

Project Report
ON
“Hotel Management System”
Submitted to



Department of CSE
Sitamarhi Institute of Technology
Aryabhatta Knowledge University, Patna

Submitted by

Gautam Kumar Jha	16105127023
Prashant Kumar	16105127027
Pankaj Kumar	16105127015
Deepak Kumar Ram	16105127028

Under the Guidance of

Mr. Neeraj Kumar

Assistant Professor
Department of CSE
SIT, Sitamarhi

Declaration

I hereby declare that, the project being submitted by us entitled “**HOTEL MANAGEMENT SYSTEM**” for the partial fulfilment for the award of the “**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE & ENGINEERING**” is an authenticated record of work carried by us in the fourth year of B.Tech submitted to *Department of CSE, Sitamarhi Institute of Technology, Sitamarhi*, which is AFFILIATED to AICTE under the guidance of Prof. Neeraj kumar.

*Gautam Kumar Jha
Prashant Kumar
Pankaj Kumar
Deepak Kumar Ram*

Certificate

This is to certify that the project work entitled “HOTEL MANAGEMENT SYSTEM” which is being submitted by Gautam Kumar Jha, Prashant Kumar, Pankaj Kumar and Deepak Kumar Ram in partial fulfilment of the requirements for the 7th Semester Sessional Examination of Bachelor of Technology in Computer Science & Engineering during the academic year 2016-20. They have worked under the guidance of Prof. Neeraj Kumar (Asst. Prof. Department of CSE). This work is submitted to the department as a part of evaluation of 7th Semester Project.

.....
Prof. Neeraj Kumar

Project Guide

.....
Prof. Sadique Nayeem

H.O.D.
Department of CSE

.....
External Examiner

PREFACE

Programming languages, paradigms and practices don't stand still very long. It often seems that the methods and techniques we applied yesterday are out of date today of course this rapid rate of change is also one of the things that keep programming existing.

There is always something new on the horizon. One characteristic that is constant in software industry today is the "change". Change is one of the most critical aspects of s/w development and management. New tools and new approaches are announced almost every day. The impact of these developments is often very extensive. Most important among them is maintaining ability, reusability, portability, security, and integrity and user friendliness. To build today's complex s/w we need to wound construction techniques and program structures that are easy to comprehend, implement and modify in wide variety of situations.

Acknowledgement

First of all, We would like to thank the almighty God for listening our prayer and giving us strength to complete the project work.

We would like to express a deep sense of gratitude and thanks profusely to Mr. Neeraj Kumar, Assistant Professor, Department of CSE, Sitamarhi Institute of Technology, Sitamarhi (Bihar), who guided us throughout the project. Without his willing disposition, spirit of accommodation, frankness, timely clarification and above of all faith in us, this project could not have been completed in due time.

We would also like to thank whole of the faculty of the college for their cooperation and important support.

We must also express our deep regards and thanks to our parents for supporting and boosting our morale.

We finally pray that Almighty fulfils the aspirations of all of the who have been a part of this journey and those who will be a part of future journeys.

CONTENTS

<u>Sl. No.</u>	<u>Chapter Name</u>	<u>Page No.</u>
1.	Declaration	<i>i</i>
2.	Certificate	<i>ii</i>
3.	Preface	<i>iii</i>
4.	Acknowledgement	<i>iv</i>
5.	Contents	<i>v</i>
6.	An Introduction	1
7.	Objective	2
8.	The technologies used to develop this application are	3 - 5
9.	Data Flow Diagram	6 - 9
10.	Project Layout	10 - 13
11.	Benefits	14
12.	Sample Code	15 - 29
13.	Future Enhancement	30
14.	Bibliography	31
15.	Conclusion	32

Introduction

Hotel Management System is being launched because hotels need to make their record hassle free, which saves the time of both hotel and customers. Using this console based software hotels can manage their records. As there are many software in market for the purpose of hotel management, this software is an attempt to understand the working of the software.

Purpose

Hotel Management System fulfils the requirements of the hotels to manage all real-time records. They do not have to make records in traditional way in register manually. Manager or receptionist can maintain daily updates in the hotel records.

The main aim of the entire activity is to automate the process of day-to-day activities of Hotel like Room activities, Admission of a New Customer, Assign a room according to customer's demand, checkout of a computer and releasing the room, etc.

The limited time and resources have restricted us to incorporate, in this project, only a main activities that are performed in a HOTEL Management System, but utmost care has been taken to make the system efficient and user friendly.

"HOTEL Management System" has been designed to computerized the following functions that are performed by the system:

- Room Detail Functions
- Opening a New Room
- Check-in and check-out Detail Functions
- Admission of New customer
- Check-out of customer
- Statement of Customer Details
- Check-in customer
- Check-out customer
- Room Details
- Total number of Customers in the Hotel
- Individual customer Report



OBJECTIVE

During the past several decades personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources.

- A computer based management system is designed to handle all the primary information required to calculate monthly statements. Separate database is maintained to handle all the details required for the correct statement calculation and generation.
- This project intends to introduce more user friendliness in the various activities such as record updation, maintenance, and searching.
- The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification of that customer.
- Similarly, record maintenance and updation can also be accomplished by using the identification of the customer with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.
- The entire information has maintained in the database or Files and whoever wants to retrieve can't retrieve, only authorization user can retrieve the necessary information which can be easily be accessible from the file.



The technologies used to develop this Application are:-

Languages:-

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system capable of collecting reference cycles. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3.

The biggest strength of the Python is large library which can be used for the following:-

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like OpenCV, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia
- Scientific computing
- Text processing and many more..



Features of Python:-

1) Easy to Learn and Use

Python is easy to learn and use. It is developer-friendly and high level programming language.

2) Expressive Language

Python language is more expressive means that it is more understandable and readable.

3) Interpreted Language

Python is an interpreted language i.e. interpreter executes the code line by line at a time. This makes debugging easy and thus suitable for beginners.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that Python is a portable language.

