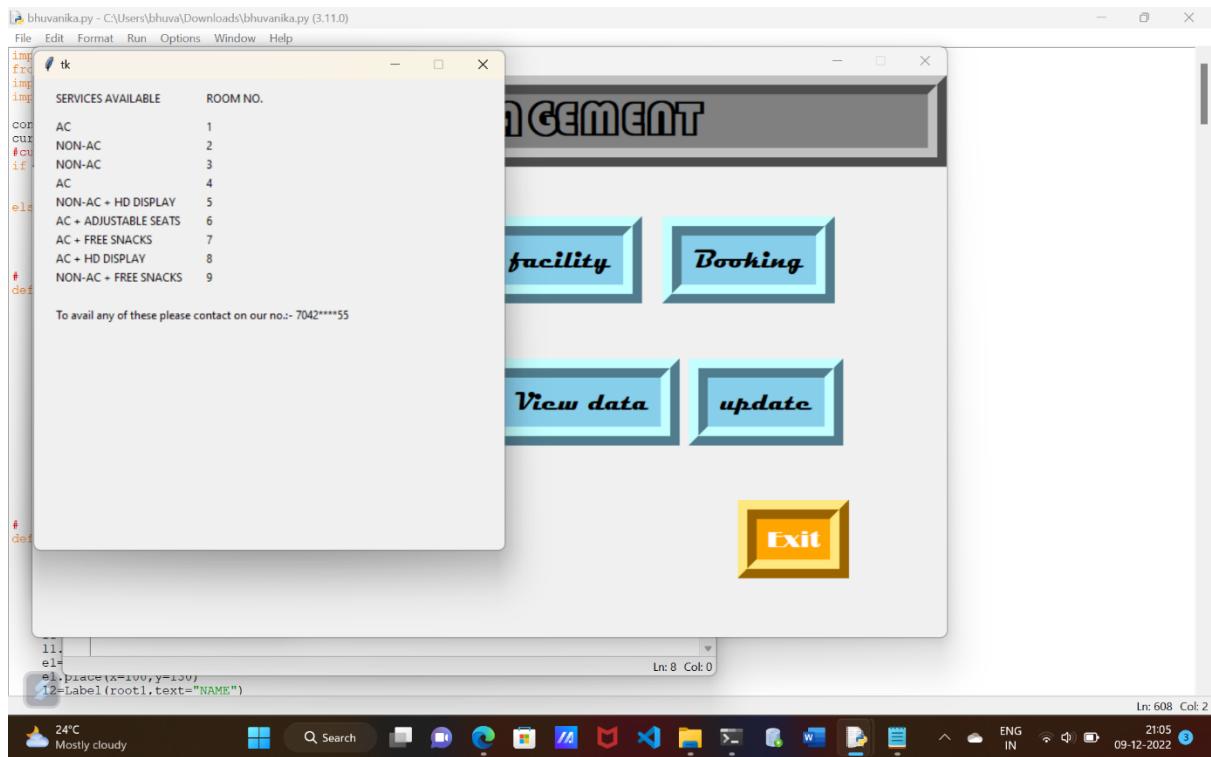
## 6.4 INSERT IN TO BOOKING FORM



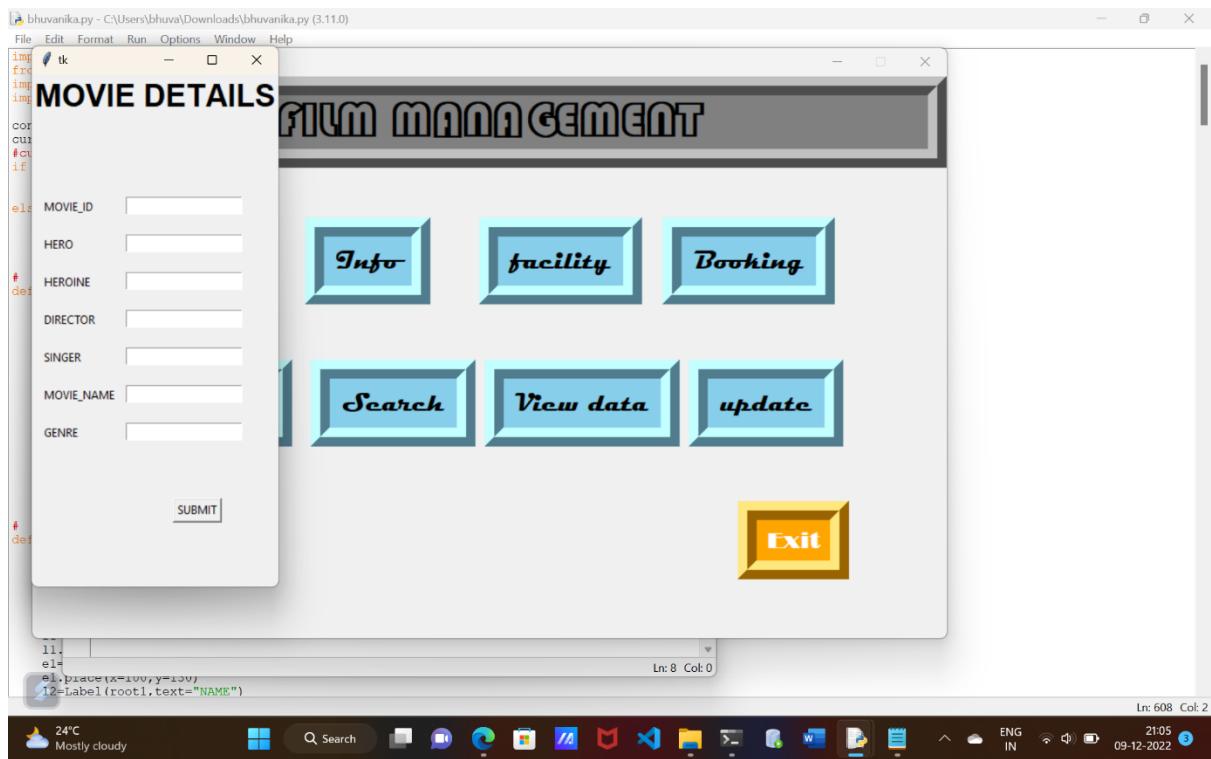
## 6.5 INSERTED



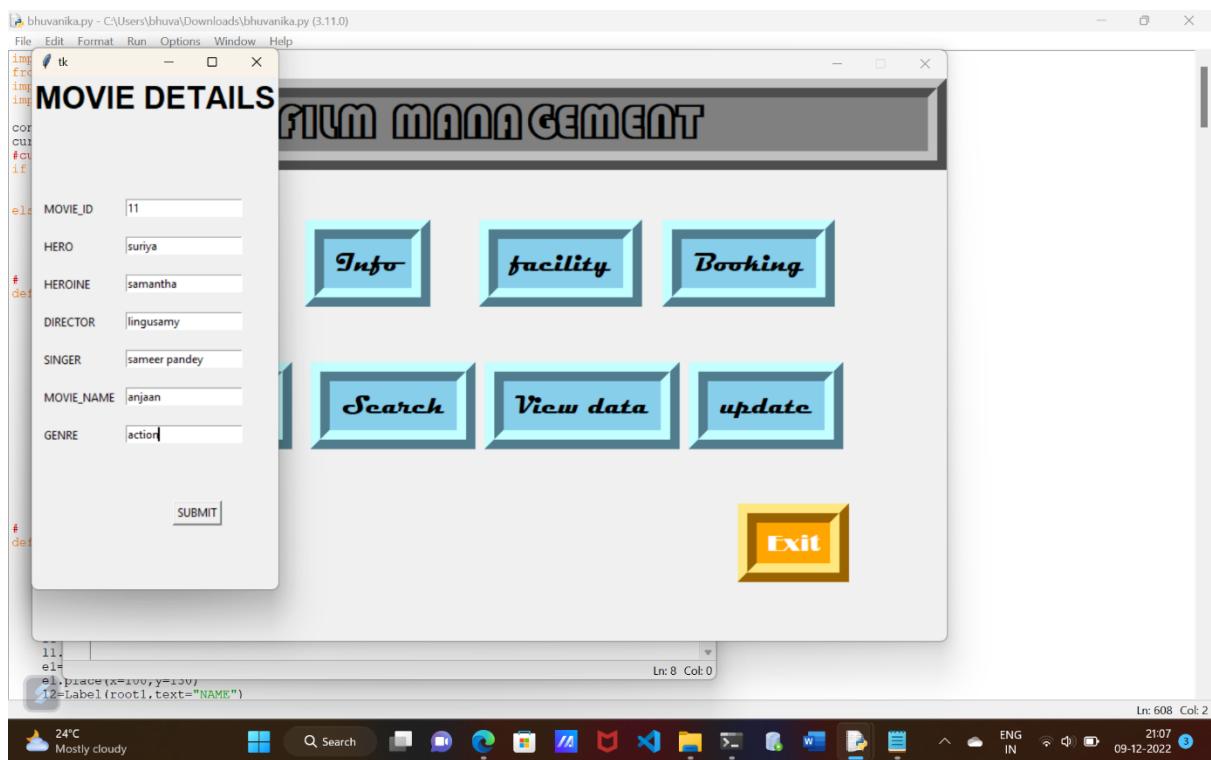
## 6.6 FACILITY



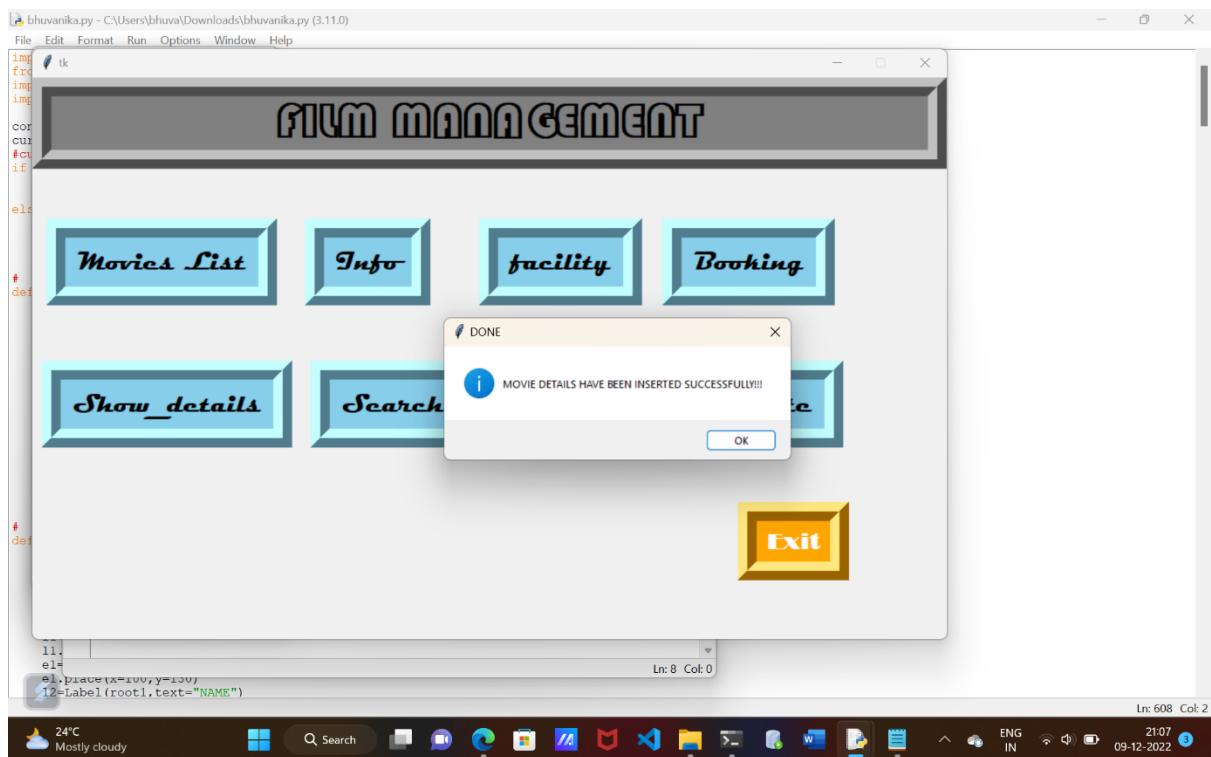
