ACKNOWLEDGEMENT

I sincerely thank the Institution for providing all facilities in completing the project. I am indebted to the Management and Principal **Prof. Veena T** for supporting us in all our endeavours in completion of the project.

I am grateful to HOD and project guide **Mrs. Latha H R** for constant support and encouragement.

Finally, I take this opportunity to thank all teaching and non-teaching staff for their immense Cooperation during the course.

AMRUTHA C
REG NO: 202VSB7002

MEGHANA M
REG NO: 202VSB7031

DECLARATION

I hereby declare that the project entitled “**HOTEL BOOKING SYSTEM”** is a bonafied work carried out under the guidance of **Mrs. Latha H R.** The project depicted in the report is my genuine work and has not been presented in any other university for award. This project is submitted as partial fulfillment for the **Bachelor of Computer Applications (BCA)** course of Bangalore University.

I further declare that this project work is my original and has not been submitted to any other university or institution for the award of degree.

ABSTRACT

The project **"Hotel Booking System"** is a system based on accessing the internet to book for rooms in a hotel. The purpose of this study is to develop and implement an online hotel reservation system for hotels that will replace the manual method of booking for hotel rooms. The previous system for booking rooms were faced with so many problems like, delay in processing the customer booking or paying for rooms that is below or beyond his standard, causes difficulty for emergency booking.

The objects-oriented analysis and design methodology (OOADM) was therefore used to analyze the system in order to discover the various objects involved and how they interact with one another so that a new and improved system can be defined.

The use of online view of room rates and uploading of available rooms and facilities was used for the new system so that the customer can view and make his choice before arrival, and also in the case of emergency travelling. This new system assisted the hotel owners in managing their hotels, because they can also regulate the receptionist moves and avoid fraudulent activities. It also increased the efficiency of the hotel managers and also their profit margin, once they have better and good facilities.

TABLE OF CONTENTS

CHAPTER-1

INTRODUCTION

CHAPTER-2

LITERATURE SURVEY

CHAPTER-6

RESULTS

CHAPTER-7

VALIDATION

CHAPTER-8

CONCLUSION

CHAPTER-9

FUTURE ENHANCEMENT

CHAPTER-10

BIBLIOGRAPHY

INTRODUCTION

1.1 INTRODUCTION TO HOTEL BOOKING SYSTEM

A Hotel booking system is an online booking engine that allows guests to make secure online reservations through hotel website and helps hotels to accept bookings and collect payments online.

Hotel booking system is complete hotel quotation booking system that comes with the key role of Hotel channel manager, Hotel extranet and own contracting to help hotels to automate day-to-day hotel operations and to increase bookings.

Hotel booking system is complete hotel reservation system which is easily integrated in hotel website allowing guests to make online reservations through hotel website and provide hoteliers with hotel channel management and other resources to improve customer experience and increase online bookings.

It allows hoteliers to provide their customers with best options for hotel rooms, amenities coupled with online booking and payment option to improve guest experience from booking to post-stay.

1.2 PROJECT DEFINITION

The current manual reservation system uses paper and direct human interaction to book the hotel room and manage reservations. This makes delays exchanging of information in the hotel.

As Hotels booking is having manual booking system, they are facing some problems issuing booking requests of customers. All the necessary booking stuffs are being done in hard copy. So it become much difficult for staffs to keep the records updated all the time. As for example, if the customers need to change the check in date it become difficult for them to find out the customers booking details for updating as there are so many customers booking records.

Again, regarding current system customers cannot give feedback online and also staff cannot reply to them promptly.

After analysed various traditional hotel room booking system, we had noticed the below problems in their system.

- Manual system for storing records is not consistent as some inaccuracy can creep in while writing records manually.
- Guest or visitors may face hard time to getting place to stay in area.
- It is difficult to store record of availability of room and huge number of customer's records.
- More manpower is required and the current system consumes too much time
- to making reservation and storing data.
- Maintaining of up to date availability of inventory is bit difficult, thus, it is making hard for executive to search the particular records and room.
- Payment processing and collection is difficult.
- Inaccurate of records or data.
- There is no centralized database can be created as information not at one place.
- More money and paper and other resources are wasted to store the record of available rooms and customers.

1.3 OBJECTIVES

A flexible web-based learning experience allows you to go through a guided curriculum or choose lessons on an as-needed basis. Following are the main objectives: The main purpose of this work is therefore to develop a web application program that would circumvent all those problems encountered in the manual hotel booking system, so that customers can easily go online with their mobile phones, tablets or laptops in order to browse the relevant information they need

About the hotels within their locality so that they can book for the appropriate suite that is within their budget.

• Customers can easily go online with their mobile phones or laptops to browse about hotels within their vicinity,

• Guests can be able to book for rooms within their budgets after seeing and accessing these hotels within their vicinity.

• Receptionists can easily access customer's information online without delay or with little delay.

• Fraud done by the receptionist by not registering every customer will not occur because the Customers details will be online and can be accessed by the admin too.

- To enable online booking via the internet.
- To enable automated data entry methods.
- Ensure efficient and reliable communication within the hotel.
- Avoid data entry errors by use of input masks.
- Enable easy authorized modification of data.
- Enforce security measures to avoid unauthorized access to guest records.
- Enable fast and easy retrieval of guest records and data for fast reference activities.

1.4 SCOPE

This study is aimed at finding out how effective the online room reservation or booking system will improve the operations of room reservations in hotels. However, out of the several departments that makes up the hotel, this research project is restricted to only one section (room reservation) section.

- Guests, Receptionist, Managers and hotel owners would benefit immensely when the new system is implemented, in the sense that:
- Guests would experience faster access time with the system. Front desk office attendants (receptionists) would also experience minimal efforts in attending to guests, as they would only need to access the guests' details through internet.
- The managers can easily carryout an inventory to know the total number of guests that checked in or out of the rooms, and those that booked in advance.
- Increased efficiency in the operations of the business would definitely bring about increased profitability.

LITERATURE SURVEY

2.1 LITERATURE REVIEW

Computer-based hotel booking system defines as structured systems that rely on computer hardware and software technology to collect, process, and store and distribute information. Information systems are employed to support decision making and control in an organization. Information systems can also be used to analyze problems, visualize complex subjects, and create new products. Input, processing and output are the three activities in an information system that produces the information an organization needs. Advantages of Computer-Based Information Systems Compute-based information systems have been in widespread use since the 1990s in industry These systems provide fast, centralized access to databases of personnel information, reference reading, best practices and on-the-job training, and are easily customizable to meet an organization's needs. With the Internet and technology use of computer-based information networks is growing faster each year.

2.2 EXISTING SYSTEM

The problem arises in existing system are:

- There is maximum possibility of losing customers record and files or mix up customer's record.
- As we are not sure and we can to re-use past records, unwanted duplication of information and data is possible.
- As the actual file is available, there are chances that one can use the files and information whiteout permission or right.
- Maintaining file security and standard is difficult
- As very easy access to guest information by unintended users, guest information is extremely unconfident.
- Reusing or retrieves of guest records is extremely difficult

- Human are required to fill forms and enter data, manual data entryleas to errors.

2.3 PROPOSED SYSTEM

Most of the modern applications have similar setups. Main aim of designing this project is to design a system that can manage and handle the activities that are involved in a hotel reservation in an organized, cost-effective and reliable way.

- **Python - front end**
- **MySQL- Database**

- It will help the administrative to do their work in very simple manner without causing any trouble and Visitors or customer to reserve their room according to needs before they come to hotel. Customer can use any electronic device like laptop, computers, smart phones or tablet etc to book and confirm their stay with hotel. Based on the limitation and shortcomings identified in the existing Hotel Room reservation system, the solutions have been provided for the better management to the organization and ease of access to the customer.
- The proposed solution will have ability to manage customer's account where they can save their preferences and personal information.
- Customer will have many options to compare rates and services based on their requirement. Efficient searching, sorting and filtering offered to the customer.
- Travellers or any other customers can book and secure their place to stay before or without actually visiting the hotel. Executives and business owner can easily manage the inventory and rates of their hotel rooms.

- With the new system business owner are capable to market their product on resme.com and also they can create some packages and deals based to offer to the customer.
- The proposed system will be able to handle payment and transactions.

2.4 MERITS OF PROPOSED SYSTEM

- Efficient data management and reusability.
- Highly accessible for users.
- No resource like time or money and papers is wasted in paper works neither we have to use so much human resources though it is a web-based application, all the data stores to database and we can retrieve data based on requirements.
- Security of User information and other can maintain.
- On a figure tip, we can access any information with proper authority.
- Customer can create and manage their account to save preferences, personal information and payment information.

2.5 TOOLS USED

2.5.1 HARDWARE AND SOFTWARE REQUIREMENTS

HARDWARE

- PROCESSOR: Intel core 2.0 GHz and more
- RAM: 4.00 GB and more
- MONITOR: 14"CRT or LCD monitor
- KEYBOARD: Normal or Multimedia
- MOUSE: Compatible mouse

- HARD DISK: 80 GB or more
- WINDOWS 10

SOFWARE

- **PYCHARM**
- **MYSQL**

2.5.2 INTRODUCTION TO PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Possum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

PYTHON FEATURES

Python's features include −

- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting. Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.

Tkinter is the standard GUI library for Python.

Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit. Import the Tkinter module.