5) Free and Open Source

Python language is freely available at official web address. The source-code is also available. Therefore it is open source.

6) Object-Oriented Language

Python supports object oriented language and concepts of classes and objects come into existence.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our python code.

8) Large Standard Library

Python has a large and broad library and provides rich set of module and functions for rapid application development.

9) GUI Programming Support

Graphical user interfaces can be developed using Python.

10) Integrated

It can be easily integrated with languages like C, C++, JAVA etc.



Softwares:-

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as Data Science with Anaconda.

PyCharm is cross-platform, with Windows, macOS and Linux versions. The Community Edition is released under the Apache License,[8] and there is also Professional Edition with extra features – released under a proprietary license.

PyCharm provides API so that developers can write their own plugins to extend PyCharm features. Several plugins from other JetBrains IDE also work with PyCharm. There are more than 1000 plugins which are compatible with PyCharm.

Features:-

- Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes
- Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages
- Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others
- Support for web frameworks: Django, web2py and Flask
- Integrated Python debugger
- Integrated unit testing, with line-by-line code coverage
- Google App Engine Python development
- Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge



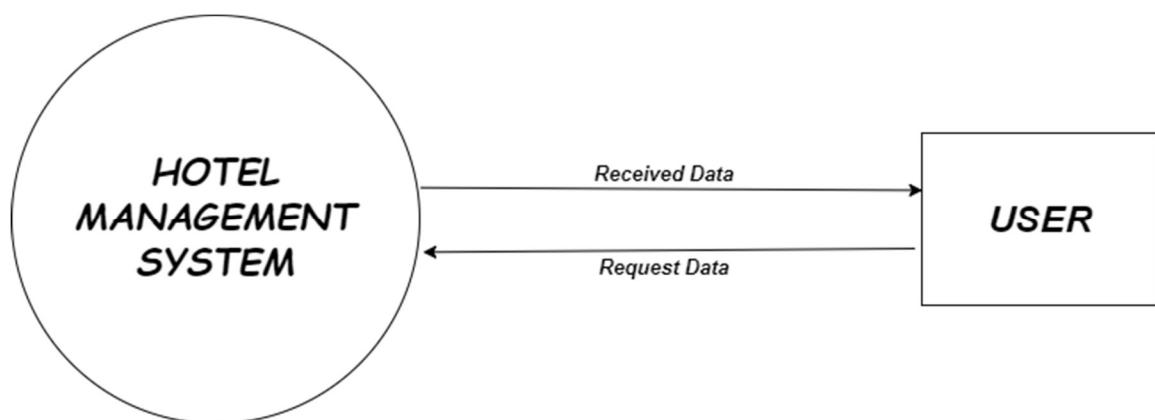
DATA FLOW DIAGRAM:

Data flow diagram is a graphic tool. It is used to describe and analyze the movement of data through a system manual or computerize.

They focus on the data flowing into the system, between processes in and out of the data stores.

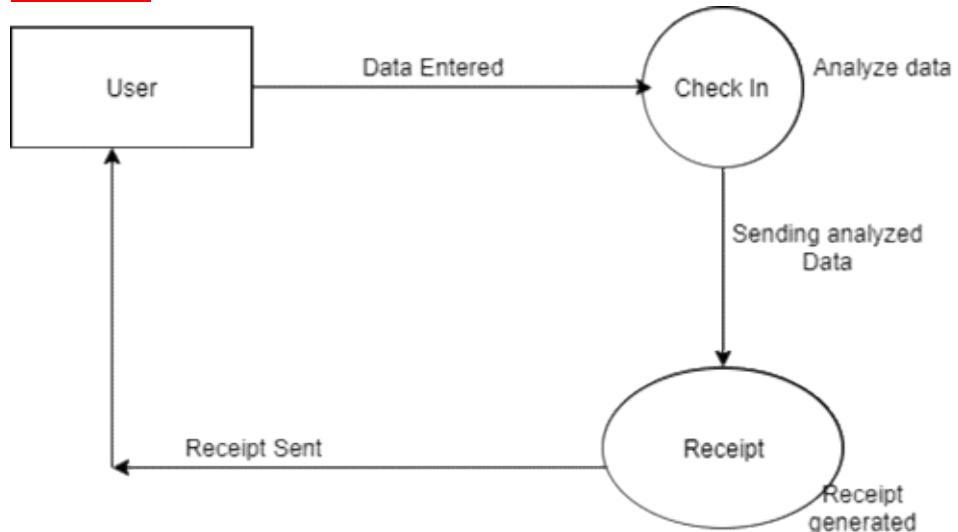
This diagram represents a online examination system, which maintains user name. In this example user can login and gives the examination.

Level 0:-



Level 1:-

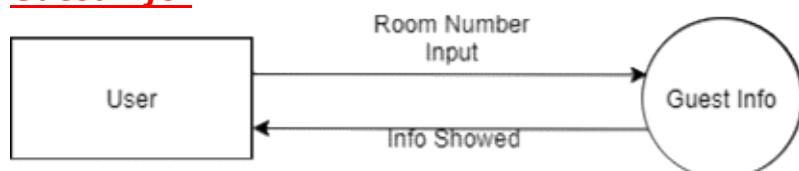
CheckIn:



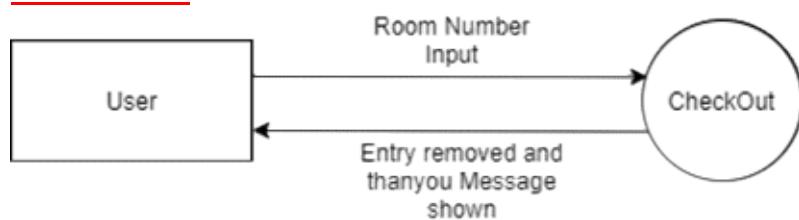
Guest List:



Guest Info:

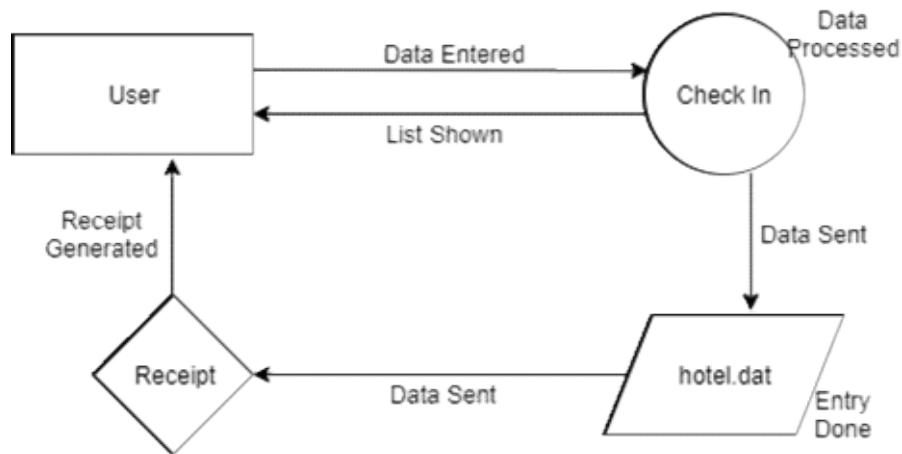


Check Out:

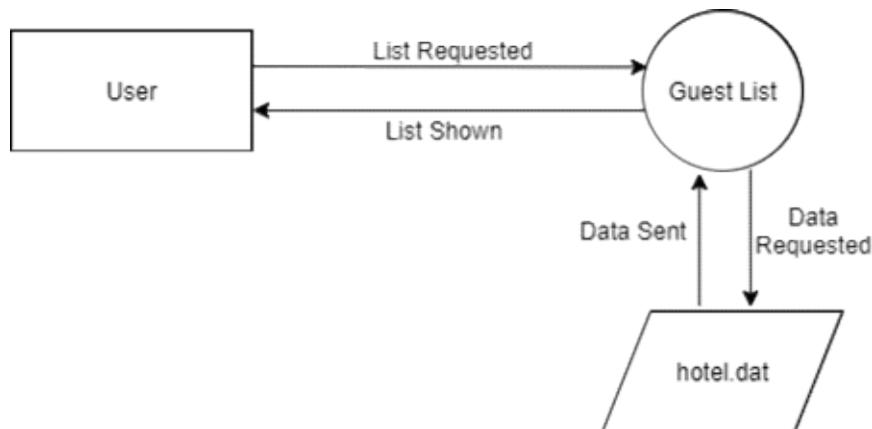


Level 2:-

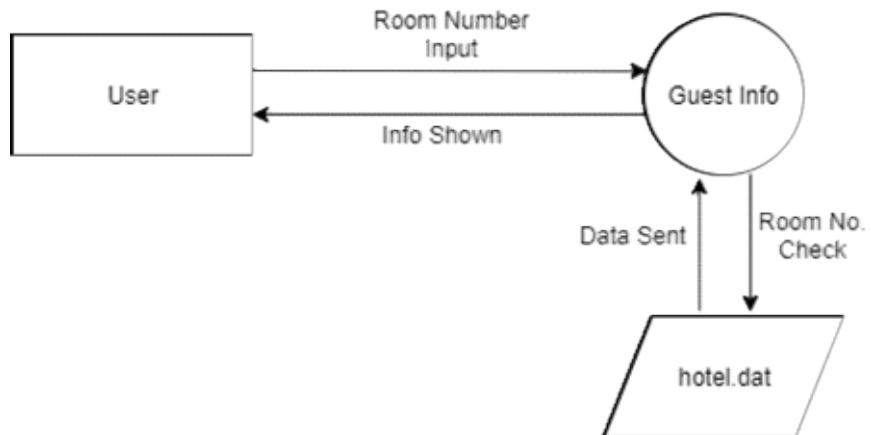
CheckIn:



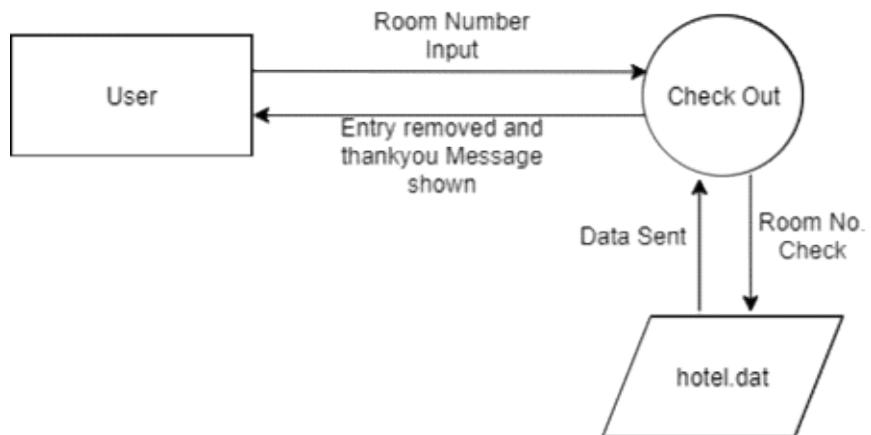
Guest List:



Guest Info:



Check Out:

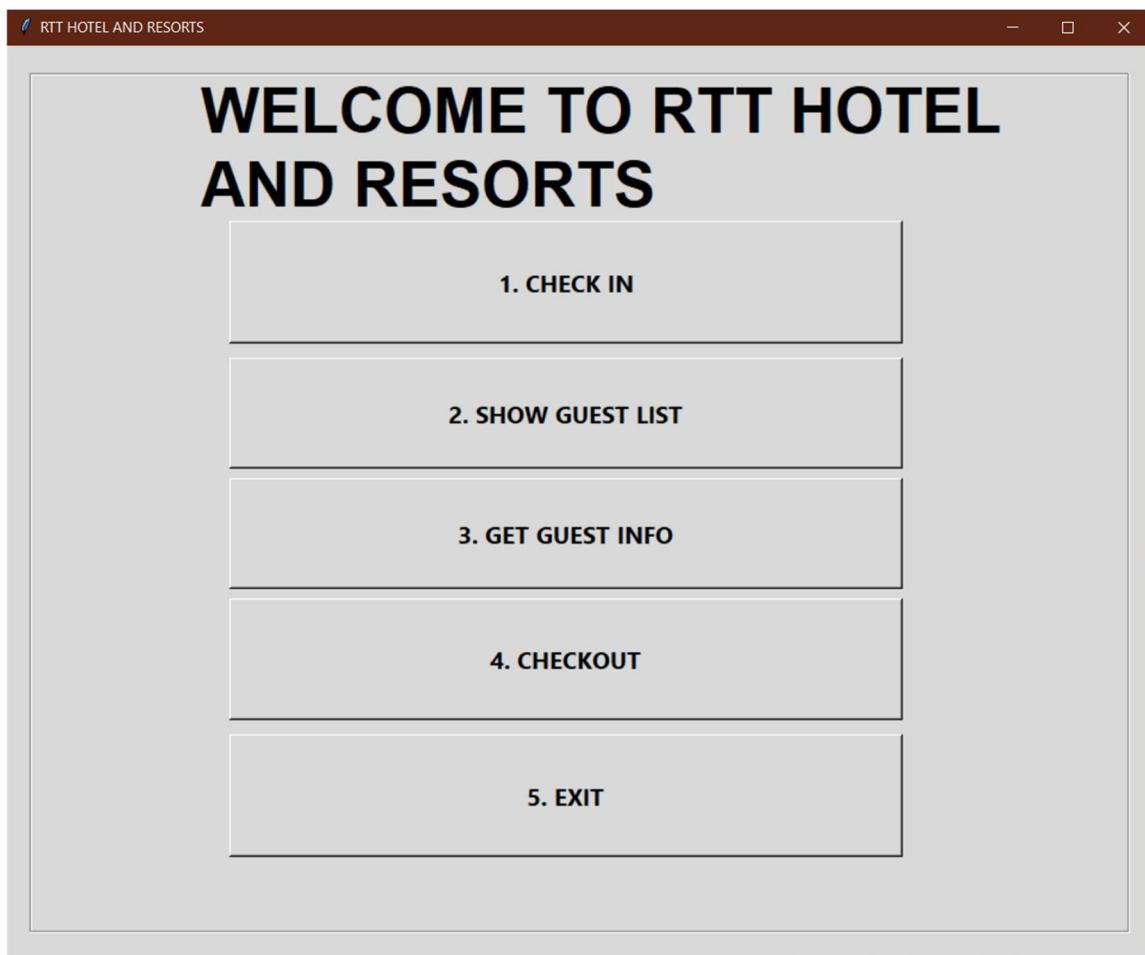


PROJECT LAYOUT

Modular Description:-

- Main/Home UI
- Check In
- Receipt Generation
- View Guest list
- Checkout
- Guest Information

Home Page



Check In Page

RTT HOTEL AND RESORTS : CHECK IN

YOU CLICKED ON : CHECK IN

ENTER YOUR NAME : OK

ENTER YOUR ADDRESS : OK

ENTER YOUR NUMBER : OK

NUMBER OF DAYS : OK

CHOOSE YOUR ROOM

DELUXE GENERAL

SEMI DELUXE JOINT

CHOOSE PAYMENT METHOD

By cash By credit/debit card

SUBMIT

Receipt Generated

RECEIPT

RTT HOTEL AND RESORTS SITAMARHI SERVING GUEST SINCE 2016

NAME-Prashant Kumar
ADDRESS-Muzaffarpur
MOBILE NO.-9471907240
YOUR TOTAL BILL IS Rs.-4000
YOUR ROOM NUMBER IS 2



Guest List Page

LIST OF ALL GUEST	
NAMES	ROOM NO.
PRASJADFGFG PRASHANT KUMAR	1
	2

Guest Info Page

RTT HOTEL AND RESORTS : GUEST INFO

GET INFO HERE ..!!