## 6.7 BOOKING FOR MOVIE



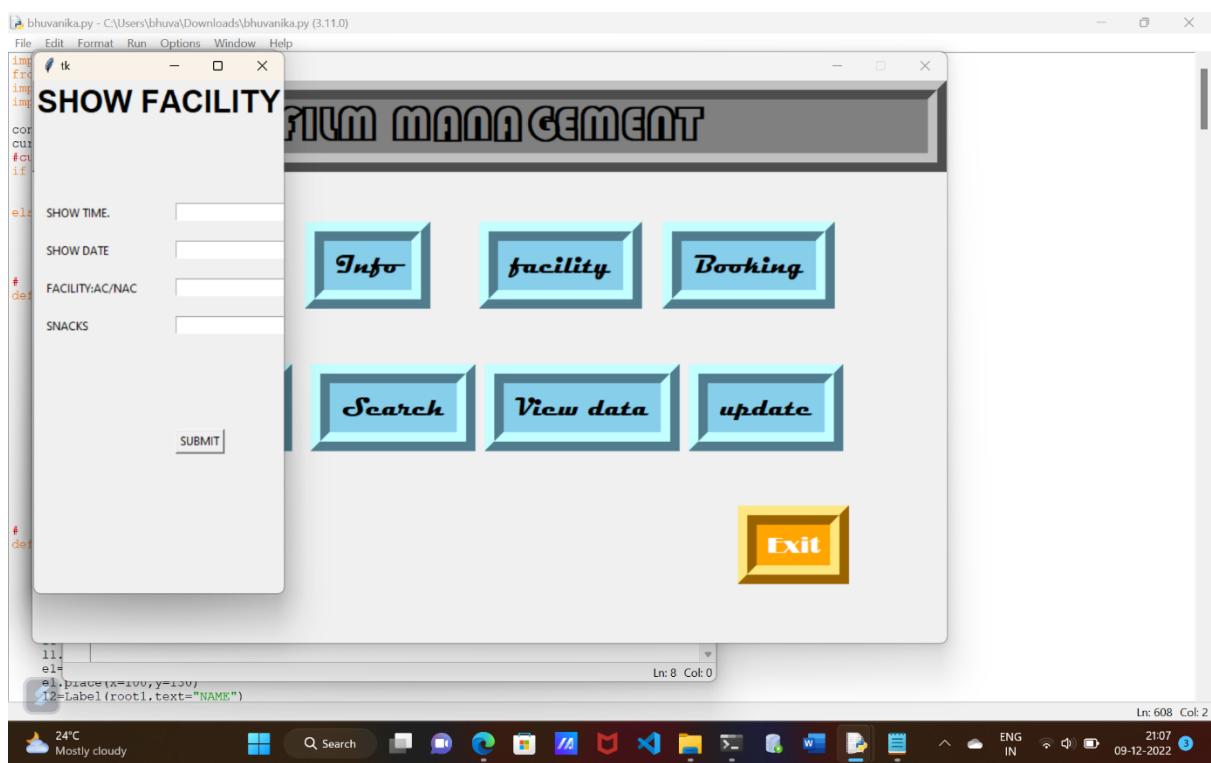
## 6.8 INSERTING VALUES FOR BOOKING



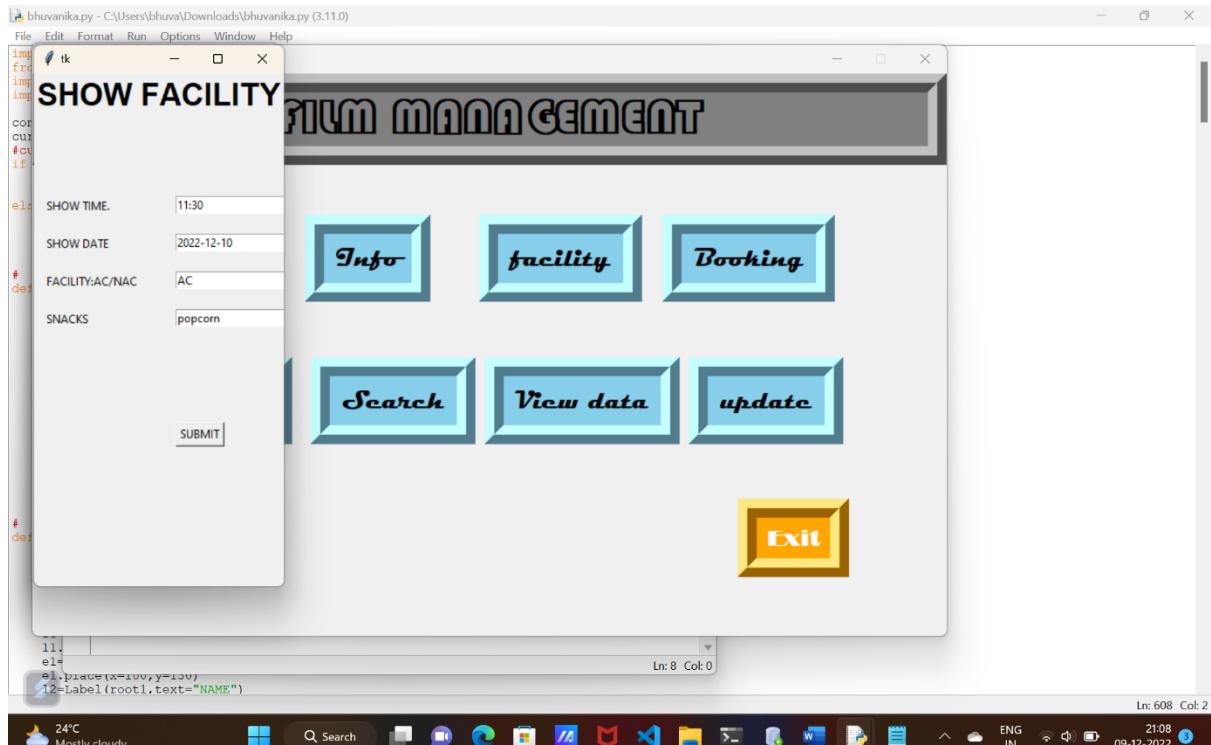
## 6.9 BOOKED FOR MOVIE



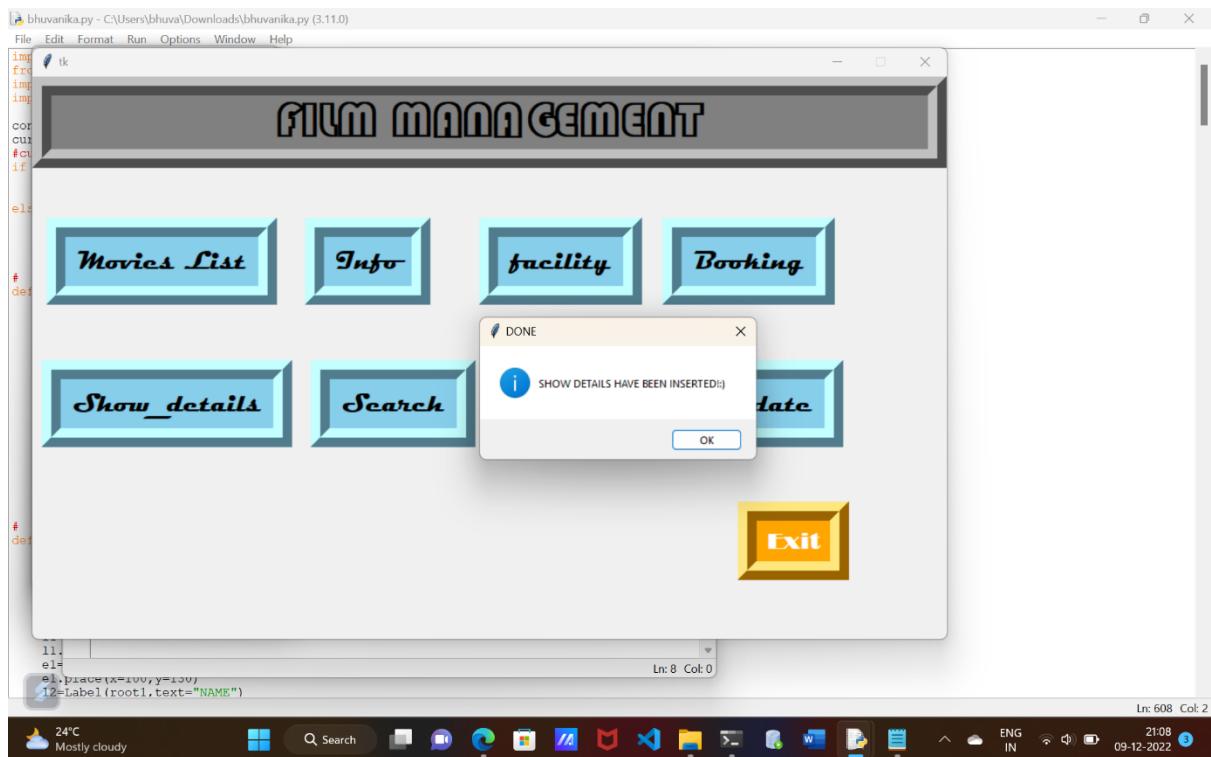