2.5.4 INTRODUCTION TO MYSQL

MySQL is a fast, easy - to - use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons-

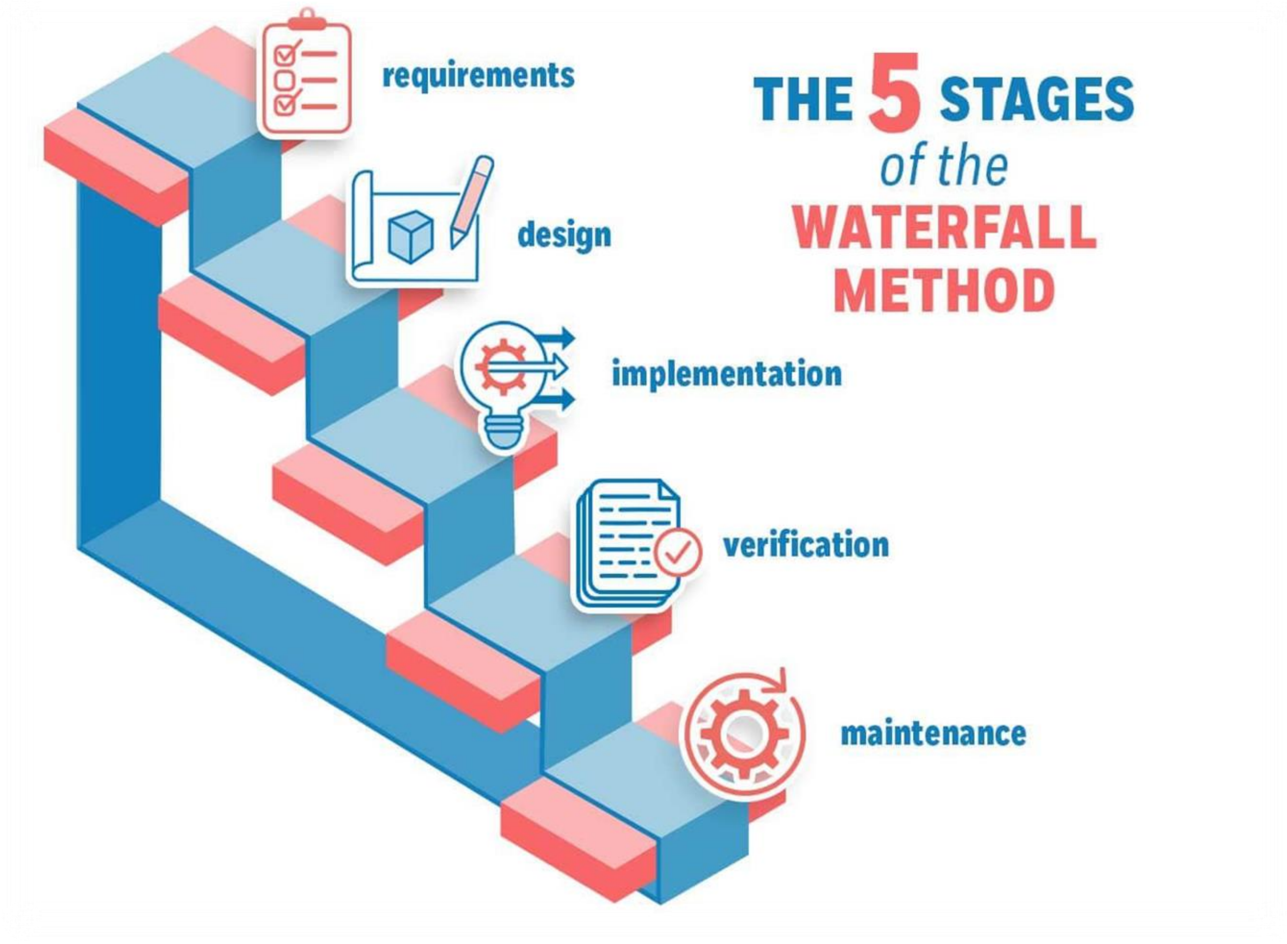
- MySQL is released under an open - source license. So, you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well - known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, and JAVA. Etc.
- MySQL works very quickly and works well even with large datasets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open - source GPL license allows programmers to modify the MySQL software to fit their own specific environments

SYSTEM ANALYSIS AND DESIGN

3.1 SOFTWARE REQUIREMENT SPECIFICATION

Software requirements are detailed versions of the user requirements. These specifications describe the external behavior of the software requirement specification of the whole software since it would be used for the implementation of the software. The final output of system requirement is the software requirements specification document. It is the official document that includes both, user requirements and detailed specifications of the software requirements.

There can be a diverse set of users of the SRS document such as software customers, managers, software engineers, software test engineers etc. Keeping in mind, the diversity of the users, the SRS document must communicate the requirements to customers, defining the requirements in detail for developers and testers. It should also include information about software evaluation.



WATER FALL MODEL:

The waterfall model is a breakdown of project activities into linear sequential phases, where each phase depends on the deliverables of the previous one and corresponds to a specialization of tasks. The approach is typical for certain areas of engineering design. In software development, it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, deployment and maintenance.

Waterfall development model originated in the manufacturing and construction industries; where the highly structured physical environments meant that design changes became prohibitively expensive much sooner in the development process. When first adopted for software development, there were no recognized alternatives for knowledge-based creative work

MODEL:

In Royce's original waterfall model, the following phases are followed in order:

1. System and software requirements: captured in a product requirements document
2. Analysis: resulting in models, schema, and business rules
3. Design: resulting in the software architecture
4. Coding: the development, proving, and integration of software
5. Testing: the systematic discovery and debugging of defects
6. Operations: the installation, migration, support, and maintenance of complete systems

Thus the waterfall model maintains that one should move to a phase only when it’s preceding phase is reviewed and verified.

Various modified waterfall models (including Royce's final model), however, can include slight or major variations on this process. These variations included returning to the previous cycle after flaws were found downstream or returning all the way to the design phase if downstream phases deemed insufficient

3.2 FEASIBILITY STUDY:

Feasibility analysis begins once the goals are defined its starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things-generate new ideas. There is a need to go into the detail system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization.

It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined.

3.2.1 ECONOMICAL FEASIBILITY

Economic feasibility is a kind of cost-benefit analysis of the examined project, which assesses whether it is possible to implement it. This term means the assessment and analysis of a project's potential to support the decision-making process by objectively and rationally identifying its strengths, weaknesses, opportunities and risks associated with it, the resources that will be needed to implement the project, and an assessment of its chances of success.

Development of these applications is highly economically feasible. The only to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the paperwork completely. The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.

3.2.2 TECHNICAL FEASIBILITY

Technical feasibility is one of the most important criteria for selecting material for digitization. The physical characteristics of source material and the project goals for capturing, presenting and storing the digital surrogates dictate the technical requirements. Libraries must evaluate those requirements for each project and determine whether they can be met with the resources available.

If the existing staff, hardware and software resources cannot meet the requirements, then the project will need funding to upgrade equipment or hire an outside conversion agency. If these resources are not available, or if the technology does not exist to meet the requirements, then it is not technically feasible to digitize that material.

The technical requirement for the system is economic and it does not use any other the additional hardware and software. Technical evaluation must also assess whether the existing systems can be upgraded to use the new technology and whether the organization as the expertise to use it.

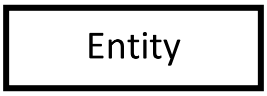
3.2.3 OPERATIONAL FEASIBILITY

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance includes issues such as determining whether system can provide the right information for the department personnel student details, and whether the system can be organized so that it always delivers this information at the right place on time using internal services.

3.3 SYSTEM DESIGN

3.3.1 ER DIAGRAM

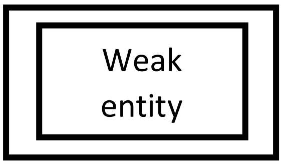
As entity relationship diagram is a data modeling technique that creates a graphical representation of the entities and the relationships between entities.



Entity

An entity is an object or concept about

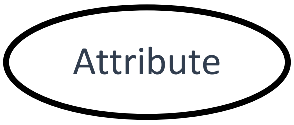
Which you want to store information.



Weak entity

A weak entity is dependent on another

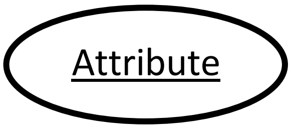
Entity to exist.



Attribute

Attributes are the properties or characteristics

Of an entity.



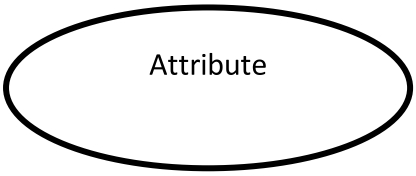
Key attribute

A key attribute is the unique, distinguishing

Characteristics of the entity.

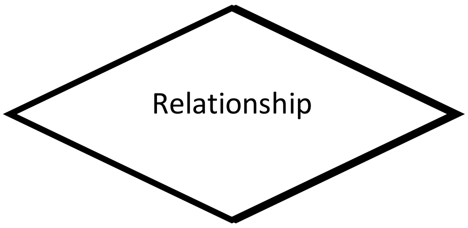
For example: an employee's social security

Number might be the employee's key attribute



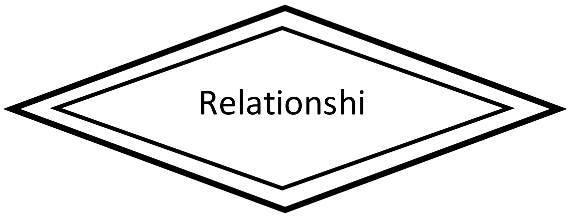
multi valued attribute

a multi valued attribute can have more than one value.



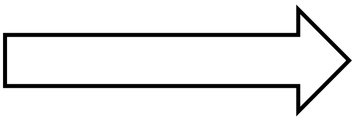
Relationship

Relationships illustrate how two entities Share information in the database Structure.



Weak relationship

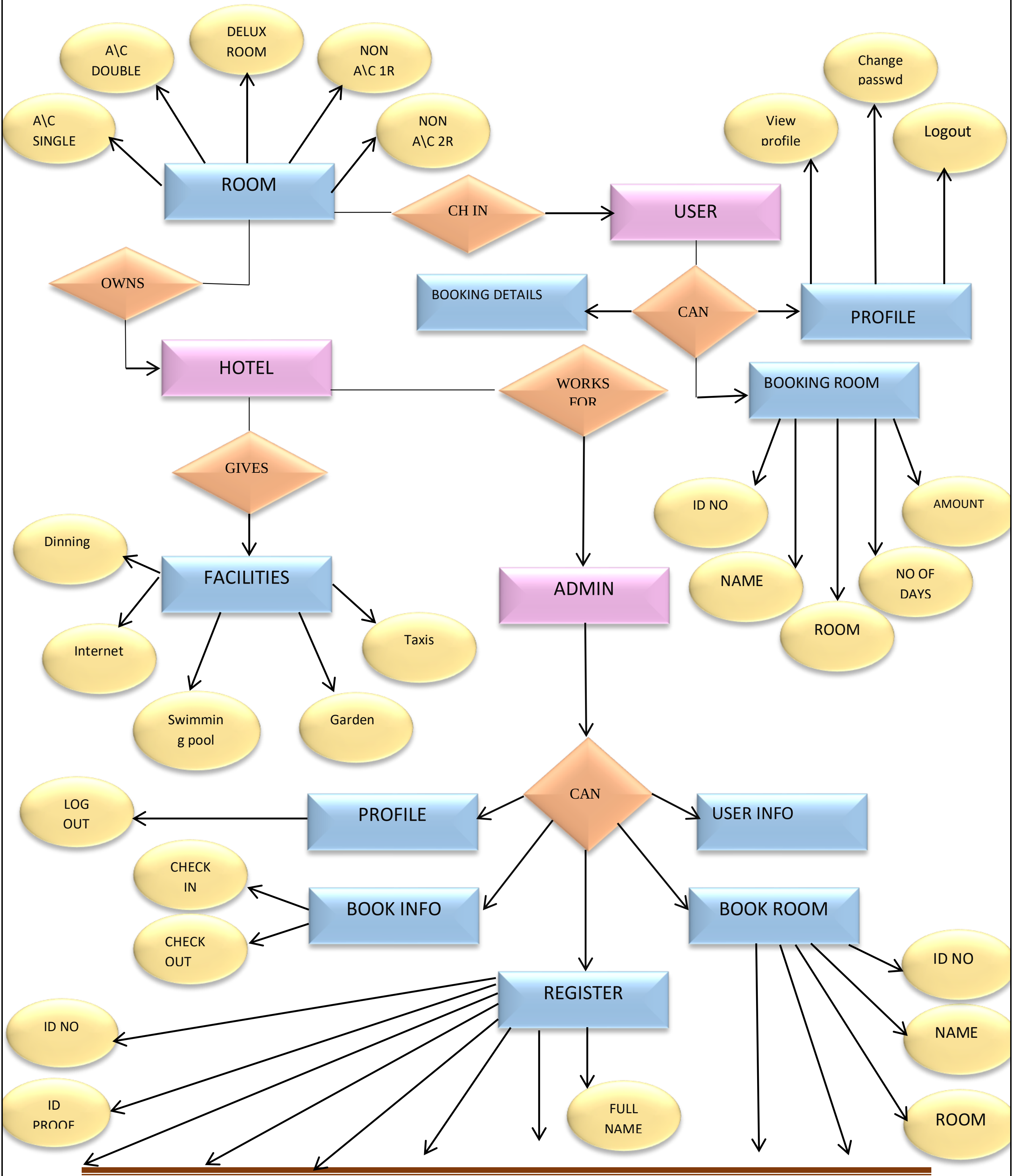
To connect a weak entity with others, You should use a weak relationship Notation.