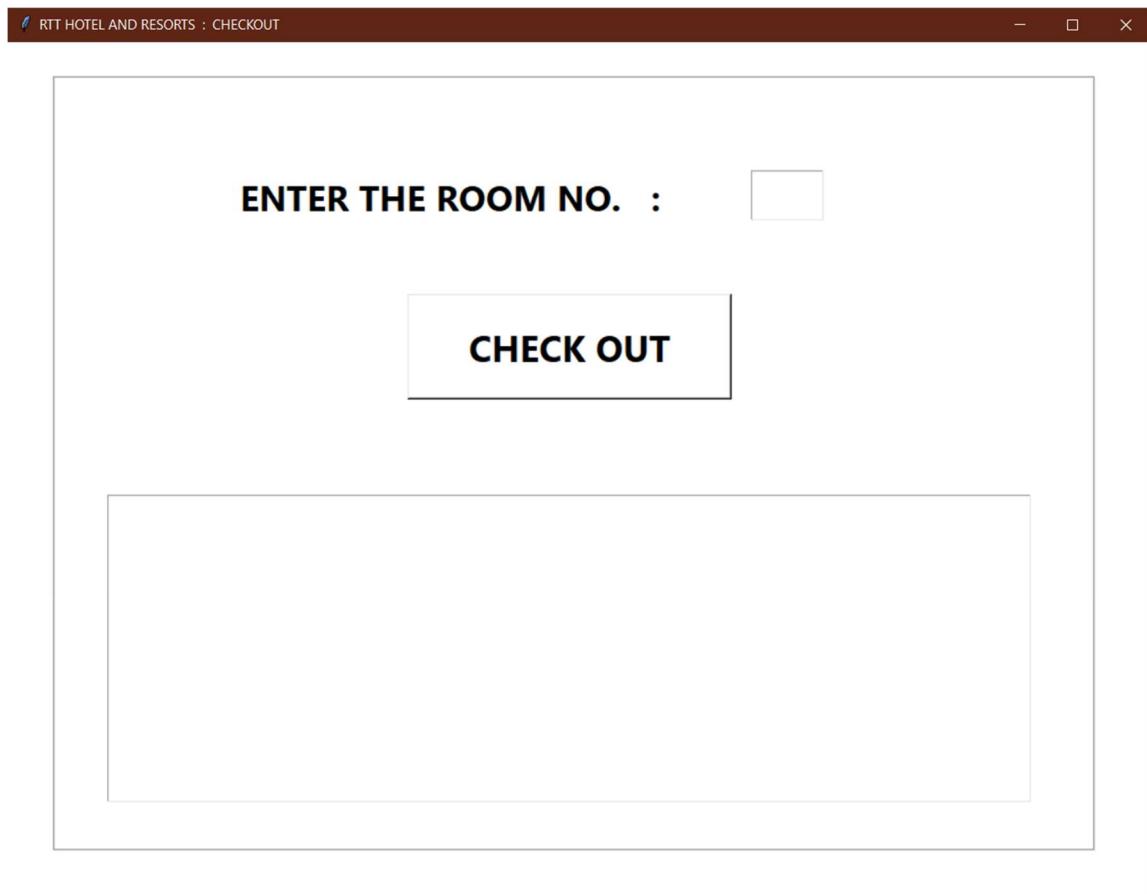
ENTER THE ROOM NO. :

SUBMIT

Valid room number

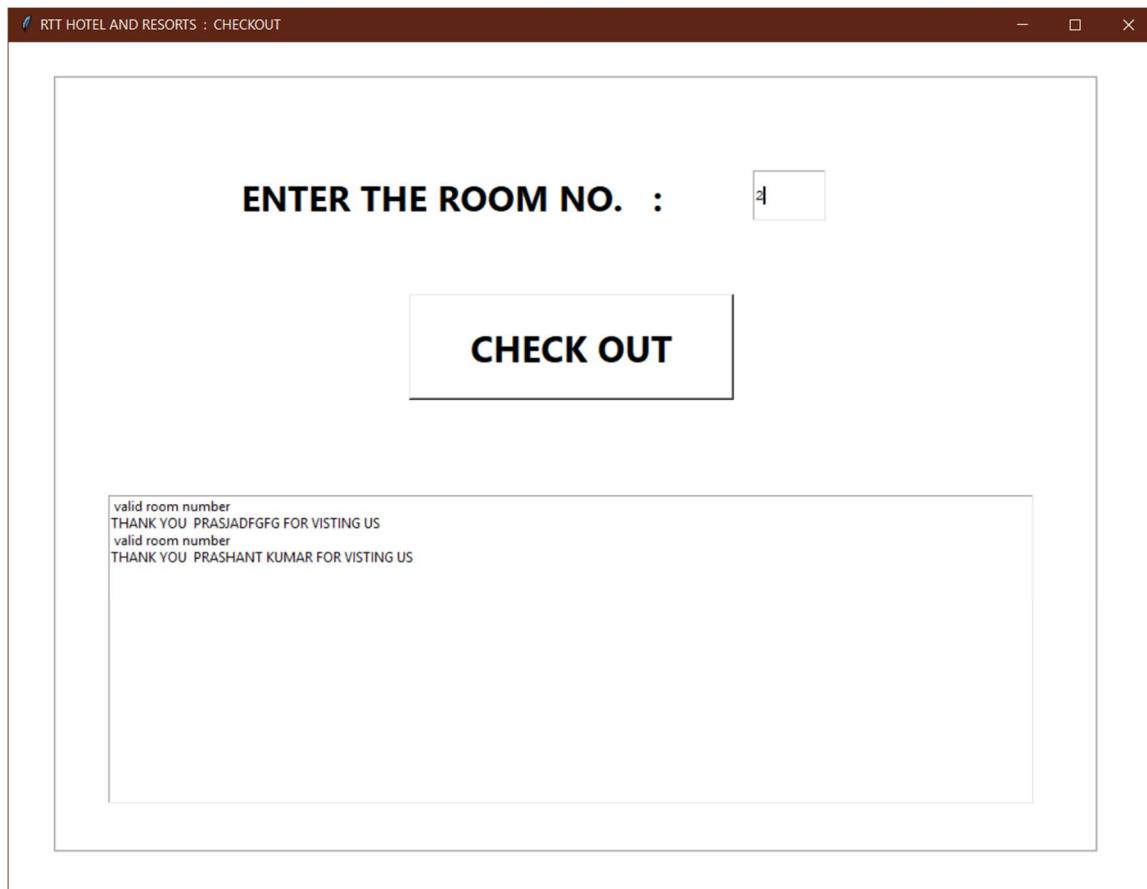


Checkout Page



A screenshot of a Windows application window titled "RTT HOTEL AND RESORTS : CHECKOUT". The main interface is white with black text. At the top center, it says "ENTER THE ROOM NO. :". To its right is a small input field containing a single digit. Below this is a large rectangular area for entering room numbers. In the center of this area is a button labeled "CHECK OUT".