## 6.10 SHOW FACILITIES



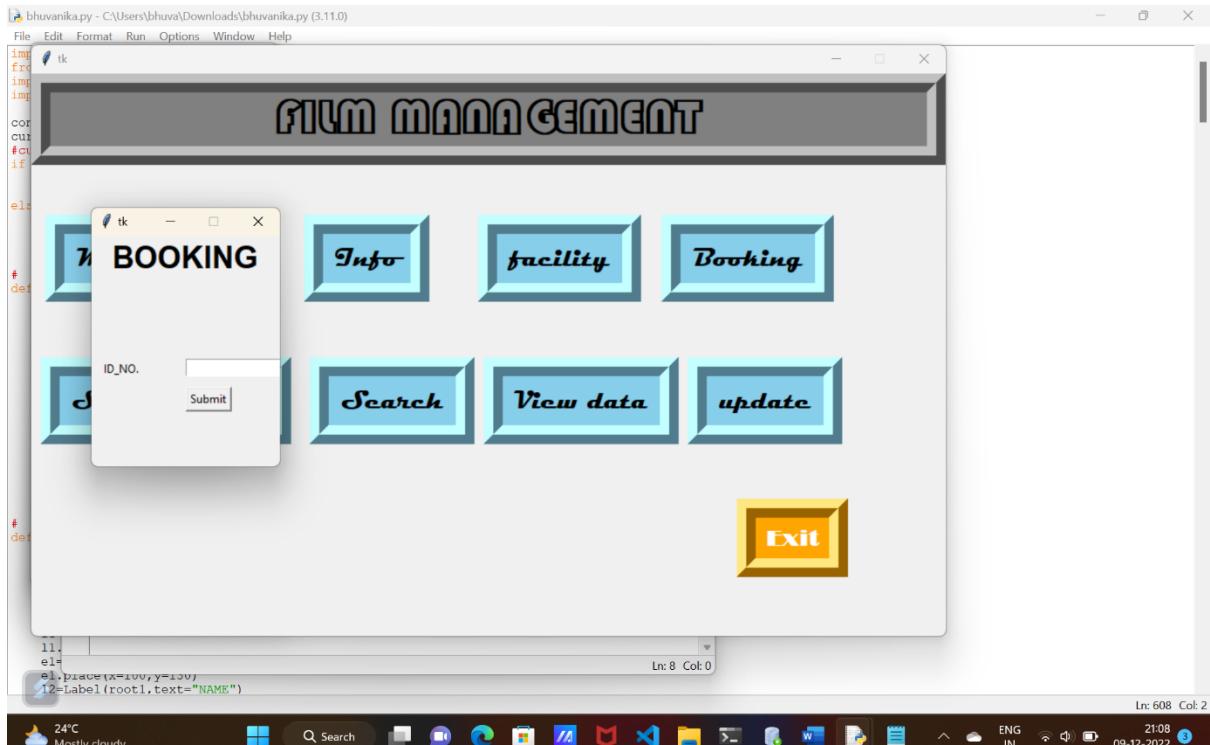
## 6.11 INSERTING SHOW DETAILS



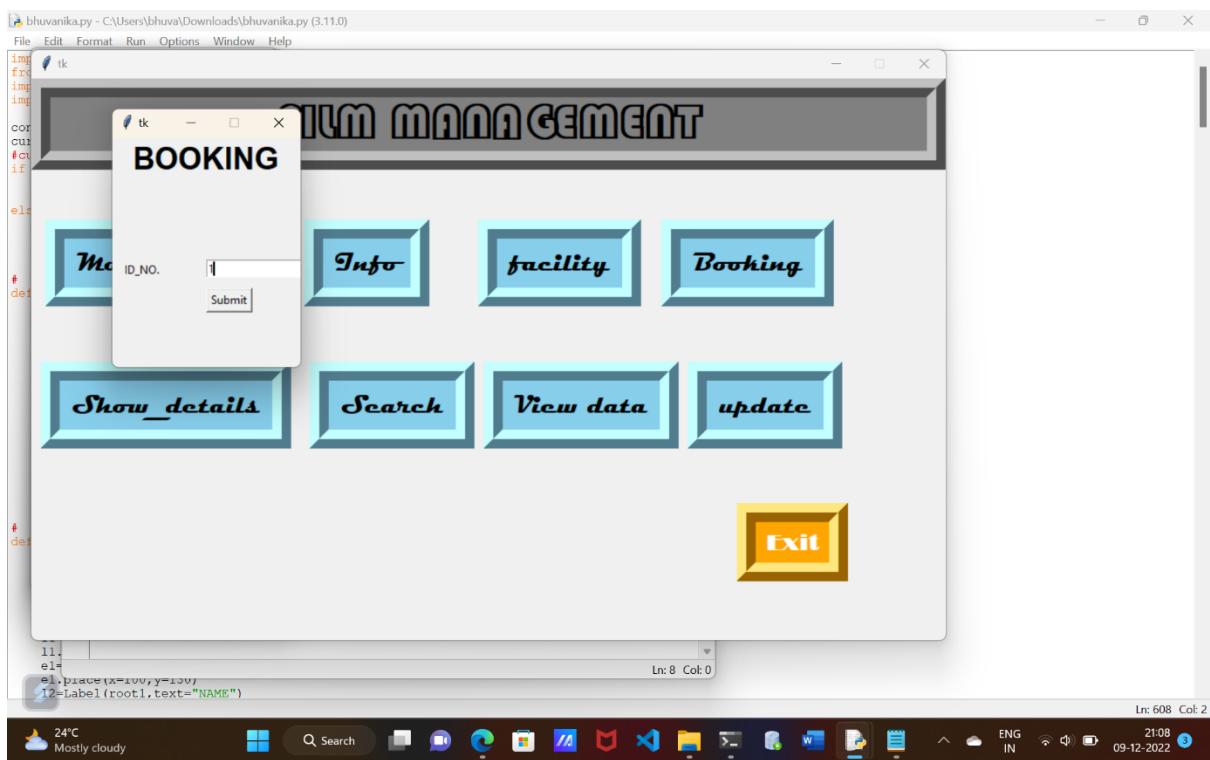
## **6.12 SHOW DETAILS INSERTED**



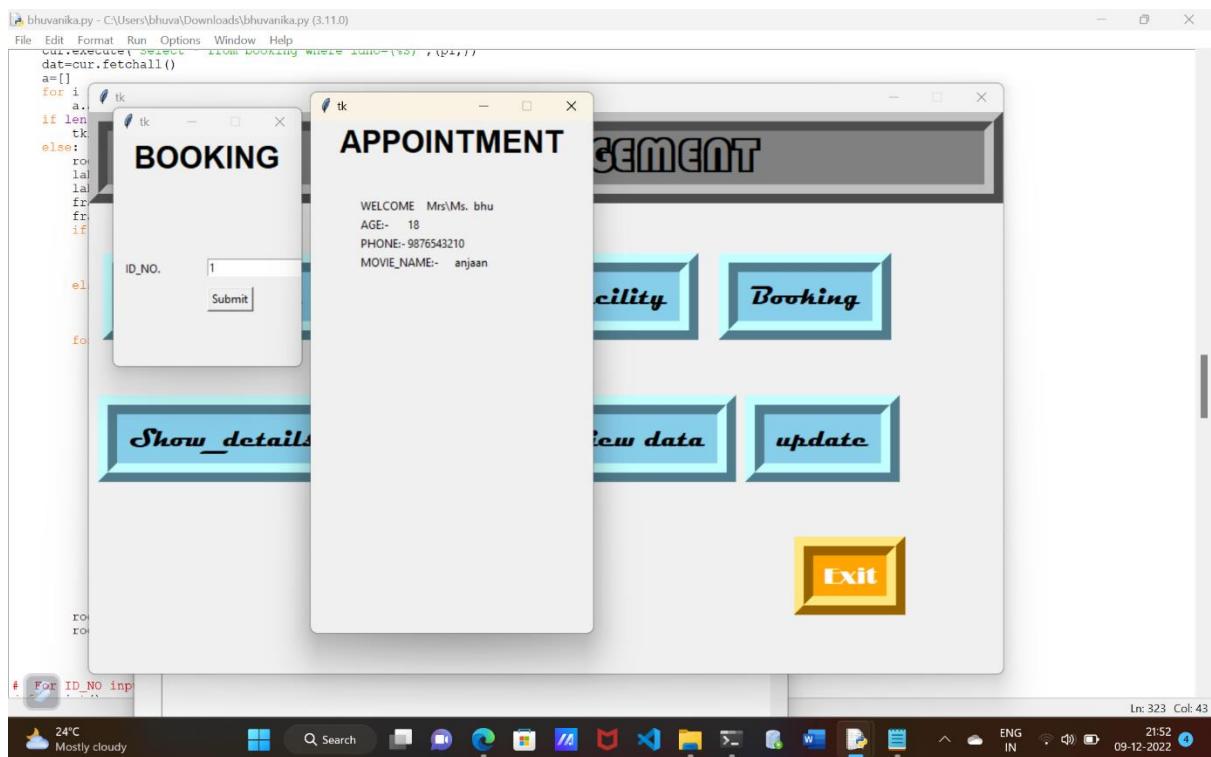
## 6.13 SEARCH



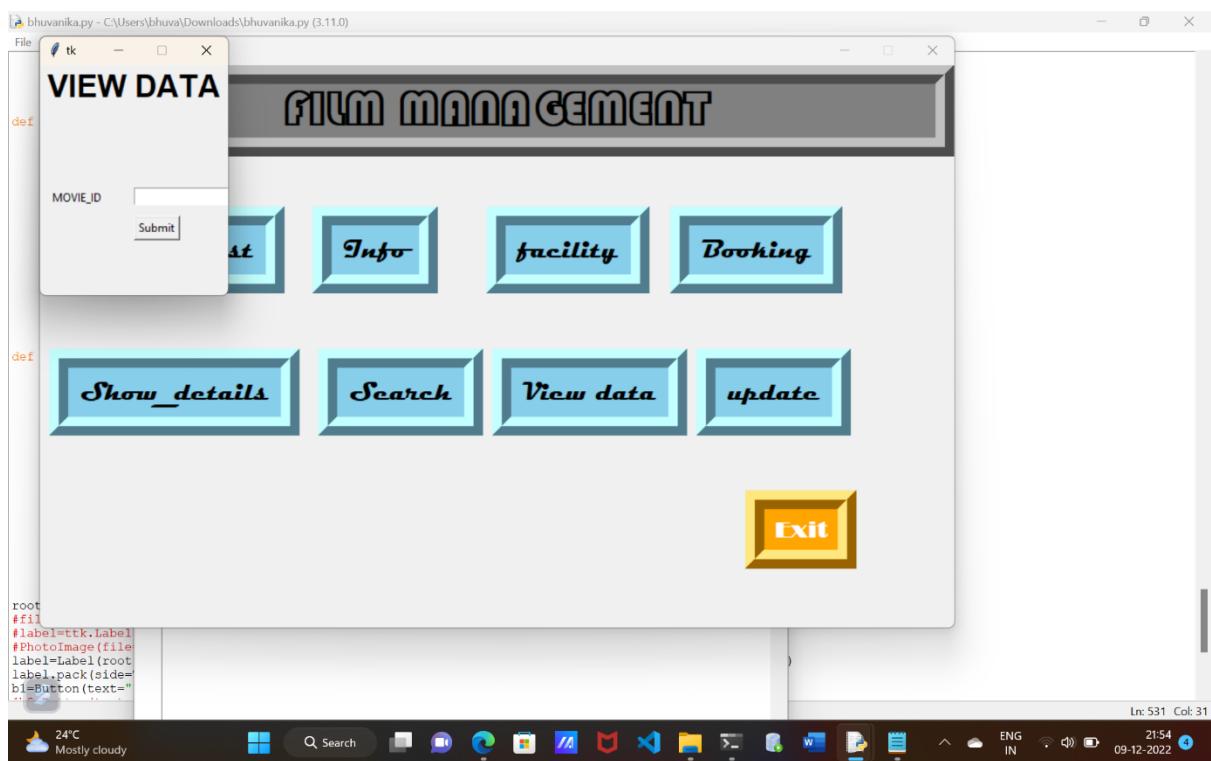