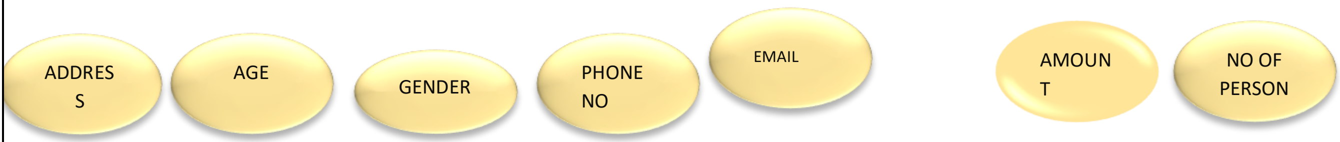


Flow control

To connect the difference controls in the ER diagram.

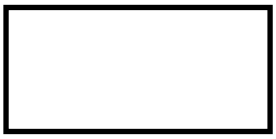
ENTITY RELATIONSHIP DIAGRAM



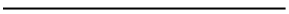


3.3.2 DATA FLOW DIAGRAM

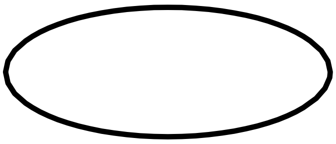
A data flow diagram (DFD) is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart. For each data flow, at least one of the endpoints (source and /or destination) must exist in a process. There are several notations for displaying data-flow-diagrams they are