Checkout done Page



A screenshot of the same Windows application window after a room number has been entered. The "ENTER THE ROOM NO." label now has a value of "4" in its input field. The "CHECK OUT" button remains below the input field. In the bottom-left corner of the main input area, there is a message box containing the following text:

```
valid room number
THANK YOU PRASJADFGFG FOR VISTING US
valid room number
THANK YOU PRASHANT KUMAR FOR VISTING US
```



BENEFITS

The following are the benefits of using the application:

User Convenience:

1. User can do all the works in a click.
2. Every work become very easy compared to traditional way

Analysis:

1. User can make analysis of its earnings and guests

Go Green:

1. It saves papers in traditional way hotel have to manage records in registers.



SAMPLE CODE

Home Page:-

```
import os
from subprocess import call

import sys

try:
    from Tkinter import *
except ImportError:
    from tkinter import *

try:
    import ttk
    py3 = False
except ImportError:
    import tkinter.ttk as ttk
    py3 = True
def click_checkinn():
    call(["python", "checkin_gui_and_program.py"])
def click_list():
    call(["python", "listgui.py"])
def click_checkout():
    call(["python", "checkoutgui.py"])
def click_getinfo():
    call(["python", "getinfoui.py"])

class HOTEL_MANAGEMENT:
    def __init__(self):
        root = Tk()
        '''This class configures and populates the toplevel
window.'''
        top = self
        _bgcolor = '#d9d9d9' # X11 color: 'gray85'
        _fgcolor = '#000000' # X11 color: 'black'
        _compcolor = '#ffffff' # X11 color: 'white'
        _ana1color = '#ffffff' # X11 color: 'white'
        _ana2color = '#ffffff' # X11 color: 'white'
        font14 = "-family {Segoe UI} -size 15 -weight bold -slant
"
        \ "roman -underline 0 -overstrike 0"
        font16 = "-family {Swis721 BlkCn BT} -size 40 -weight
bold "
        \ "-slant roman -underline 0 -overstrike 0"
        font9 = "-family {Segoe UI} -size 9 -weight normal -slant
"
        \
```



```

    "roman -underline 0 -overstrike 0"

root.geometry("963x749+540+110")
root.title("RTT HOTEL AND RESORTS")
root.configure(background="#d9d9d9")
root.configure(highlightbackground="#d9d9d9")
root.configure(highlightcolor="black")

self.menubar =
Menu(root,font=font9,bg=_bgcolor,fg=_fgcolor)
root.configure(menu = self.menubar)

self.Frame1 = Frame(root)
self.Frame1.place(relx=0.02, rely=0.03, relheight=0.94,
relwidth=0.96)
self.Frame1.configure(relief=GROOVE)
self.Frame1.configure(borderwidth="2")
self.Frame1.configure(relief=GROOVE)
self.Frame1.configure(background="#d9d9d9")
self.Frame1.configure(highlightbackground="#d9d9d9")
self.Frame1.configure(highlightcolor="black")
self.Frame1.configure(width=925)

self.Message6 = Message(self.Frame1)
self.Message6.place(relx=0.09, rely=0.01, relheight=0.15,
relwidth=0.86)
self.Message6.configure(background="#d9d9d9")
self.Message6.configure(font=font16)
self.Message6.configure(foreground="#000000")
self.Message6.configure(highlightbackground="#d9d9d9")
self.Message6.configure(highlightcolor="black")
self.Message6.configure(text='''WELCOME TO RTT HOTEL AND
RESORTS'''')
self.Message6.configure(width=791)

self.Button2 = Button(self.Frame1)
self.Button2.place(relx=0.18, rely=0.17, height=103,
width=566)
self.Button2.configure(activebackground="#d9d9d9")
self.Button2.configure(activeforeground="#000000")
self.Button2.configure(background="#d9d9d9")
self.Button2.configure(disabledforeground="#bfbfbf")
self.Button2.configure(font=font14)
self.Button2.configure(foreground="#000000")
self.Button2.configure(highlightbackground="#d9d9d9")

```



```

        self.Button2.configure(highlightcolor="black")
        self.Button2.configure(pady="0")
        self.Button2.configure(text='''1. CHECK IN''')
        self.Button2.configure(width=566)
        self.Button2.configure(command=click_checkinn)

        self.Button6 = Button(self.Frame1)
        self.Button6.place(relx=0.18, rely=0.77, height=103,
width=566)
        self.Button6.configure(activebackground="#d9d9d9")
        self.Button6.configure(activeforeground="#000000")
        self.Button6.configure(background="#d9d9d9")
        self.Button6.configure(disabledforeground="#bfbfbf")
        self.Button6.configure(font=font14)
        self.Button6.configure(foreground="#000000")
        self.Button6.configure(highlightbackground="#d9d9d9")
        self.Button6.configure(highlightcolor="black")
        self.Button6.configure(pady="0")
        self.Button6.configure(text='''5. EXIT''')
        self.Button6.configure(width=566)
        self.Button6.configure(command=quit)
root.mainloop()

if __name__ == '__main__':
    GUUEST=HOTEL_MANAGEMENT()

```

Check In:-

```

import os
import pickle
import sys
import os
from subprocess import call

import sys

try:
    from Tkinter import *
except ImportError:
    from tkinter import *

try:
    import ttk
    py3 = False
except ImportError:
    import tkinter.ttk as ttk
    py3 = True
details_list=[]

```



```

def file_save():
    NAME_PRO = details_list[0]
    ADDRESS_PRO = details_list[1]
    MOBILE_NO_PRO = details_list[2]
    ROOM_NO_PRO = details_list[3]
    PRICE_PRO = details_list[4]
    f = open("hotel.dat", "ab")

a=save(NAME_PRO,ADDRESS_PRO,MOBILE_NO_PRO,ROOM_NO_PRO,PRICE_PRO)
pickle.dump(a,f,protocol=2)
f.close()

listq=[str(NAME_PRO),str(ADDRESS_PRO),str(MOBILE_NO_PRO),str(ROOM
_NO_PRO),str(PRICE_PRO)]
myVars =
{'A':NAME_PRO,"B":ADDRESS_PRO,"C":MOBILE_NO_PRO,"D":ROOM_NO_PRO,"
E":PRICE_PRO }

fo=open("receipt.txt","w+")
for h in range(0,5):
    fo.write(listq[h]+\r\n")
fo.close()
call(["python", "receipt.py"])
restart_program()

u = list()
Delux = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
Semi_Delux = (11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
24, 25)
General = (26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38,
39, 40, 41, 42, 43, 44, 45)
Joint_Room = (46, 47, 48, 49, 50, 46, 47, 48, 49, 50)
m = [9]
G = []
def restart_program():
    """Restarts the current program.
    Note: this function does not return. Any cleanup action (like
    saving data) must be done before calling this function."""
    python = sys.executable
    os.execl(python, python, * sys.argv)

    ak = self.g.isdigit()