## 6.14 INSERTING VALUE TO SEARCH



## 6.15 AFTER SEARCH



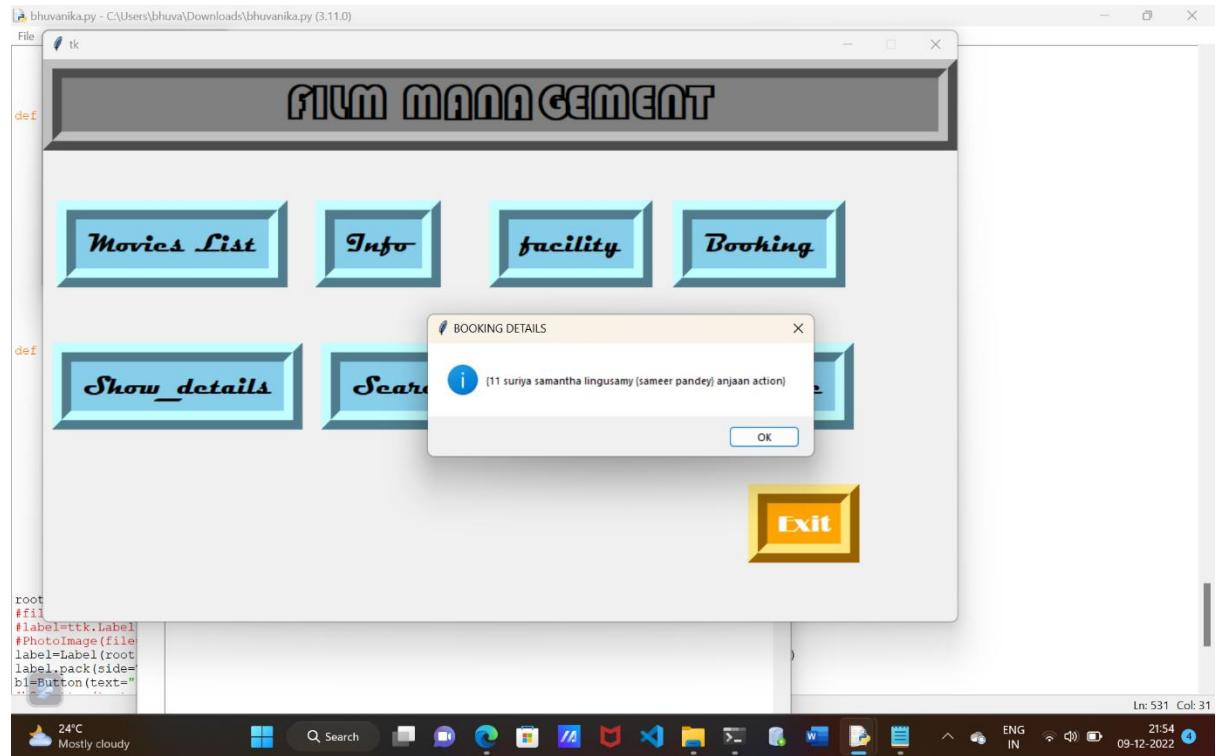
## 6.16 VIEW DATA



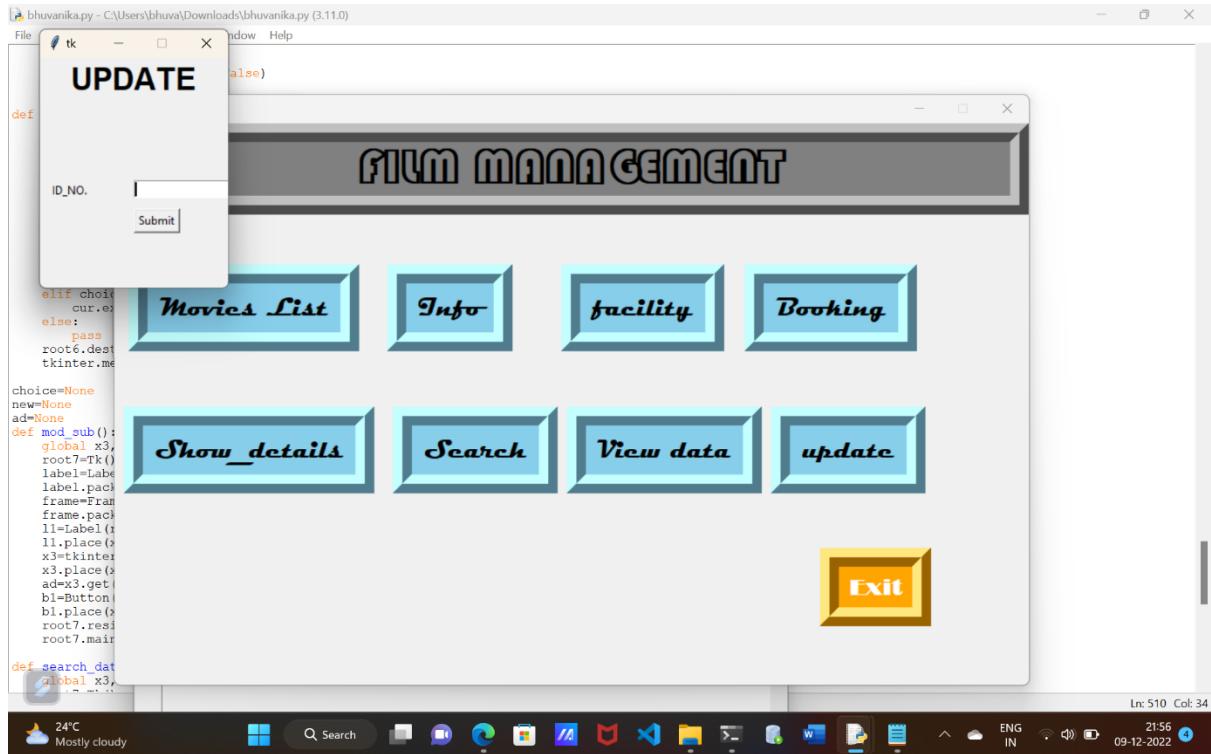
## 6.16 ENTER THE MOVIE\_ID TO VIEW THE DETAILS