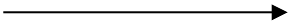
- Functional / Transformation



- Data Store

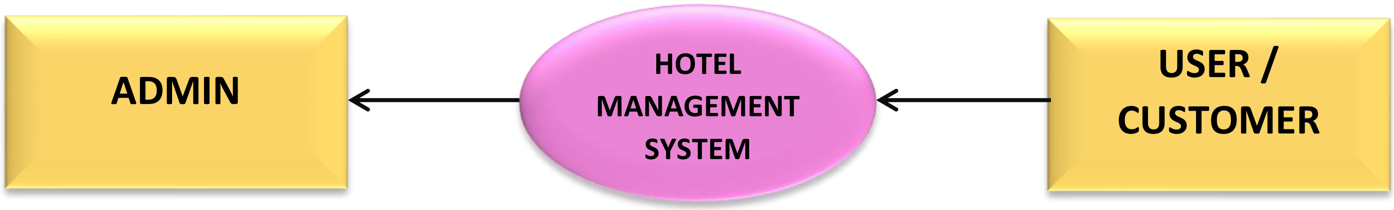


- Process

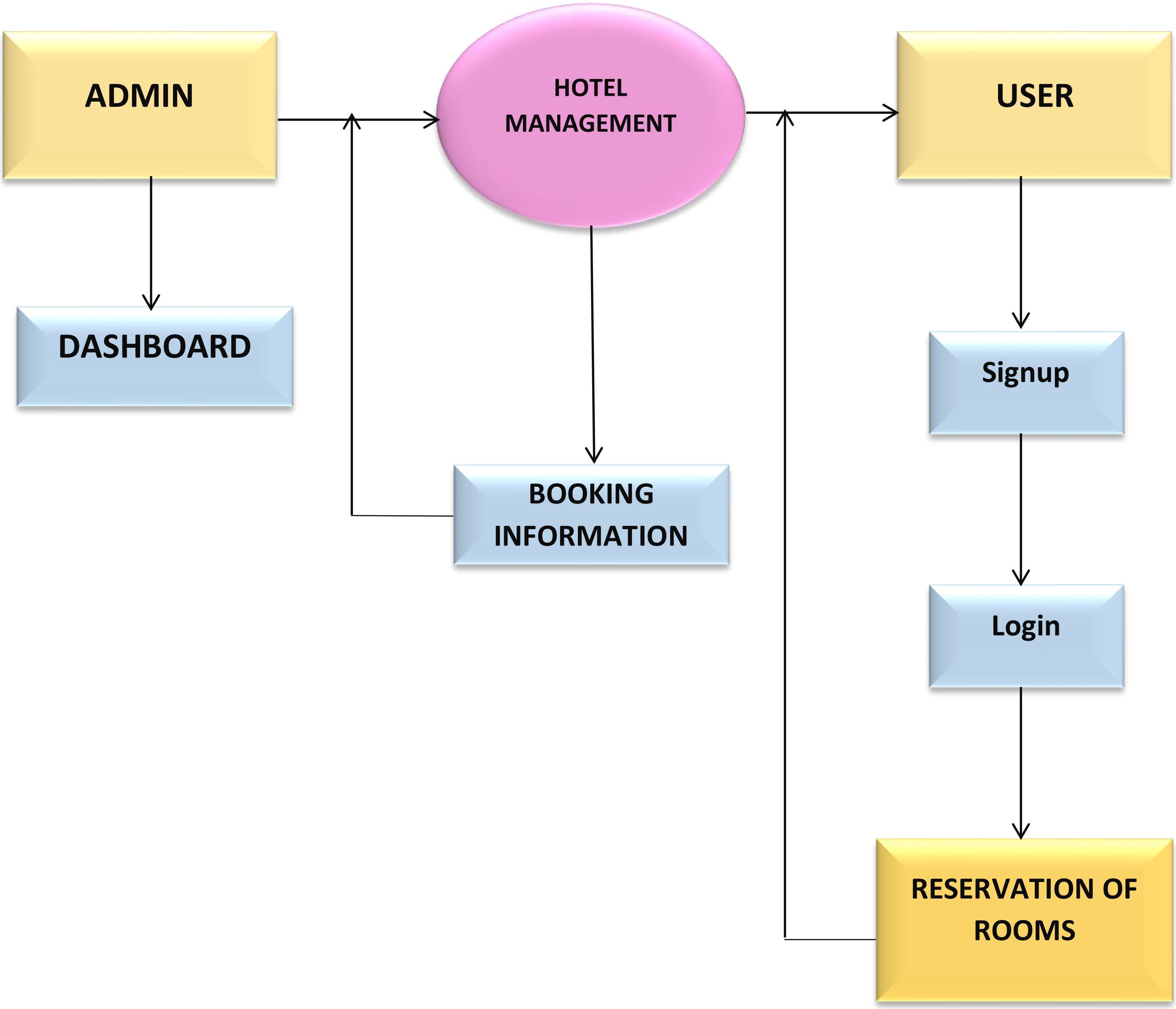


- Data Flow

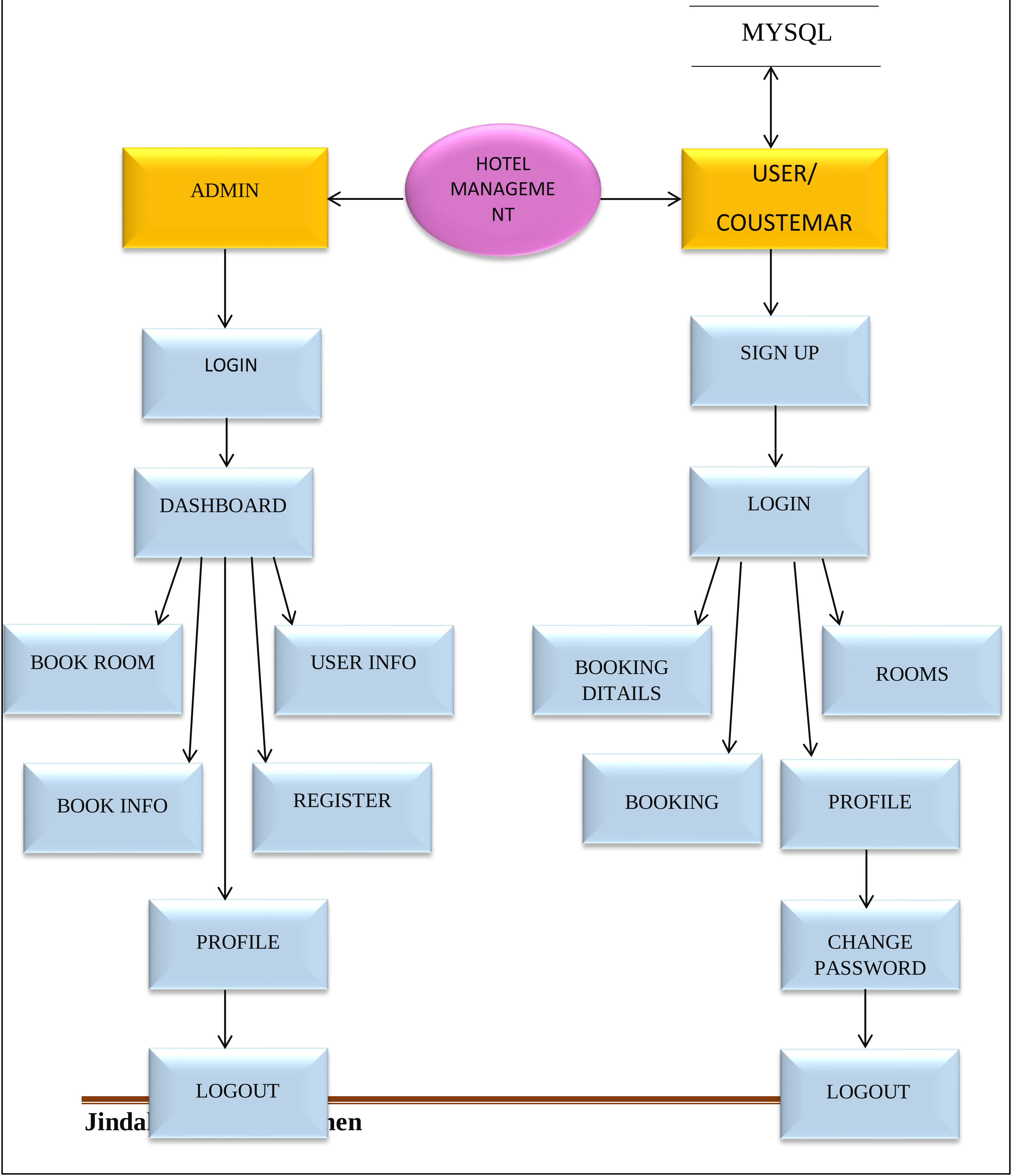
ZERO - LEVEL DATA FLOW DIAGRAM



FIRST – LEVEL DATA FLOW DIAGRAM



SECOND – LEVEL DATA FLOW DIAGRAM



3.4 MODULAR DESCRIPTION

Hotel Management System uses Python and MySQL Database. This project allows the customers to book the hotel rooms in online and keeps records of booking, clients, and hotel services. Hotel Booking Management System has two modules i.e. admin and user.

ADMIN MODULE

1. **Home:** When the admin clicks on this tab, it will display the other modules and also in this page admin can book the rooms for customers, and also he can view the booking details, user information so on.
2. **Reg Users**: In this section, admin can make registration of customers.
3. Room Booking: In this section, admin can book the rooms for the customer.
4. **User Info:** In this section, admin can view the user information by using the e-mail id of the customer.
5. **Booking Info:** In this section, admin can view the customer booking information that can be check-in and check-out. And also by entering the particular date on check-in and check-out admin can get the customer information between that dates specified.
6. **Profile:** In this section, admin can exit from the admin panel

USER MODULE

1. **Sing Up:** Users can register through sign up page.
2. **Log In**: It is login page for users.
3. **Home:** When the user clicks on this tab, it will display the other modules.
4. **Facilities:** In this section, user can view the services provided by the hotel
5. **Room Booking:** In this section, user can book the hotel room as per their requirement.
6. **Booking Details**: In this section, user can view the details of their booked room.
7. **Contact**: It is a contact page where customers can contact for the queries.
8. **Profile:** In this section, user can change the password, update the profile and also can logout

CODING

CODING

USER PAGE

LOGIN

```
from tkinter import *

from tkinter import messagebox

import query2 as q


root = Tk()

#root.overrideredirect(1)

root.title('login page')

root.geometry('920x500+300+100')

root.configure(bg="#fff")

root.resizable(False, False)


#pas=PhotoImage(file='C:\\Users\\Admin\\PycharmProjects\\hotel\\pass.png')

img = PhotoImage(file='C:\\Users\\sys\\Downloads\\hotel\\hotel\\login.png')

Label(root, image=img, bg='white').place(x=50, y=50)


frame = Frame(root, width=350, height=350, bg="white")

frame.place(x=480, y=70)


heading = Label(frame, text='Sign In', fg='#57a1f8', bg='white', font=('Microsoft YaHei UI
Light', 23, 'bold'))

heading.place(x=100, y=5)


def signin():

    em=user.get()

    pas=password.get()

    r=q.login(em,pas)
    print(r)
    if r=="T":
        q.login_insert(em)
        root.destroy()
        import customer
    else:
        messagebox.showerror("error!","Invalid email or  password!")

def on_enter(e):
    user.delete(0, 'end')


def on_leave(e):
    name = user.get()
    if name == '':
        user.insert(0, 'Email')



user = Entry(frame, width=25, fg='black', border=0, bg="white", font=('Microsoft YaHei UI
Light', 11))
user.place(x=30, y=80)
user.insert(0, 'Email')
user.bind("<FocusIn>", on_enter)
user.bind("<FocusOut>", on_leave)

Frame(frame, width=295, height=2, bg='black').place(x=25, y=107)

def on_enter(e):
    password.delete(0, 'end')
    password.config(show="*")



def on_leave(e):
    name = password.get()
    if name == '':
        password.config(show="")
        password.insert(0, 'password')

def show():
    if(sh_v.get()==1):
        password.config(show='')
    else:
        password.config(show='*')

password = Entry(frame, width=25, fg='black', border=0, bg="white", font=('Microsoft YaHei UI Light', 11))
password.place(x=30, y=150)
password.insert(0, "password")
password.bind("<FocusIn>", on_enter)
password.bind("<FocusOut>", on_leave)

Frame(frame, width=295, height=2, bg='black').place(x=25, y=177)

sh_v=IntVar(value=0)

sh= Checkbutton(frame,width=6, text='($)',variable=sh_v, border=0, bg='white', cursor='hand2', fg='#57a1f8', command=show)
sh.place(x=300, y=150)

Button(frame, width=39,text="sign In", pady=7, bg='#57a1f8', fg='white', border=0,command=signin).place(x=35, y=204)
label=Label(frame, text="Don't have an account?", fg='black', bg='white', font=('Microsoft YaHei UI Light', 9))
label.place(x=75,y=270)

def signup():
    root.destroy()

    import register


sign_up= Button(frame,width=6, text='Sign Up', border=0, bg='white', cursor='hand2', 
fg='#57a1f8', command=signup)

sign_up.place(x=215, y=270)


root.mainloop()
```

REGISTER PAGE

```
from tkinter import *
from tkinter import messagebox
import query2 as q1
import re


root = Tk()
# root.overrideredirect(1)
root.title('login page')
root.geometry('500x500+300+100')
root.configure(bg="#fff")
root.resizable(False, False)


def val(n, m, p, g, a, ad, idf, idn, pa, pas):
    if n == "" or m == "" or p == "" or g == "" or a == "" or ad == "" or idf == "" or idn == ""
or pa == "" or pas == "":
        messagebox.showerror("ERROR", "Fields cannot be Blank")
    else:
        validate_name(n, m, p, g, a, ad, idf, idn, pa, pas)


def validate_name(n, m, p, g, a, ad, idf, idn, pa, pas):
    if (n.isalpha()):
        validate_email(n, m, p, g, a, ad, idf, idn, pa, pas)
    else:
        messagebox.showerror("ERROR", "Invalid Name")


def validate_email(n, m, p, g, a, ad, idf, idn, pa, pas):
    pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
    if (re.fullmatch(pattern, m)):
        validate_ph(n, m, p, g, a, ad, idf, idn, pa, pas)
    else:
        messagebox.showerror("ERROR", "Invalid email")


def validate_ph(n, m, p, g, a, ad, idf, idn, pa, pas):
    r = re.fullmatch('[6-9][0-9]{9}', p)
    if r != None:
        validate_age(n, m, p, g, a, ad, idf, idn, pa, pas)
    else:
        messagebox.showerror("ERROR", "Invalid Phone Number")


def validate_age(n, m, p, g, a, ad, idf, idn, pa, pas):
    r = re.fullmatch('[1-9]{2}', a)
    if r != None and int(a) <= 100:
        validate_pas(n, m, p, g, a, ad, idf, idn, pa, pas)
    else:
        messagebox.showerror("ERROR", "Invalid age")


def validate_pas(n, m, p, g, a, ad, idf, idn, pa, pas):
    if pa != pas:
        messagebox.showerror("ERROR", "Passwords don't match")
    else:
        register(n, m, p, g, a, ad, idf, idn, pa, pas)

def register(n, m, p, g, a, ad, idf, idn, pa, pas):
    qa=q1.register(n, m, p, g, a, ad, idf, idn, pa, pas)
    if qa=="T":
        messagebox.showinfo("sucess", "Register sucessfully_/\__/\_")
        fullname_entry.delete(0, END)
        email_entry.delete(0, END)
        Phone_entry.delete(0, END)
        age_entry.delete(0, END)
        add_entry.delete(0, END)
        idno_entry.delete(0, END)
        pas_entry.delete(0, END)
        conpas_entry.delete(0, END)


def signup():
    name = fullname_entry.get()
    mail = email_entry.get()
    ph = Phone_entry.get()
    gen = menu1.get()
    ag = age_entry.get()
    ad = add_entry.get()
    idf = menu.get()
    idn = idno_entry.get()
add = Label(text='Address', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10,
'bold'))
add.place(x=100, y=230)

add_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))
add_entry.place(x=190, y=230)

id = Label(text='ID proof', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10,
'bold'))
id.place(x=100, y=270)

idpr = ["Adhaar", "Voter ID", "Driving Licence", "PAN"]
menu = StringVar()
menu.set("select id proof")
drop = OptionMenu(root, menu, *idpr).place(x=190, y=270)

idno = Label(text='ID Number', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10,
'bold'))
idno.place(x=100, y=300)

idno_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))
idno_entry.place(x=190, y=300)

pas = Label(text='Password', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10,
'bold'))
pas.place(x=100, y=330)

pas_entry = Entry(show="*", width=20, fg='black', bg="white", font=('Microsoft YaHei UI
Light', 11))
pas_entry.place(x=190, y=330)


def show():
    if (sh_v.get() == 1):
        pas_entry.config(show='')
    else:
        pas_entry.config(show='*')


sh_v = IntVar(value=0)

sh = Checkbutton(width=6, text='($)', variable=sh_v, border=0, bg='white', cursor='hand2',
fg='#57a1f8', command=show)
sh.place(x=350, y=330)

conpas = Label(text='Confirm Password', fg='black', bg='white', font=('Microsoft YaHei UI
Light', 10, 'bold'))
conpas.place(x=40, y=370)
```

```
conpas_entry = Entry(show="*", width=20, fg='black', bg="white", font=('Microsoft YaHei
UI Light', 11))
conpas_entry.place(x=190, y=370)


def show():
    if (sh1_v.get() == 1):
        conpas_entry.config(show='')
    else:
        conpas_entry.config(show='*')


sh1_v = IntVar(value=0)

sh1 = Checkbutton(width=6, text='($)', variable=sh1_v, border=0, bg='white', cursor='hand2',
fg='#57a1f8', command=show)
sh1.place(x=350, y=370)

sign_up = Button(width=10, text='Sign Up', bg='white', cursor='hand2', fg='#57a1f8',
command=signup)
sign_up.place(x=180, y=420)

label = Label(text="Already have as Account?", fg='black', bg='white', font=('Microsoft
YaHei UI Light', 9))
label.place(x=100, y=450)


def signin():
    root.destroy()
    import login


sign_in = Button(width=6, text='Sign In', border=0, bg='white', cursor='hand2', fg='#57a1f8',
command=signin)
sign_in.place(x=250, y=450)

root.mainloop()
```