```



```

        if len(self.g)!= 0 and ak!=True:
            self.ADDERESS=self.g
            self.Text1.insert(INSERT, "address has been
inputed""\n")
            break
        else:
            self.Text1.insert(INSERT, "invalid input
please input a valid address""\n")

            break

    def chk_mo():
        while True:

            self.h = str(self.mobile.get())
            if self.h.isdigit() == True and len(self.h) != 0
and len(self.h) == 10:
                self.MOBILE = self.h
                self.Text1.insert(INSERT, "mobile number has
been inputed""\n")
                break
            else:
                self.Text1.insert(INSERT, "invalid input
please input a valid mobile number""\n")
                break

    def chk_day():
        while True:

            self.l = str(self.days.get())

            if self.l.isdigit() == True and len(self.l) != 0:
                self.DAYS = int(self.l)
                self.Text1.insert(INSERT, "days has been
inputed""\n")
                break
            else:
                self.Text1.insert(INSERT, "invalid input
""\n")
                break

    def enter(self):
        self.name = self.NAME
        self.address = self.ADDERESS
        self.mobile_no = self.MOBILE
        self.no_of_days = int(self.DAYS)

    def tor(self):
        self.Button3 = Button(self.Frame2)

```



```

        self.Button3.place(relx=0.91, rely=0.31, height=33,
width=43)
        self.Button3.configure(activebackground="#ffffff")
        self.Button3.configure(activeforeground="#000000")
        self.Button3.configure(background="#fffff")
        self.Button3.configure(disabledforeground="#bfbfbf")
        self.Button3.configure(foreground="#000000")
        self.Button3.configure(highlightbackground="#fffff")
        self.Button3.configure(highlightcolor="black")
        self.Button3.configure(pady="0")
        self.Button3.configure(text='''OK''')
        self.Button3.configure(command=chk_mo)

        self.Button4 = Button(self.Frame2)
        self.Button4.place(relx=0.76, rely=0.66, height=83,
width=156)
        self.Button4.configure(activebackground="#fffff")
        self.Button4.configure(activeforeground="#000000")
        self.Button4.configure(background="#fffff")
        self.Button4.configure(disabledforeground="#bfbfbf")
        self.Button4.configure(font=font16)
        self.Button4.configure(foreground="#000000")
        self.Button4.configure(highlightbackground="#fffff")
        self.Button4.configure(highlightcolor="black")
        self.Button4.configure(pady="0")
        self.Button4.configure(text='''SUBMIT''')
        self.Button4.configure(command=submit_clicked)

        self.Label1 = Label(self.Frame2)
        self.Label1.place(relx=0.05, rely=0.43, height=44,
width=260)
        self.Label1.configure(background="#fffff")
        self.Label1.configure(disabledforeground="#bfbfbf")
        self.Label1.configure(font=font13)
        self.Label1.configure(foreground="#000000")
        self.Label1.configure(text='''NUMBER OF DAYS''')

root.mainloop()

if __name__ == '__main__':
    hotel=HOTEL_MANGMENT_checkin()

```

List:-

```

class HOTEL_MANGMENT_checkin:
    def __init__(self):
        root = Tk()

```



```

'''This class configures and populates the toplevel
window.

    top is the toplevel containing window.'''
```