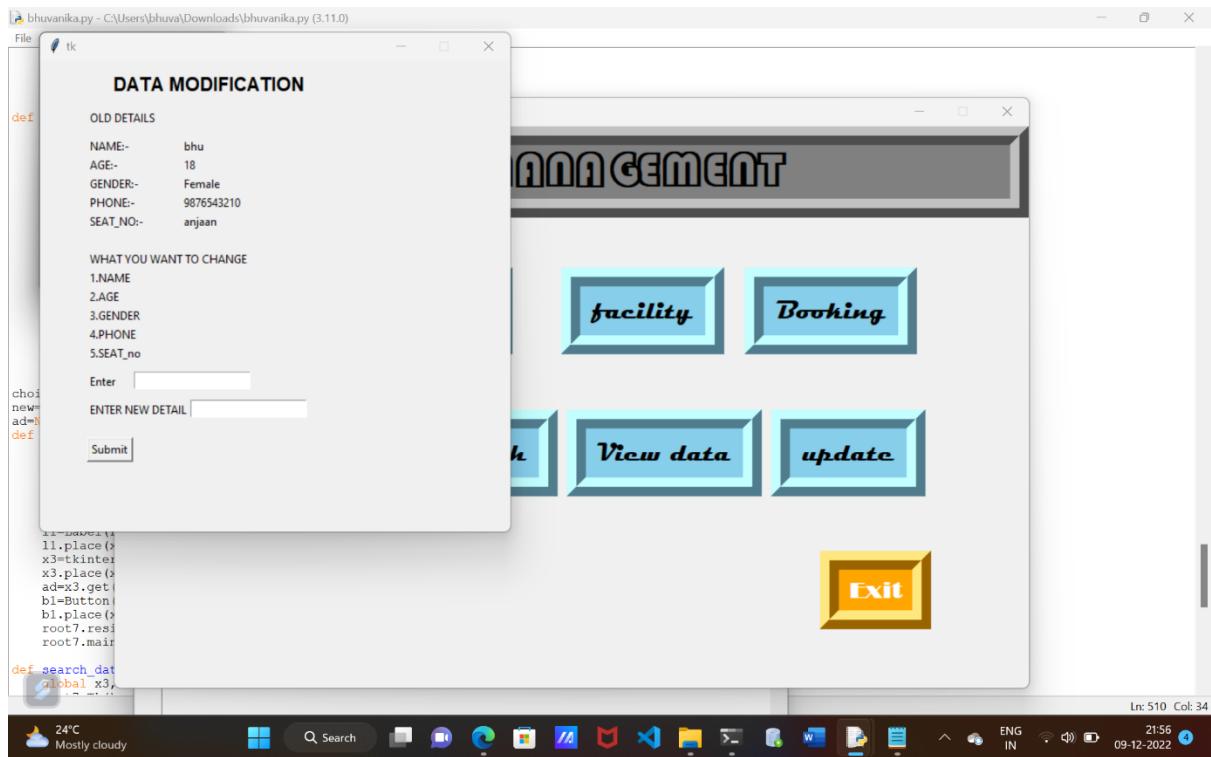
## 6.17 DETAILS



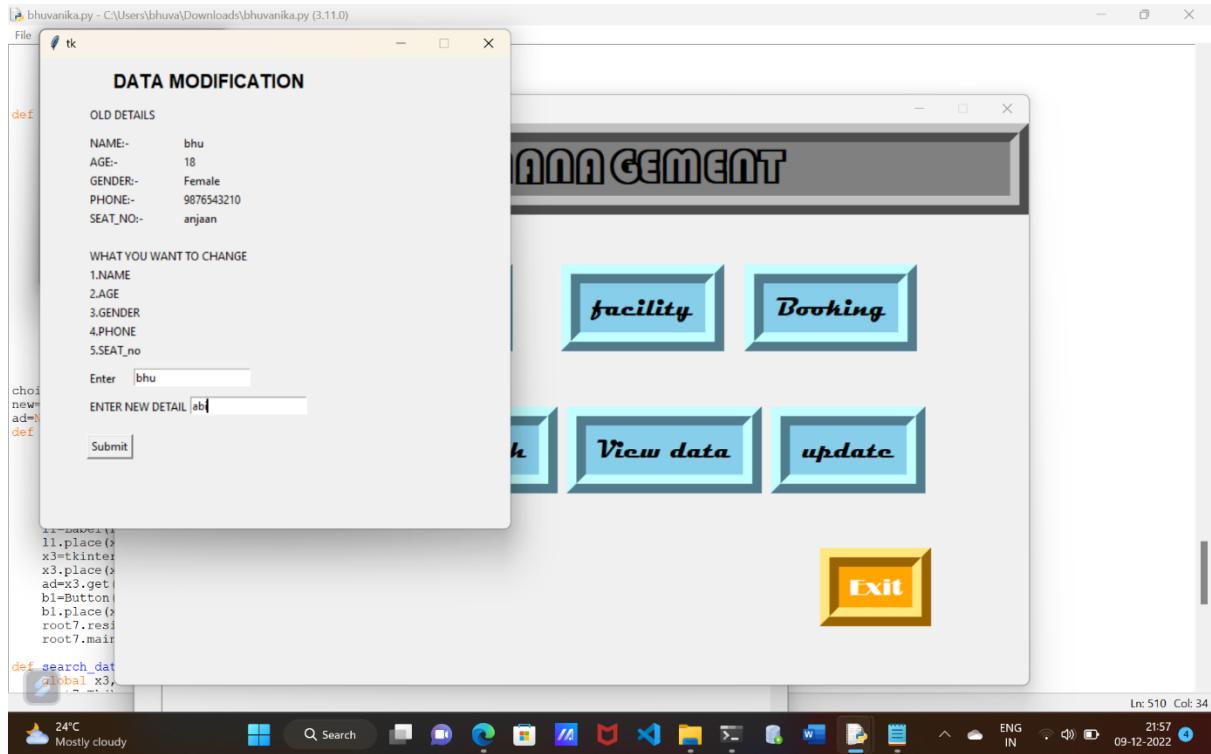
## 6.18 UPDATE



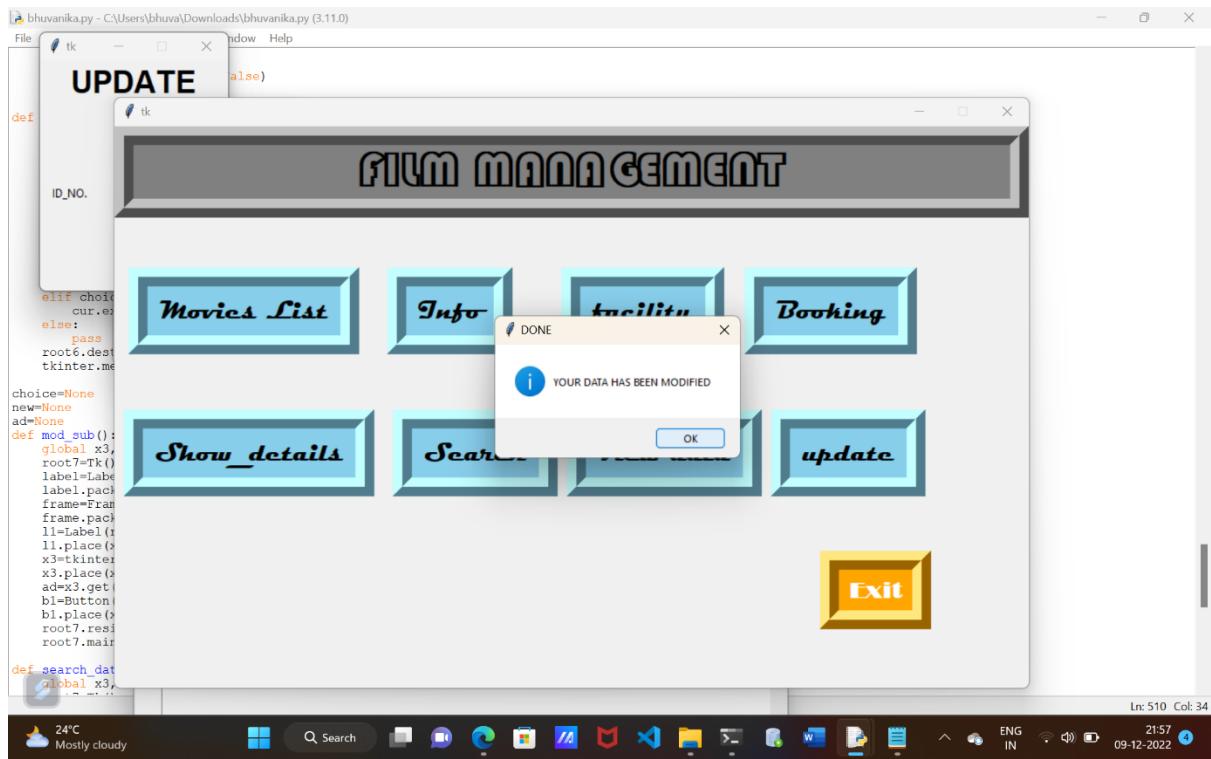
## 6.19 UPDATING FORM



## 6.20 ENTER THE VALUES TO UPDATE



## 6.21 MODIFIED



## **CHAPTER 7**

### **7.1 CONCLUSION**

The web-based application can help customer to book ticket for movie from any device which is connected to internet, the web-based application will make customers to watch trails, movie rating, and book tickets, pay from their debit cards easily

- Easily Manageable
- No More Repetition
- No need of Extra Work
- No more Manually Work
- Online Ticket Booking
- Security

The closing of our film review should remind the reader of your general thoughts and impressions of the film. You may also implicitly or explicitly state whether or not you recommend the film. Make sure to remind the reader of why the film is or is not worth seeing.

## **7.2 FUTURE ENHANCEMENTS**

### **User Features:**

1. Login
2. Register.
3. Search Movies.
4. Upcoming Movies.
5. Current Shows.
6. Book Movies.
7. Pay.

### **Admin Features:**

1. Add Movies.
2. Edit Movies.
3. Delete Movies
4. Booking History.
5. User details.
6. All Crud Operations.

## **APPENDIX**

### **SOURCE CODE**

```
import tkinter.messagebox
from tkinter import *
import pymysql as sqlcon
import random as rd

con=sqlcon.connect(host="localhost",user="root",password="bhuva2004",database="project")
">#connection to mysql
cur=con.cursor()
#cur = con.cursor(buffered=True)
if (con):
    # Carry out normal procedure
    print ("Connection successful")
else:
    print ("Connection unsuccessful")

# Message for registration
def entry():
```

```

global e1,e2,e3,e4,e5,e6
p1=e1.get()
p2=e2.get()
p3=e3.get()

p4=e4.get()
p5=e5.get()
p6=e6.get()

query="insert into booking values(\""+p1+"','"+p2+"','"+p3+"','"+p4+"','"+p5+"','"+p6+"\")"
con.commit()
cur.execute(query)
cur.execute("commit")

```

tkinter.messagebox.showinfo("DONE", "YOU HAVE BEEN BOOKED FOR MOVIE")

```

# For registration
def register():
    global e1,e2,e3,e4,e5,e6
    root1=Tk()
    label=Label(root1,text="BOOKING FORM",font='arial 25 bold')
    label.pack()
    frame=Frame(root1,height=500,width=200)
    frame.pack()
    l1=Label(root1,text="ID_NO.")
    l1.place(x=10,y=130)
    e1=tkinter.Entry(root1)