BOOKING PAGE

```
from tkinter import *
from tkinter import messagebox
from tkcalendar import *
from datetime import date
import os
import query2 as q
```

```
today = date.today()
root3 = Tk()
#root3.overrideredirect(1)
root3.title('contact')
#root3.geometry('1366x768+0+0')
root3.geometry('610x610+300+100')
root3.configure(bg="#fff")
root3.resizable(False, False)



def home():
    root3.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\customer.py')

def contact():
    root3.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\contact.py')

def room():
    root3.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\rooms.py')

def booking():
    root3.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\booking.py')

def bookdetails():
    root3.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\bookdetails.py')

def profile():
    root3.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\profile.py')

def val(idn,n,r,rn,ci,co,nod,am):
    if idn=="" or n=="" or r=="" or rn=="" or ci=="" or co=="" or am=="" or nod=="":
        messagebox.showerror("ERROR","Fields cannot be Blank")
    else:
        sd=q.insert_book(idn,n,r,rn,ci,co,nod,am)
        if sd=="T":
            messagebox.showinfo("Sucesss", "room has been booked")
            root3.destroy()
            os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\customer.py')



def calc():
    num=0
    ch=0
```

```
ch0=0
ch=cal.get_date()
ch0=cal1.get_date()
num= (ch0-ch).days
noro_entry.insert(0,num)
noro_entry.config(state="normal")
noro_entry.delete(0, END)
noro_entry.insert(0, num)
noro_entry.config(state="disable")

sig= 1500
dob=2000
delux=2500
NonACsingle=700
NonACDouble=1000
BigRoom=3000

if menu1.get()=="A\C single room ":
    amoun = int(num * sig)
    amount_entry.config(state="normal")
    amount_entry.delete(0, END)
    amount_entry.insert(0, amoun)
    amount_entry.config(state="disable")
    non_entry.config(state="normal")
    non_entry.delete(0, END)
    non_entry.insert(0, "1")
    non_entry.config(state="disable")
elif menu1.get()=="A\C Double room ":
     amoun = int(num * dob)
     amount_entry.config(state="normal")
     amount_entry.delete(0, END)
     amount_entry.insert(0, amoun)
     amount_entry.config(state="disable")
     non_entry.config(state="normal")
     non_entry.delete(0, END)
     non_entry.insert(0, "2")
     non_entry.config(state="disable")
elif menu1.get()=="delux room":
      amoun = int(num * delux)
      amount_entry.config(state="normal")
      amount_entry.delete(0, END)
      amount_entry.insert(0, amoun)
      amount_entry.config(state="disable")
      non_entry.config(state="normal")
      non_entry.delete(0, END)
      non_entry.insert(0, "2")
      non_entry.config(state="disable")
elif menu1.get()=="Non A\C single room ":
      amoun = int(num * NonACsingle)
      amount_entry.config(state="normal")
```

```
            amount_entry.delete(0, END)
            amount_entry.insert(0, amoun)
            amount_entry.config(state="disable")
            non_entry.config(state="normal")
            non_entry.delete(0, END)
            non_entry.insert(0, "1")
            non_entry.config(state="disable")
    elif menu1.get()=="Non A\C Double room":
            amoun = int(num * NonACDouble)
            amount_entry.config(state="normal")
            amount_entry.delete(0, END)
            amount_entry.insert(0, amoun)
            amount_entry.config(state="disable")
            non_entry.config(state="normal")
            non_entry.delete(0, END)
            non_entry.insert(0, "2")
            non_entry.config(state="disable")
    elif menu1.get()=="Big Room ":
            amoun = int(num * BigRoom)
            amount_entry.config(state="normal")
            amount_entry.delete(0, END)
            amount_entry.insert(0, amoun)
            amount_entry.config(state="disable")
            non_entry.config(state="normal")
            non_entry.delete(0, END)
            non_entry.insert(0, "max 4")
            non_entry.config(state="disable")



def bookt():
    idn=id_entry.get()
    name= name_entry.get()
    ro= menu1.get()
    noper= non_entry.get()
    amoun=amount_entry.get()
    chein=cal.get()
    cheou=cal1.get()
    nod=noro_entry.get()
    val(idn,name,ro,noper,chein,cheou,nod,amoun)


i = q.get_login()
for data in i:
    id1=data[0]
    em=data[1]
    break

d = q.get_book(id1)
for data1 in d:
    name1 = data1[0]




frame = Frame(root3, width=650, height=650, bg="white")
frame.place(x=0, y=30)

Button(root3,width=15,text="Home", pady=7, bg='#57a1f8', fg='white', border=0,
command=home).place(x=0, y=0)
Button(root3,width=15,text="About", pady=7, bg='#57a1f8', fg='white',
border=0,command=room).place(x=80, y=0)
Button(root3,width=15,text="Booking", pady=7, bg='#57a1f8', fg='white',
border=0,command=booking).place(x=160, y=0)
Button(root3,width=17,text="Booking details", pady=7, bg='#57a1f8', fg='white',
border=0,command=bookdetails).place(x=270, y=0)
Button(root3,width=15,text="Contact us", pady=7, bg='#57a1f8', fg='white', border=0,
command=contact).place(x=390, y=0)
Button(root3,width=15,text="profile", pady=7, bg='#57a1f8', fg='white',
border=0,command=profile).place(x=500, y=0)

heading = Label(text='BOOK ROOM', fg='blue', bg='white', font=('Microsoft YaHei UI
Light', 15, 'bold'))
heading.place(x=250, y=50)

id= Label(text='ID Number',fg='black', bg='white', font=('Microsoft YaHei UI Light', 10,
'bold'))
id.place(x=200, y=90)

id_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))
id_entry.place(x=300, y=90)
id_entry.insert(0, id1)
id_entry.config(state= "disabled")

name= Label(text='Name',fg='black', bg='white', font=('Microsoft YaHei UI Light', 10,
'bold'))
name.place(x=200, y=120)

name_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light',
11))
name_entry.place(x=300, y=120)
name_entry.insert(0,name1)
name_entry.config(state= "disabled")

room= Label(text='Room',fg='black', bg='white', font=('Microsoft YaHei UI Light', 10,
'bold'))
room.place(x=200, y=160)

rom=["A\C single room ","A\C Double room ", "delux room","Non A\C single room","Non
A\C Double room ","Big Room "]
```

```
menu1= StringVar()
book.place(x=300, y=390)

root3.mainloop()
```

BOOKING DETAILS PAGE

```
from tkinter import *
from tkinter import messagebox
import os
import query2 as q

root4 = Tk()
root4.overrideredirect(1)
root4.title('login page')
root4.geometry('610x610+300+100')
root4.configure(bg="#fff")
root4.resizable(False, False)

def home():
    root4.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\customer.py')

def contact():
    root4.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\contact.py')

def room():
    root4.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\rooms.py')

def booking():
    root4.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\booking.py')

def bookdetails():
    root4.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\bookdetails.py')

def profile():
    root4.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\profile.py')

frame = Frame(root4, width=650, height=650, bg="white")
frame.place(x=0, y=50)
```

```
Button(root4,width=15,text="Home", pady=7, bg='#57a1f8', fg='white', border=0,
command=home).place(x=0, y=0)
Button(root4,width=15,text="About", pady=7, bg='#57a1f8', fg='white',
border=0,command=room).place(x=80, y=0)
Button(root4,width=15,text="Booking", pady=7, bg='#57a1f8', fg='white',
border=0,command=booking).place(x=160, y=0)
Button(root4,width=17,text="Booking details", pady=7, bg='#57a1f8', fg='white',
border=0,command=bookdetails).place(x=270, y=0)
Button(root4,width=15,text="Contact us", pady=7, bg='#57a1f8', fg='white', border=0,
command=contact).place(x=390, y=0)
Button(root4,width=15,text="profile", pady=7, bg='#57a1f8', fg='white',
border=0,command=profile).place(x=500, y=0)

i = q.get_login()
for data in i:
    id1=data[0]
    em=data[1]

lis=q.get_booking(id1)

i=0
for bo in lis:
    for j in range(len(bo)):
        e = Entry(frame, width=14, fg='blue')
        e.grid(row=i, column=j)
        e.insert(END, bo[j])
    i=i+1


root4.mainloop()
```

PROFILE PAGE

```
from tkinter import *
from tkinter import messagebox
import os
import query2 as q

root5 = Tk()
#root5.overrideredirect(1)
root5.title('login page')
root5.geometry('610x610+300+100')
root5.configure(bg="#fff")
root5.resizable(False, False)

def home():
    root5.destroy()
    os.startfile('customer.py')

def contact():
    root5.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\contact.py')

def room():
    root5.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\rooms.py')

def booking():
    root5.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\booking.py')

def bookdetails():
    root5.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\bookdetails.py')

def profile():
    root5.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\profile.py')

def viewprof():
    root5.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\viewprofile.py')

def changepass():
    root5.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\hotel\\hotel\\changepas.py')

def logout():
    q.logout()
    root5.destroy()
    import login


i = q.get_login()
for data in i:
    id1=data[0]
    em=data[1]

d = q.get_book(id1)
for data in d:
    name1 = data[0]

frame = Frame(root5, width=650, height=650, bg="white")
frame.place(x=0, y=30)


Button(root5,width=15,text="Home", pady=7, bg='#57a1f8', fg='white', border=0,
command=home).place(x=0, y=0)
Button(root5,width=15,text="About", pady=7, bg='#57a1f8', fg='white',
border=0,command=room).place(x=80, y=0)
```

```
Button(root5,width=15,text="Booking", pady=7, bg='#57a1f8', fg='white',
border=0,command=booking).place(x=160, y=0)
Button(root5,width=17,text="Booking details", pady=7, bg='#57a1f8', fg='white',
border=0,command=bookdetails).place(x=270, y=0)
Button(root5,width=15,text="Contact us", pady=7, bg='#57a1f8', fg='white', border=0,
command=contact).place(x=390, y=0)
Button(root5,width=15,text="profile", pady=7, bg='#57a1f8', fg='white',
border=0,command=profile).place(x=500, y=0)

imgcus = PhotoImage(file='C:\\Users\\sys\\Downloads\\hotel\\hotel\\pro.png')
lab= Label(root5, image=imgcus, bg='white')
lab.place(x=200, y=40)

label1=Label(frame, text=name1, fg='Black', bg='white', font=('Microsoft YaHei UI Light',
12, 'bold'))
label1.place(x=270,y=250)

view= Button(frame,width=45, text='View Profile', border=0, bg='white', cursor='hand2',
fg='#57a1f8', font=('Microsoft YaHei UI Light', 12, 'bold'), command=viewprof)
view.place(x=80, y=280)

log= Button(frame,width=45, text='Change password', border=0, bg='white', cursor='hand2',
fg='#57a1f8', font=('Microsoft YaHei UI Light', 12, 'bold'),command=changepass)
log.place(x=80, y=310)

log= Button(frame,width=45, text='Logout', border=0, bg='white', cursor='hand2',
fg='#57a1f8', font=('Microsoft YaHei UI Light', 12, 'bold'),command=logout)
log.place(x=80, y=350)

root5.mainloop()
```

ADMIN PAGE

BOOKING INFO

```
from tkinter import *

from tkinter import messagebox

import query2 as q1

import re
```

```
root = Tk()
# root.overrideredirect(1)
root.title('login page')
root.geometry('500x500+300+200')
root.configure(bg="#fff")
root.resizable(False, False)



def val(n, m, p, g, a, ad, idf, idn, pa, pas):
    if n == "" or m == "" or p == "" or g == "" or a == "" or ad == "" or idf == "" or idn == "" or pa == "" or pas == "":
        messagebox.showerror("ERROR", "Fields cannot be Blank")
    else:
        validate_name(n, m, p, g, a, ad, idf, idn, pa, pas)



def validate_name(n, m, p, g, a, ad, idf, idn, pa, pas):
    if (n.isalpha()):
        validate_email(n, m, p, g, a, ad, idf, idn, pa, pas)
    else:
        messagebox.showerror("ERROR", "invalid Name")



def validate_email(n, m, p, g, a, ad, idf, idn, pa, pas):
    else:
        register(n, m, p, g, a, ad, idf, idn, pa, pas)

def register(n, m, p, g, a, ad, idf, idn, pa, pas):
    qa=q1.register(n, m, p, g, a, ad, idf, idn, pa, pas)
    if qa=="T":
        messagebox.showinfo("sucess", "register sucessfully")
        fullname_entry.delete(0, END)
        email_entry.delete(0, END)
        Phone_entry.delete(0, END)
        age_entry.delete(0, END)
        add_entry.delete(0, END)
        idno_entry.delete(0, END)
        pas_entry.delete(0, END)
        conpas_entry.delete(0, END)



def signup():
    name = fullname_entry.get()
    mail = email_entry.get()
    ph = Phone_entry.get()
    gen = menu1.get()
    ag = age_entry.get()
    ad = add_entry.get()
    idf = menu.get()
    idn = idno_entry.get()
```

```
Phone_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))
Phone_entry.place(x=190, y=120)

gender = Label(text='Gender', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10, 'bold'))
gender.place(x=100, y=170)

gen = ["Male", "Female", "Others"]
menu1 = StringVar()
menu1.set("select gender")
drop1 = OptionMenu(root, menu1, *gen).place(x=190, y=170)

age = Label(text='Age', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10, 'bold'))
age.place(x=100, y=200)

age_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))
age_entry.place(x=190, y=200)

add = Label(text='Address', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10, 'bold'))
add.place(x=100, y=230)

add_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))
```

```
add_entry.place(x=190, y=230)

id = Label(text='ID proof', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10, 'bold'))
id.place(x=100, y=270)

idpr = ["Adhaar", "Voter ID", "Driving Lisence", "PAN"]
menu = StringVar()
menu.set("select id proof")
drop = OptionMenu(root, menu, *idpr).place(x=190, y=270)

idno = Label(text='ID Number', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10, 'bold'))
idno.place(x=100, y=300)

idno_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))
idno_entry.place(x=190, y=300)

pas = Label(text='Password', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10, 'bold'))
pas.place(x=100, y=330)

pas_entry = Entry(show="*", width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))
pas_entry.place(x=190, y=330)
```

```
def show():
    if (sh_v.get() == 1):
        pas_entry.config(show='')
    else:
        pas_entry.config(show='*')



sh_v = IntVar(value=0)


sh = Checkbutton(width=6, text='($)', variable=sh_v, border=0, bg='white',
cursor='hand2', fg='#57a1f8', command=show)
sh.place(x=350, y=330)


conpas = Label(text='Confirm Password', fg='black', bg='white',
font=('Microsoft YaHei UI Light', 10, 'bold'))
conpas.place(x=40, y=370)


conpas_entry = Entry(show="*", width=20, fg='black', bg="white",
font=('Microsoft YaHei UI Light', 11))
conpas_entry.place(x=190, y=370)



def show():
    if (sh1_v.get() == 1):
        conpas_entry.config(show='')
    else:
        conpas_entry.config(show='*')
```

```
sh1_v = IntVar(value=0)


sh1 = Checkbutton(width=6, text='($)', variable=sh1_v, border=0, bg='white',
cursor='hand2', fg='#57a1f8', command=show)
sh1.place(x=350, y=370)


sign_up = Button(width=10, text='sign Up', bg='white', cursor='hand2',
fg='#57a1f8', command=signup)
sign_up.place(x=180, y=420)


label = Label(text="Already have as Account?", fg='black', bg='white',
font=('Microsoft YaHei UI Light', 9))
label.place(x=100, y=450)



def signin():
    root.destroy()
    import login



sign_in = Button(width=6, text='sign In', border=0, bg='white', cursor='hand2',
fg='#57a1f8', command=signin)
sign_in.place(x=250, y=450)


root.mainloop()
```

USER INFO

```
from tkinter import *

from tkinter import messagebox

import os

import query2 as q

import re


root24 = Tk()

root24.overrideredirect(1)

root24.title('login page')

root24.geometry('610x610+300+200')

root24.configure(bg="#fff")

root24.resizable(False, False)


def home():

    root24.destroy()

    os.startfile('C:\\Users\\sys\\Downloads\\fullproject\\hotel\\admin_home.py')


def reg():

    root24.destroy()

    os.startfile('C:\\Users\\sys\\Downloads\\fullproject\\hotel\\admin_register.py')


def bookr():

    root24.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\fullproject\\hotel\\admin_book.py')

def user():
    root24.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\fullproject\\hotel\\admin_user.py')

def bookin():
    root24.destroy()

os.startfile('C:\\Users\\sys\\Downloads\\fullproject\\hotel\\admin_bookinfo.py')

def profile():
    root24.destroy()
    os.startfile('C:\\Users\\sys\\Downloads\\fullproject\\hotel\\admin_profile.py')

def show(lis):
    i=0
    for bo in lis:
        for j in range(len(bo)):
            e = Entry(frame1, width=14, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, bo[j])
        i=i+1

def show_val(lis):
    if lis==0:
        messagebox.showerror("ERROR", "No user found")
    else:
        show(lis)



def validate_email(em):
    pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
    if (re.fullmatch(pattern, em)):
         lis=q.get_user(em)
         show_val(lis)
    else:
        messagebox.showerror("ERROR", "invalid email")


def val():
    email = fullname_entry.get()
    validate_email(email)


 frame = Frame(root24, width=650, height=650, bg="white")
frame.place(x=0, y=30)


img =
PhotoImage(file='C:\\Users\\sys\\Downloads\\fullproject\\hotel\\hotel4.png')
Label(frame, image=img, bg='white').place(x=200, y=120)


Button(root24,width=15,text="Home", pady=7, bg='#57a1f8', fg='white',
border=0, command=home).place(x=0, y=0)
```

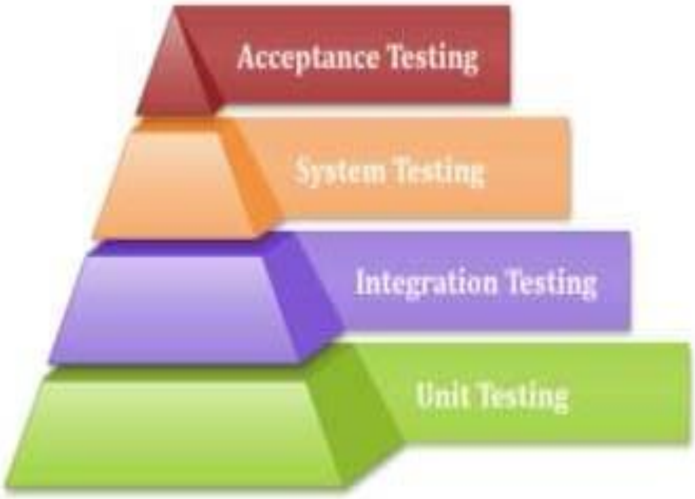
```
Button(root24,width=15,text="Register", pady=7, bg='#57a1f8', fg='white', border=0,command=reg).place(x=80, y=0)

Button(root24,width=15,text="book room", pady=7, bg='#57a1f8', fg='white', border=0,command=bookr).place(x=160, y=0)

Button(root24,width=17,text="User info", pady=7, bg='#57a1f8', fg='white', border=0,command=user).place(x=270, y=0)

Button(root24,width=15,text="booking info", pady=7, bg='#57a1f8', fg='white', border=0, command=bookin).place(x=390, y=0)

Button(root24,width=15,text="profile", pady=7, bg='#57a1f8', fg='white', border=0,command=profile).place(x=500, y=0)


frame = Frame(root24, width=650, height=650, bg="white")

frame.place(x=0, y=30)


fullname = Label(text='email', fg='black', bg='white', font=('Microsoft YaHei UI Light', 10, 'bold'))

fullname.place(x=130, y=60)


fullname_entry = Entry(width=20, fg='black', bg="white", font=('Microsoft YaHei UI Light', 11))

fullname_entry.place(x=220, y=60)
```

SOFTWARE TESTING

5. SOFTWARE TESTING

SOFTWARE TESTING

There are many different testing levels which help to check behavior and performance for software testing. These testing levels are designed to recognize missing areas and reconciliation between the development lifecycle states.



1) UNIT TESTING:

A Unit is a smallest testable portion of system or application which can be compiled, liked, loaded, and executed. This kind of testing helps to test each module separately.

The aim is to test each part of the software by separating it. It checks that component is fulfilling functionalities or not. This kind of testing is performed by developers.

2) INTEGRATION TESTING:

Integration means combining. For Example, in this testing phase, different software modules are combined and tested as a group to make sure that integrated system is ready for system testing.

Integrating testing checks the data flow from one module to other modules. This kind of testing is performed by testers.