    e1.place(x=100,y=130)
    l2=Label(root1,text="NAME")
    l2.place(x=10,y=170)
    e2=tkinter.Entry(root1)
    e2.place(x=100,y=170)
    l3=Label(root1,text="AGE")
    l3.place(x=10,y=210)
    e3=tkinter.Entry(root1)
    e3.place(x=100,y=210)
    l4=Label(root1,text="GENDER M\F")
    l4.place(x=10,y=250)
    e4=tkinter.Entry(root1)
    e4.place(x=100,y=250)
    l5=Label(root1,text="PHONE")
    l5.place(x=10,y=290)
    e5=tkinter.Entry(root1)

```

```

e5.place(x=100,y=290)
l6=Label(root1,text="MOVIE_NAME")
l6.place(x=10,y=330)
e6=tkinter.Entry(root1)
e6.place(x=100,y=330)
b1=Button(root1,text="SUBMIT",command=entry)
b1.place(x=150,y=370)

root.resizable(False,False)
root1.mainloop()

# Message for registration
def entry1():
    global a1,a2,a3,a4
    p1=a1.get()
    p2=a2.get()
    p3=a3.get()
    p4=a4.get()

    query="insert into showdetails values(\""+p1+"','"+p2+"','"+p3+"','"+p4+"\")"
    con.commit()
    cur.execute(query)
    cur.execute("commit")

    tkinter.messagebox.showinfo("DONE", "SHOW DETAILS HAVE BEEN INSERTED!:")
")

# For theatre_details
def theatre_details():
    global a1,a2,a3,a4
    root1=Tk()
    label=Label(root1,text="SHOW FACILITY",font='arial 25 bold')
    label.pack()
    frame=Frame(root1,height=500,width=200)
    frame.pack()
    l1=Label(root1,text="SHOW TIME.")
    l1.place(x=10,y=130)
    a1=tkinter.Entry(root1)
    a1.place(x=150,y=130)
    l2=Label(root1,text="SHOW DATE")
    l2.place(x=10,y=170)
    a2=tkinter.Entry(root1)

```

```

a2.place(x=150,y=170)
l3=Label(root1,text="FACILITY:AC/NAC")
l3.place(x=10,y=210)
a3=tkinter.Entry(root1)
a3.place(x=150,y=210)
l4=Label(root1,text="SNACKS")
l4.place(x=10,y=250)
a4=tkinter.Entry(root1)
a4.place(x=150,y=250)

b1=Button(root1,text="SUBMIT",command=entry1)
b1.place(x=150,y=370)

```

```

root.resizable(False,False)
root1.mainloop()

```

```

# Message for registration
def entry2():
    global e1,e2,e3,e4,e5,e6,e7
    p1=e1.get()
    p2=e2.get()
    p3=e3.get()
    p4=e4.get()
    p5=e5.get()
    p6=e6.get()
    p7=e7.get()

```

```

query="insert into movie
values(""+p1+"','"+p2+"','"+p3+"','"+p4+"','"+p5+"','"+p6+"','"+p7+"')"
con.commit()
cur.execute(query)
cur.execute("commit")

```

```

tkinter.messagebox.showinfo("DONE", "MOVIE DETAILS HAVE BEEN INSERTED
SUCCESSFULLY!!!")