3) SYSTEM TESTING:

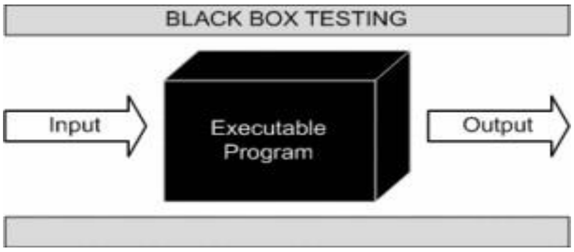
System testing is performed on a complete, integrated system. It allows checking system’s compliance as per the requirements. It tests the overall interaction of components. It involves load, performance, reliability and security testing. System testing most often the final test to verify that the system meets the specification. It evaluates both functional and non-functional need for the testing.

4) ACCEPTANCE TESTING:

Acceptance testing is a test conducted to find if the requirements of a specification or contract are met as per its delivery. Acceptance testing is basically done by the user or customer. However, other stockholders can be involved in this process.

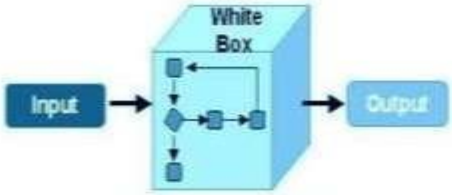
TYPES OF TESTING:

Black Box Testing



Black box testing is defined as a testing technique in which functionality of the Application under Test is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software. This type of testing is based entirely on software requirements and specifications. In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

WHITE BOX TESTING



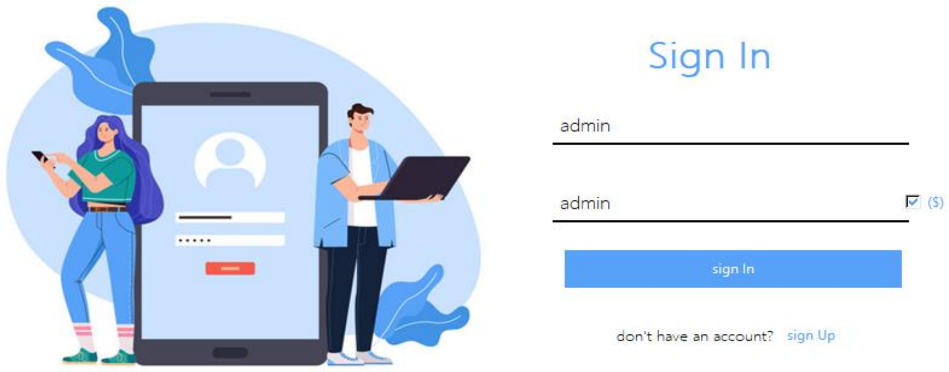
White-box testing is also known as clear box testing, glass box testing, transparent box testing. It is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and determine the expected outputs.

RESULTS\OUTPUT

RESULTS\OUTPUT

USER PAGE

LOGIN PAGE



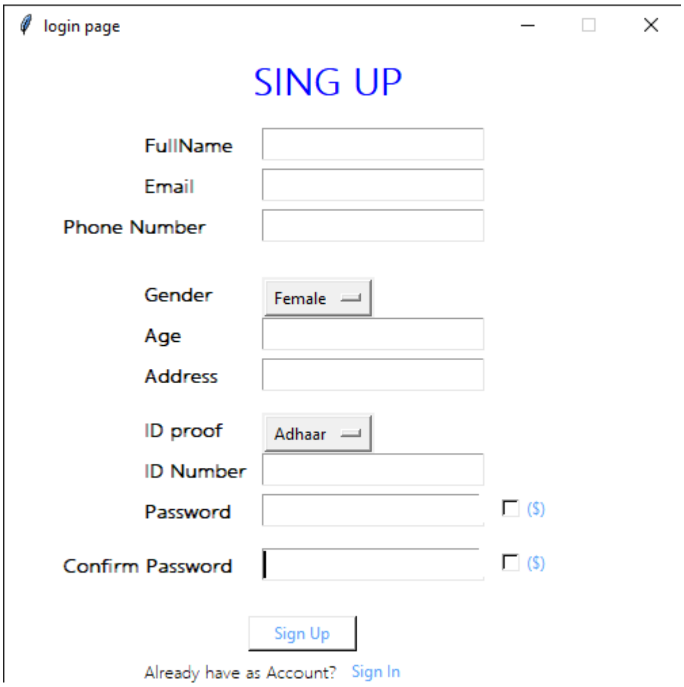
Sign In

admin

admin ☑ (S)

don't have an account? sign Up

SIGNUP PAGE



login page

SING UP

FullName

Email

Phone Number

Gender Female

Age

Address

ID proof Adhaar

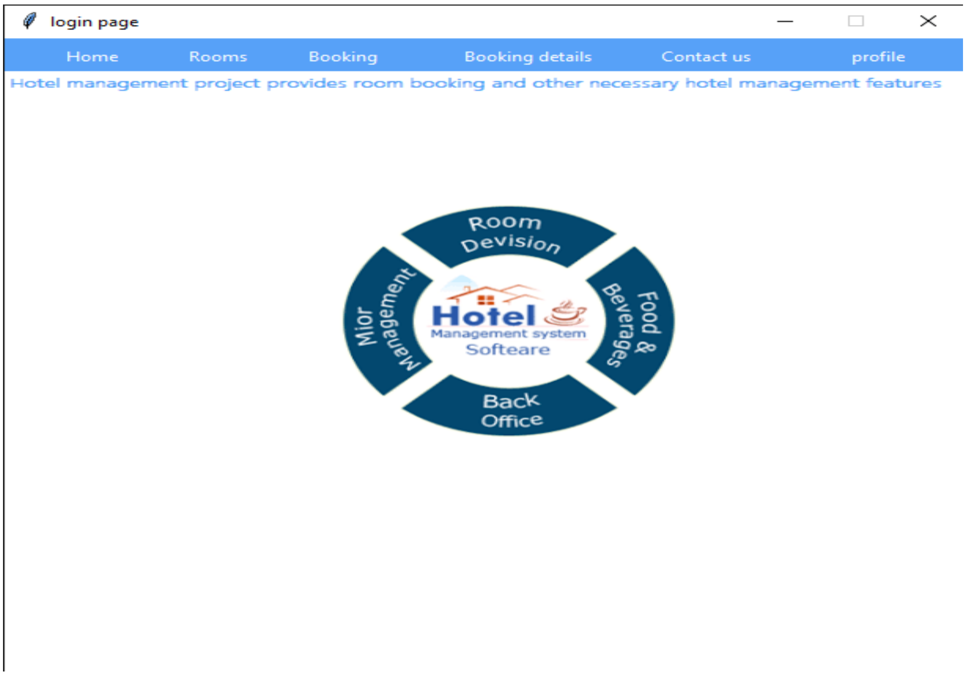
ID Number

Password ☐ (S)

Confirm Password ☐ (S)

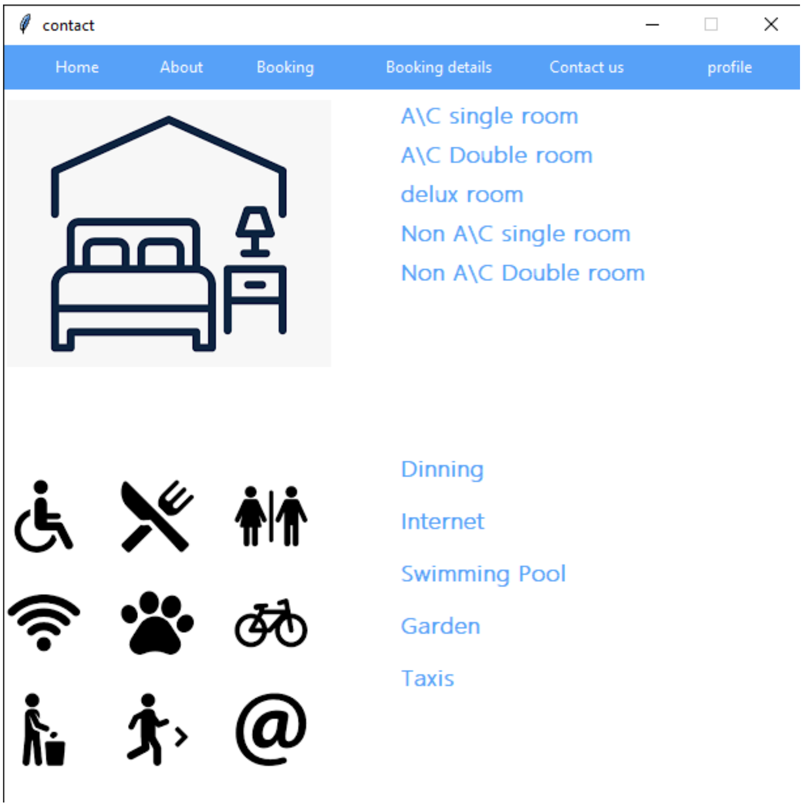
Already have as Account? Sign In

HOME PAGE



Hotel management project provides room booking and other necessary hotel management features