```

```

# For registration
def movie():
    global e1,e2,e3,e4,e5,e6,e7
    root1=Tk()

```

```

label=Label(root1,text="MOVIE DETAILS",font='arial 25 bold')
label.pack()
frame=Frame(root1,height=500,width=200)
frame.pack()
l1=Label(root1,text="MOVIE_ID")
l1.place(x=10,y=130)
e1=tkinter.Entry(root1)
e1.place(x=100,y=130)
l2=Label(root1,text="HERO")
l2.place(x=10,y=170)
e2=tkinter.Entry(root1)
e2.place(x=100,y=170)
l3=Label(root1,text="HEROINE")
l3.place(x=10,y=210)
e3=tkinter.Entry(root1)
e3.place(x=100,y=210)
l4=Label(root1,text="DIRECTOR")
l4.place(x=10,y=250)
e4=tkinter.Entry(root1)
e4.place(x=100,y=250)
l5=Label(root1,text="SINGER")
l5.place(x=10,y=290)
e5=tkinter.Entry(root1)
e5.place(x=100,y=290)
l6=Label(root1,text="MOVIE_NAME")
l6.place(x=10,y=330)
e6=tkinter.Entry(root1)
e6.place(x=100,y=330)
l7=Label(root1,text="GENRE")
l7.place(x=10,y=370)
e7=tkinter.Entry(root1)
e7.place(x=100,y=370)
b1=Button(root1,text="SUBMIT",command=entry2)
b1.place(x=150,y=450)

```

```

root.resizable(False,False)
root1.mainloop()

```

```

# Message for appointment
def apo_details():
    global x1,x2,h,p1,p2,p3,o,x4,x3
    p1=x2.get()
    p2=x3.get()

```

```

p3=x4.get()
if int(p1)==1:
    i=("ANJAAN \nRoom no:- 10")
    j=("MASTER \nRoom no:- 11")
    q=(i,j)
    h=rd.choice(q)
    u=(23,34,12,67,53,72)
    o=rd.choice(u)
    det=("Your movie booking details",h,
          "\nDate:-",p2,
          "\nShow_Time:-",p3,
          '\nFacility:-',o)

    query='insert into booking_details values("{}", "{}", "{}", "{}",
    "{}").format(p1,h,p2,p3,o)
    cur.execute(query)
    tkinter.messagebox.showinfo("BOOKING DETAILS",det)

elif int(p1)==2:
    i=("VISWASAM \nRoom no. 16")
    j=("PETTA \nRoom no. 17")
    q=(i,j)
    h=rd.choice(q)
    u=(23,34,12,67,53,72)
    o=rd.choice(u)
    det=("Your movie booking details",h,
          "\nDate:-",p2,
          "\nShow_Time:-",p3,
          '\nFacility:-',o)

    query='insert into booking_details values("{}", "{}", "{}", "{}",
    "{}").format(p1,h,p2,p3,o)
    cur.execute(query)
    tkinter.messagebox.showinfo("BOOKING DETAILS",det)

elif int(p1)==3:
    i=("DON \nRoom no. 12")
    j=("96 \nRoom no. 13")
    q=(i,j)
    h=rd.choice(q)
    u=(23,34,12,67,53,72)
    o=rd.choice(u)
    det=("Your movie booking details",h,
          "\nDate:-",p2,
          "\nShow_Time:-",p3,

```

```

'\nFacility:-',o)
query='insert into booking_details values("{}", "{}", "{}", "{}",
"{}")'.format(p1,h,p2,p3,o)
cur.execute(query)
tkinter.messagebox.showinfo("BOOKING DETAILS",det)

elif int(p1)==4:
    i=("NGK, \nRoom no. 18")
    j=(" MAYA\nRoom no. 19")
    q=(i,j)
    h=rd.choice(q)
    u=(23,34,12,67,53,72)
    o=rd.choice(u)
    det=("Your movie booking details",h,
          "\nDate:-",p2,
          "\nShow_Time:-",p3,
          '\nFacility:-',o)
    query='insert into booking_details values("{}", "{}", "{}", "{}",
"{}")'.format(p1,h,p2,p3,o)
    cur.execute(query)
    tkinter.messagebox.showinfo("BOOKING DETAILS",det)

elif int(p1)==5:
    i=("LOVE TODAY \nRoom no. 14")
    j=("VIKRAM VEDHA \nRoom no. 15")
    q=(i,j)
    h=rd.choice(q)
    u=(23,34,12,67,53,72)
    o=rd.choice(u)
    det=("Your movie booking details",h,
          "\nDate:-",p2,
          "\nShow_Time:-",p3,
          '\nFacility:-',o)
    query='insert into booking_details values("{}", "{}", "{}", "{}",
"{}")'.format(p1,h,p2,p3,o)
    cur.execute(query)
    tkinter.messagebox.showinfo("BOOKING DETAILS",det)

elif int(p1)==6:
    i=("SARKAR \nRoom no. 001")
    j=("D BLOCK \nRoom no. 002")
    k=("ET \nRoom no. 003")
    l=("THUPPARIVALAN \nRoom no. 004")
    q=(i,j,k,l)
    h=rd.choice(q)
    u=(23,34,12,67,53,72)

```

```

o=rd.choice(u)
det=("Your movie booking details",h,
      "\nDate:-",p2,
      "\nShow_Time:-",p3,
      '\nFacility:-',o)
query='insert into booking_details values("{}", "{}", "{}", "{}",
"{}")'.format(p1,h,p2,p3,o)
cur.execute(query)
tkinter.messagebox.showinfo("BOOKING DETAILS",det)
else:
    tkinter.messagebox.showwarning('WRONG INPUT','PLEASE ENTER VALID
VALUE')

# For appointment
def get_apoint():
    global x1,x2,x3,x4
    p1=x1.get()
    cur.execute('select * from booking where idno=(%s)',(p1,))
    dat=cur.fetchall()
    a=[]
    for i in dat:
        a.append(i)
    if len(a)==0:
        tkinter.messagebox.showwarning("ERROR", "NO DATA FOUND!!")
    else:
        root3=Tk()
        label=Label(root3,text="APPOINTMENT",font='arial 25 bold')
        label.pack()
        frame=Frame(root3,height=500,width=300)
        frame.pack()
        if i[3]=='M' or i[3]=='m':
            x="Mr."
            name2=Label(root3,text=i[1])
            name2.place(x=140,y=80)
        else:
            x="Mrs\Ms."
            name2=Label(root3,text=i[1])
            name2.place(x=170,y=80)
        for i in dat:
            name=Label(root3,text='WELCOME')
            name.place(x=50,y=80)
            name1=Label(root3,text=x)
            name1.place(x=120,y=80)
            age=Label(root3,text='AGE:-')

```

```

age.place(x=50,y=100)
age1=Label(root3,text=i[2])
age1.place(x=100,y=100)
phone=Label(root3,text='PHONE:-')
phone.place(x=50,y=120)
phone1=Label(root3,text=i[4])
phone1.place(x=100,y=120)
bg=Label(root3,text='MOVIE_NAME:-')
bg.place(x=50,y=140)
bg1=Label(root3,text=i[5])
bg1.place(x=150,y=140)

```

```

root3.resizable(False,False)
root3.mainloop()

```

```

# For ID_NO input
def apoint():
    global x1
    root2=Tk()
    label=Label(root2,text="BOOKING",font='arial 25 bold')
    label.pack()
    frame=Frame(root2,height=200,width=200)
    frame.pack()
    l1=Label(root2,text="ID_NO.")
    l1.place(x=10,y=130)
    x1=tkinter.Entry(root2)
    x1.place(x=100,y=130)
    b1=Button(root2,text='Submit',command=get_apoint)
    b1.place(x=100,y=160)
    root2.resizable(False,False)
    root2.mainloop()

```

```

# List of movies
def lst_doc():
    root4=Tk()

```

```

l=["ANJAAN","MASTER","VISWASAM","PETTA","DON","96","NGK","MAYA","LOVE TODAY","VIKRAM VEDHA","SARKAR","D BLOCK","ET","THUPPARIVAALAN"]

```

```

m=["Action","Action","Action","Action","Comedy","Love","Politics","Horror","Love","Acti
on","Politics","Horror","Crime","Crime"]
n=[10,11,12,13,14,15,16,17,18,19,20,21,22,23]

frame=Frame(root4,height=500,width=500)
frame.pack()

l1=Label(root4,text='NAME OF MOVIE')
l1.place(x=20,y=10)
count=20
for i in l:
    count=count+20
    l=Label(root4,text=i)
    l.place(x=20,y=count)

l2=Label(root4,text='GENRE')
l2.place(x=140,y=10)
count1=20
for i in m:
    count1=count1+20
    l3=Label(root4,text=i)
    l3.place(x=140,y=count1)

l4=Label(root4,text='ROOM NO')
l4.place(x=260,y=10)
count2=20
for i in n:
    count2=count2+20
    l5=Label(root4,text=i)
    l5.place(x=260,y=count2)
root.resizable(False,False)
root4.mainloop()

def ser_avail():

    root5=Tk()
    frame=Frame(root5,height=500,width=500)
    frame.pack()
    l1=Label(root5,text='SERVICES AVAILABLE')
    l1.place(x=20,y=10)
    f=["AC","NON-AC","NON-AC","AC","NON-AC + HD DISPLAY","AC +
ADJUSTABLE SEATS","AC + FREE SNACKS","AC + HD DISPLAY","NON-AC +

```

```

FREE SNACKS","NON-AC + ADJUSTABLE SEATS","NON-AC + HD DISPLAY","AC +
ADJUSTABLE SEATS","AC + FREE SNACKS","AC + HD DISPLAY"]
count1=20
for i in f:
    count1=count1+20
    l3=Label(root5,text=i)
    l3.place(x=20,y=count1)
l2=Label(root5,text='ROOM NO.')
l2.place(x=180,y=10)
g=[10,11,12,13,14,15,16,17,18,19,20,21,22,23]
count2=20
for i in g:
    count2=count2+20
    l4=Label(root5,text=i)
    l4.place(x=180,y=count2)
l5=Label(root5,text='To avail any of these please contact on our no.: - 7042****55')
l5.place(x=20,y=350)
root5.resizable(False,False)
root5.mainloop()

def modify():
    global x3,x4,choice,new,x5,root6
    p1=x3.get()
    cur.execute('select * from booking where idno=(%s)',(p1,))

    dat=cur.fetchall()
    a=[]
    for i in dat:
        a.append(i)
    if len(a)==0:
        tkinter.messagebox.showwarning("ERROR", "NO DATA FOUND!!")
    else:
        root6=Tk()
        frame=Frame(root6,height=500,width=500)
        frame.pack()
        l1=Label(root6,text='DATA MODIFICATION',font="arial 15 bold")
        l1.place(x=75,y=10)
        l2=Label(root6,text='WHAT YOU WANT TO CHANGE')
        l2.place(x=50,y=200)
        l3=Label(root6,text='1.NAME')
        l3.place(x=50,y=220)
        l4=Label(root6,text='2.AGE')
        l4.place(x=50,y=240)
        l5=Label(root6,text='3.GENDER')

```

```

15.place(x=50,y=260)
16=Label(root6,text='4.PHONE')
16.place(x=50,y=280)
17=Label(root6,text='5.SEAT_no')
17.place(x=50,y=300)
x2=Label(root6,text='Enter')
x2.place(x=50,y=330)
x4=tkinter.Entry(root6)
choice=x4.get()
x4.place(x=100,y=330)
for i in dat:
    name=Label(root6,text='NAME:-')
    name.place(x=50,y=80)
    name1=Label(root6,text=i[1])
    name1.place(x=150,y=80)
    age=Label(root6,text='AGE:-')
    age.place(x=50,y=100)
    age1=Label(root6,text=i[2])
    age1.place(x=150,y=100)
    gen=Label(root6,text='GENDER:-')
    gen.place(x=50,y=120)
    gen1=Label(root6,text=i[3])
    gen1.place(x=150,y=120)
    pho=Label(root6,text='PHONE:-')
    pho.place(x=50,y=140)
    pho1=Label(root6,text=i[4])
    pho1.place(x=150,y=140)
    bg=Label(root6,text='SEAT_NO:-')
    bg.place(x=50,y=160)
    bg1=Label(root6,text=i[5])
    bg1.place(x=150,y=160)
b=Button(root6,text='Submit',command=do_modify)
b.place(x=50,y=400)
L1=Label(root6,text='OLD DETAILS')
L1.place(x=50,y=50)
L2=Label(root6,text='ENTER NEW DETAIL')
L2.place(x=50,y=360)
x5=tkinter.Entry(root6)
new=x5.get()
x5.place(x=160,y=360)

root6.resizable(False,False)
root6.mainloop()

```

```

def do_modify():
    global ad,x3,x4,x5
    ad=x3.get()
    choice=x4.get()
    new=x5.get()
    if choice=='1':
        cur.execute('update booking set name=%s where idno=%s',(x4,x5))
    elif choice=='2':
        cur.execute('update booking set age={ } where idno={ }'.format(new,ad))
    elif choice=='3':
        cur.execute('update booking set gender={ } where idno={ }'.format(new,ad))
    elif choice=='4':
        cur.execute('update booking set phone={ } where idno={ }'.format(new,ad))
    elif choice=='5':
        cur.execute('update booking set movie_name={ } where idno={ }'.format(new,ad))
    else:
        pass
    root6.destroy()
    tkinter.messagebox.showinfo("DONE", "YOUR DATA HAS BEEN MODIFIED")

choice=None
new=None
ad=None
def mod_sub():
    global x3,ad
    root7=Tk()
    label=Label(root7,text="UPDATE",font='arial 25 bold')
    label.pack()
    frame=Frame(root7,height=200,width=200)
    frame.pack()
    l1=Label(root7,text="ID_NO.")
    l1.place(x=10,y=130)
    x3=tkinter.Entry(root7)
    x3.place(x=100,y=130)
    ad=x3.get()
    b1=Button(root7,text='Submit',command=modify)
    b1.place(x=100,y=160)
    root7.resizable(False,False)
    root7.mainloop()

def search_data():
    global x3,ad
    root7=Tk()
    label=Label(root7,text="VIEW DATA",font='arial 25 bold')

```

```

label.pack()
frame=Frame(root7,height=200,width=200)
frame.pack()
l1=Label(root7,text="MOVIE_ID")
l1.place(x=10,y=130)
x3=tkinter.Entry(root7)
x3.place(x=100,y=130)
ad=x3.get()
b1=Button(root7,text='Submit',command=view_data)
b1.place(x=100,y=160)
root7.resizable(False,False)
root7.mainloop()

def view_data():
    global p1,p2
    p1=x3.get()
    p2=x3.get()
    cur.execute('select * from booking where idno=(%s',(p1,))
    cur.execute('select * from movie where movie_id=(%s',(p1,))
    dat=cur.fetchall()
    print(dat)
    a=[]
    for i in dat:
        a.append(i)
    if len(a)==0:
        tkinter.messagebox.showwarning("ERROR", "NO DATA FOUND!!")
    else:
        det=a
        tkinter.messagebox.showinfo("BOOKING DETAILS",det)

```

```

root=Tk()
#film=PhotoImage(file='C:\\Users\\bhuva\\Downloads')
#label=ttk.Label(root,image=OIP)
#PhotoImage(file='C:\\Users\\bhuva\\bhu\\Downloads')
label=Label(root,text="FILM MANAGEMENT",bd=20,relief=RIDGE,font="STCaiyun 40 bold",bg='grey',fg='black')
label.pack(side=TOP,fill=X)
b1=Button(text="Info",bd=20,relief=RIDGE,font="Magneto 20 bold",bg='sky blue',command=register)
#b2=Button(text="Search",bd=20,relief=RIDGE,font="Magneto 20 bold",bg='sky blue',command=apoint)
b3=Button(text="Movies List",bd=20,relief=RIDGE,font="Magneto 20 bold",bg='sky blue',command=lst_doc)

```

```

b4=Button(text="facility",bd=20,relief=RIDGE,font='Magneto 20 bold',bg='sky
blue',command=ser_avail)
b7=Button(text="View data",bd=20,relief=RIDGE,font='Magneto 20 bold',bg='sky
blue',command=search_data)
b5=Button(text="update",bd=20,relief=RIDGE,font='Magneto 20 bold',bg='sky
blue',command=mod_sub)
b6=Button(text="Exit",bd=20,relief=RIDGE,font='Broadway 18
bold',command=root.destroy,bg='orange',fg='white')
b8=Button(text="Show _details",bd=20,relief=RIDGE,font="Magneto 20 bold",bg='sky
blue',command=theatre_details)
b9=Button(text="Booking",bd=20,relief=RIDGE,font="Magneto 20 bold",bg='sky
blue',command=movie)
b2=Button(text="Search",bd=20,relief=RIDGE,font="Magneto 20 bold",bg='sky
blue',command=apoint)
label.pack()
#b1.pack(side=LEFT,pady=2)
#b3.pack(side=LEFT,padx=2)
b8.pack(side=LEFT,padx=10)
b9.place(x=670,y=150)
b2.pack(side=LEFT,padx=10)
b4.place(x=475,y=150)
b6.place(x=750,y=451)
#b8.place(x=290,y=150)
#b9.place(x=15,y=150)
#b2.place(x=17,y=451)
b7.pack(side=LEFT,pady=10)
b3.place(x=15,y=150)
b1.place(x=290,y=150)
b5.pack(side=LEFT,padx=10)
frame=Frame(root,height=500,width=100)
frame.pack()
root.resizable(False,False)
root.mainloop()

```

## **REFERENCES**

### **BOOKMARKS**

1. Programming in Idle  
-Guido Van Rossum
2. Create and insert in MySQL  
-Michael Widenius

### **WEBSITES**

Lib/idlelib/