FACILITIES



A\C single room

A\C Double room

delux room

Non A\C single room

Non A\C Double room

Dinning

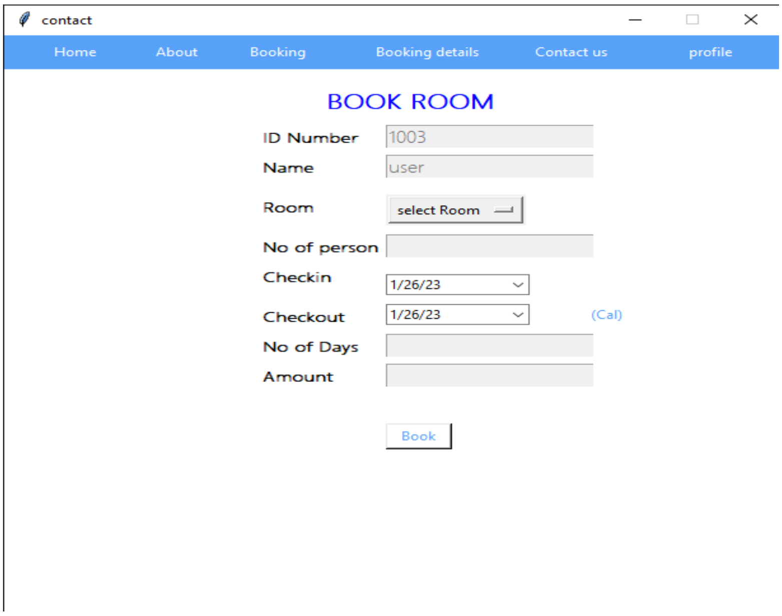
Internet

Swimming Pool

Garden

Taxis

BOOKING



BOOK ROOM

ID Number 1003

Name user

Room select Room

No of person

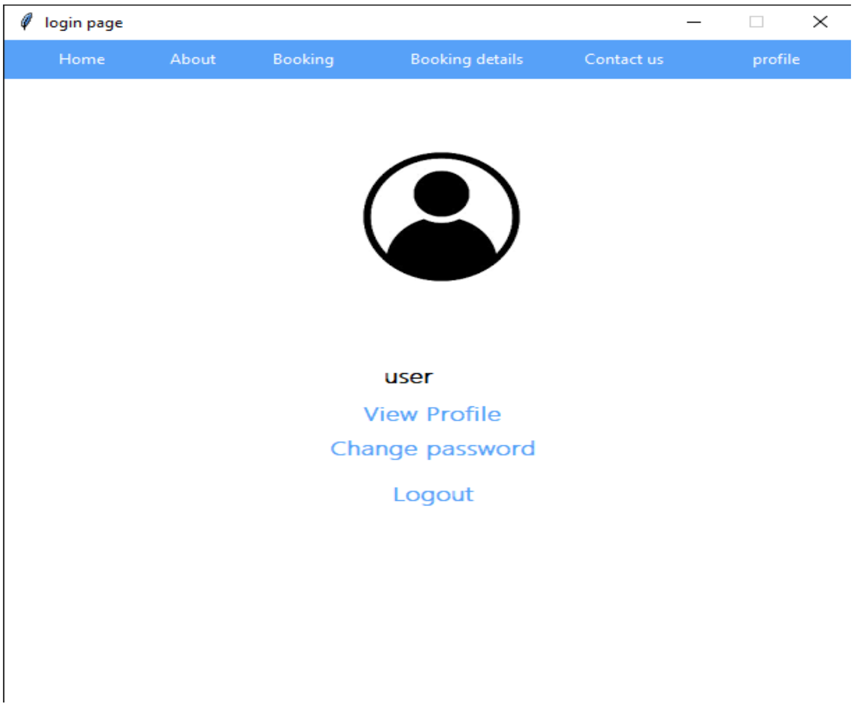
Checkin 1/26/23

Checkout 1/26/23 (Cal)

No of Days

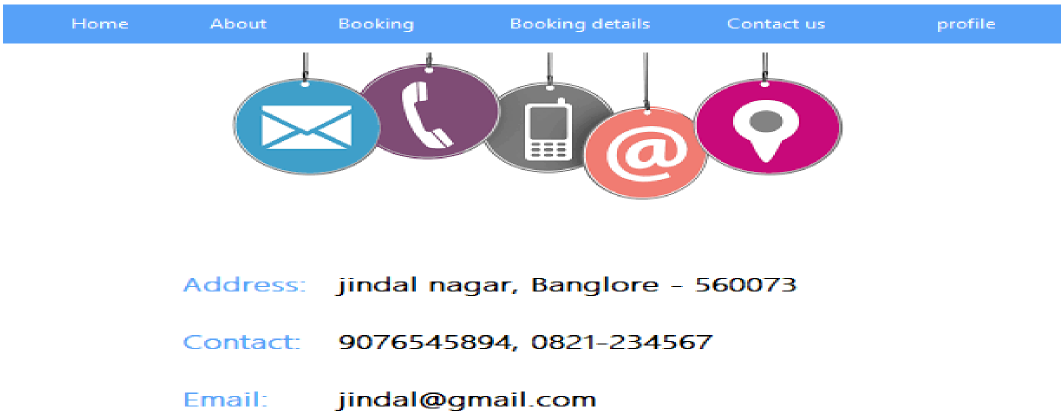
Amount

PROFILE PAGE



user

CONTACT US



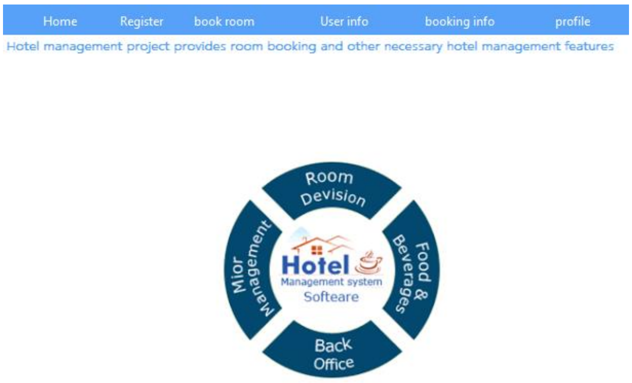
Address: jindal nagar, Banglore - 560073

Contact: 9076545894, 0821-234567

Email: jindal@gmail.com

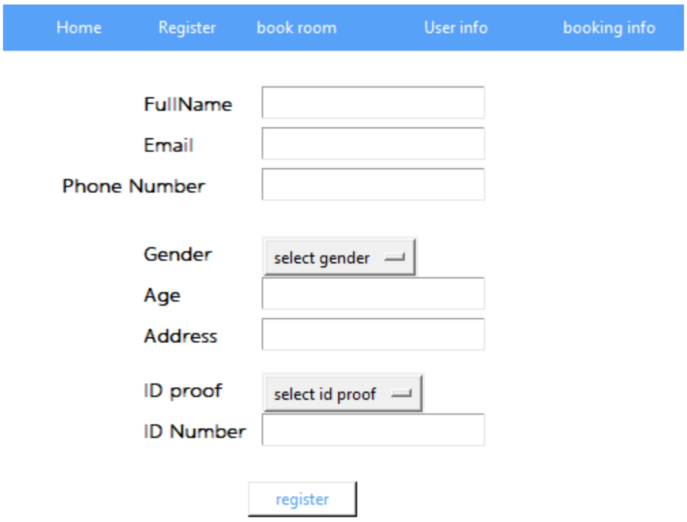
ADMIN PAGE

HOME PAGE

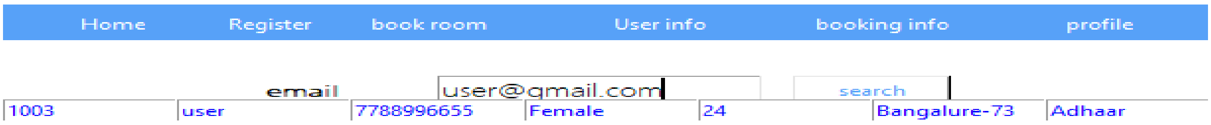


Hotel management project provides room booking and other necessary hotel management features

REGESTER PAGE



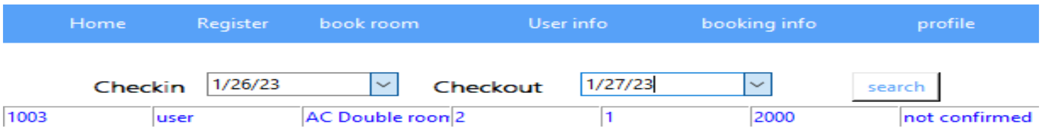
USER INFO



email user@gmail.com

1003	user	7788996655	Female	24	Bangalure-73	Adhaar

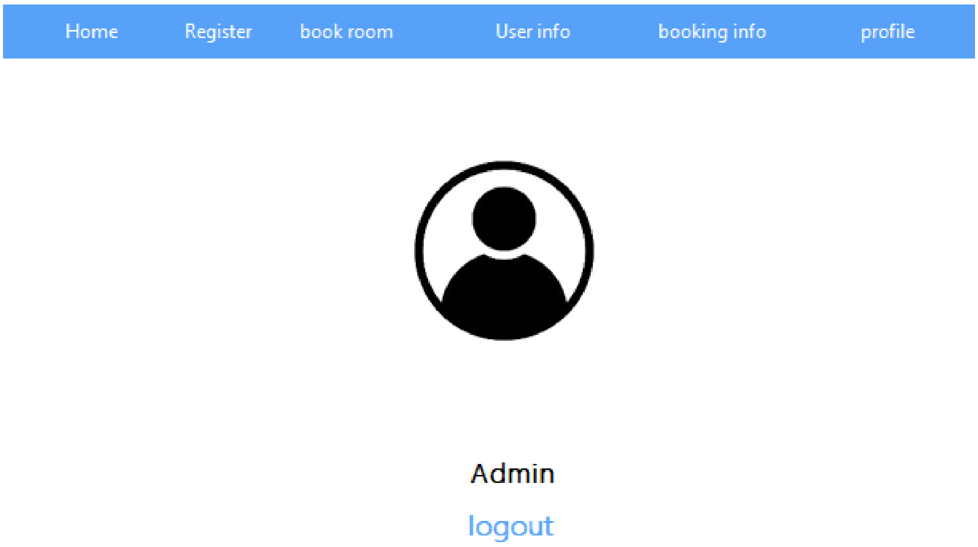
BOOKING INFO



Checkin 1/26/23 Checkout 1/27/23

1003	user	AC Double roon	2	1	2000	not confirmed

PROFILE



Admin

VALIDATION

6.VALIDATION

Data validation is a very important part in any application involving the database. We have to be aware of the specifications of the database organization and also but the kind of data that goes into the database. We have to provide for stringent error checking in order to discard invalid data and prevent it from getting into the database.

Data validation can be done into two places. While creating the database scheme we can specify certain integrity constraints that have to be maintained between various relations in the schema. This ensures that no insertion, deletion and update anomalies creep in during operation. Data validation can also be done at the interface itself to provide more robust level of data validation.

Eg: In the login from whose login Id has been established by the administration will be given login permission. This can be trapped at the interface level itself and in the form, to modify the user profile the administrator is allowed to modify only if that login password had got permission.

CONCLUSION

7. CONCLUSION

The Online Hotel Management System was developed to replace the manual process of booking for a hotel room or any other facility of the hotel. The old system does not serve the customer in a better way rather it makes customer data vulnerable. The new system keeps proper records of customers for emergency and security purposes. The hotel's advertising effort is now accompanied by a virtual tour created on the system.

FUTURE ENHANCEMEN

8. FUTURE ENHANCEMENT

The future enhancement of this project is we can build so much by accomplished using data, not only in delivering precisely what the guest want and requires – but also enhancing the stay. Within the booking flow, complementary products and services can be offered based on other preferences like the type of room the guest likes, whether they prefer a high floor or low floor, whether they consistently request late check-out and so on.

We can also enhance our project by also including

- Staff management
- Notification via mail or message for customer

BIBLIOGRAPHY

9. BIBLIOGRAPHY

Website:

- www.google.com
- www.youtube.com/
- www.wikipedia.com

Blogs:

- https://ukdiss.com/examples/hotel-room-booking-system-project.php
- https://projectworlds.in

Books:

- Software Engineering
  –Steve McConnell
- System Analysis and Design
  –James A Sen.