`_bgcolor = '#d9d9d9' # X11 color: 'gray85'`

`_fgcolor = '#000000' # X11 color: 'black'`

`_compcolor = '#ffffff' # X11 color: 'white'`

`_ana1color = '#ffffff' # X11 color: 'white'`

`_ana2color = '#ffffff' # X11 color: 'white'`

`font11 = "-family {Segoe UI} -size 17 -weight bold -slant`

`" \`

`"roman -underline 0 -overstrike 0"`

`font14 = "-family {Times New Roman} -size 16 -weight bold`

`" \`

`"-slant roman -underline 0 -overstrike 0"`

`root.geometry("780x541+504+123")`

`root.title("RTT HOTEL AND RESORTS : GUEST LIST")`

`root.configure(background="#ffffff")`

`root.configure(highlightbackground="#ffffff")`

`root.configure(highlightcolor="black")`

`self.Labelframe1 = LabelFrame(root)`

`self.Labelframe1.place(relx=0.01, rely=0.04,`

`relheight=0.95`

`, relwidth=0.97)`

`self.Labelframe1.configure(relief=GROOVE)`

`self.Labelframe1.configure(font=font11)`

`self.Labelframe1.configure(foreground="black")`

`self.Labelframe1.configure(text='''LIST OF ALL GUEST'''')`

`self.Labelframe1.configure(background="#ffffff")`

`self.Labelframe1.configure(highlightbackground="#ffffff")`

`self.Labelframe1.configure(highlightcolor="black")`

`self.Labelframe1.configure(width=760)`

`self.Frame1 = Frame(self.Labelframe1)`

`self.Frame1.place(relx=0.03, rely=0.1, relheight=0.86,`

`relwidth=0.47`

`, y=-31, h=15)`

`self.Frame1.configure(relief=GROOVE)`

`self.Frame1.configure(borderwidth="2")`

`self.Frame1.configure(relief=GROOVE)`

`self.Frame1.configure(background="#d9d9d9")`

`self.Frame1.configure(highlightbackground="#ffffff")`

`self.Frame1.configure(highlightcolor="black")`

`self.Frame1.configure(width=355)`

`self.Label1 = Label(self.Frame1)`



```

        self.Label1.place(relx=0.28, rely=0.02, height=37,
width=117)
        self.Label1.configure(activebackground="#ffffff")
        self.Label1.configure(activeforeground="black")
        self.Label1.configure(background="#d9d9d9")
        self.Label1.configure(disabledforeground="#a3a3a3")
        self.Label1.configure(font=font11)
        self.Label1.configure(foreground="#000000")
        self.Label1.configure(highlightbackground="#ffffff")
        self.Label1.configure(highlightcolor="black")
        self.Label1.configure(text='''NAMES''')


self.Text1 = Text(self.Frame1)
self.Text1.place(relx=0.06, rely=0.16, relheight=0.8,
relwidth=0.88)
self.Text1.configure(background="white")
self.Text1.configure(font=font14)
self.Text1.configure(foreground="#000000")
self.Text1.configure(highlightbackground="#d9d9d9")
self.Text1.configure(highlightcolor="black")
self.Text1.configure(insertbackground="black")
self.Text1.configure(selectbackground="#c4c4c4")
self.Text1.configure(selectforeground="black")
self.Text1.configure(width=314)
self.Text1.configure(wrap=WORD)

```

```
root.mainloop()
```

```

if __name__ == '__main__':
    f2 = open("hotel.dat", "rb")
    try:
        while True:
            s = pickle.load(f2)
            k = s.room_no
            o = s.name.upper()
            l2.append(o)

            G.append(k)
            continue

    except EOFError:
        pass
    f2.close()
    hotel=HOTEL_MANGMENT_checkin()

```



Receipt:-

```
class receipt:  
    def __init__(self):  
        root=Tk()  
        '''This class configures and populates the toplevel  
window.  
        top is the toplevel containing window.'''  
        _bgcolor = '#d9d9d9' # X11 color: 'gray85'  
        _fgcolor = '#000000' # X11 color: 'black'  
        _compcolor = '#d9d9d9' # X11 color: 'gray85'  
        _ana1color = '#d9d9d9' # X11 color: 'gray85'  
        _ana2color = '#d9d9d9' # X11 color: 'gray85'  
  
        root.geometry("800x800")  
        root.title("RECEIPT")  
        root.configure(background="#d9d9d9")  
  
  
        self.Label1 = Label(root)  
        self.Label1.configure(background="#d9d9d9")  
        self.Label1.place(relx=0, rely=0, height=800, width=800)  
        self.Label1.configure(disabledforeground="#a3a3a3")  
        self.Label1.configure(foreground="#000000")  
        self.Label1.configure(text=p)  
        self.Label1.configure(anchor=N)  
  
        self.Label1.configure(wraplength=1000)  
        self.Label1.configure(justify=LEFT)  
  
        self.Label1.configure(width=582)  
        root.mainloop()  
  
  
if __name__ == '__main__':  
    receipt1=receipt()
```

CheckOut:-

```
class New_Toplevel:  
  
    def __init__(self):  
        def check_room():  
            self.rom = str(self.data.get())  
            print(self.rom)  
            print("\n")  
            if self.rom.isdigit() == True and len(self.rom) != 0:
```



```

        self.Text1.insert(INSERT, " valid room number
"""\n")
        v = int(self.room)
        f = open("hotel.dat", "rb")
        f1 = open("hote.dat", "ab")
        n = 0
        try:
            while True:
                s = pickle.load(f)
                if s.room_no == v:
                    n = 1
                    name1 = s.name

                    print(" ")
                else:
                    pickle.dump(s, f1)
        except EOFError:
            if n == 0:
                self.Text1.insert(INSERT, "NO GUEST
FOUND"""\n")

        elif n == 1:

            self.Text1.insert(INSERT, "THANK YOU " +
name1.upper() + " FOR VIISTING US"""\n")
            pass
            f.close()
            f1.close()
            os.remove("hotel.dat")
            os.rename("hote.dat", "hotel.dat")

        else:
            self.Text1.insert(INSERT, "invalid input please
input a valid ROOM NO.""\n")

root = Tk()
'''This class configures and populates the toplevel
window.

    top is the toplevel containing window.'''
_bgcolor = '#ffffff' # X11 color: 'white'
_fgcolor = '#000000' # X11 color: 'black'
_compcolor = '#ffffff' # X11 color: 'white'
_ana1color = '#ffffff' # X11 color: 'white'
_ana2color = '#ffffff' # X11 color: 'white'
font10 = "-family {Courier New} -size 10 -weight normal -
slant"
\

    " roman -underline 0 -overstrike 0"
font11 = "-family {Segoe UI} -size 23 -weight bold -slant
" \

```



```

        "roman -underline 0 -overstrike 0"
font12 = "-family {Segoe UI} -size 24 -weight bold -slant
"
"\ "
        "roman -underline 0 -overstrike 0"
font9 = "-family {Segoe UI} -size 9 -weight normal -slant
"\ "
        "roman -underline 0 -overstrike 0"

root.geometry("1011x750")
root.title("RTT HOTEL AND RESORTS : CHECKOUT")
root.configure(background="#ffffff")
root.configure(highlightbackground="#ffffff")
root.configure(highlightcolor="black")

self.Frame1 = Frame(root)
self.Frame1.place(relx=0.04, rely=0.04, relheight=0.91,
relwidth=0.91)
self.Frame1.configure(relief=GROOVE)
self.Frame1.configure(borderwidth="2")
self.Frame1.configure(relief=GROOVE)
self.Frame1.configure(background="#ffffff")
self.Frame1.configure(highlightbackground="#ffffff")
self.Frame1.configure(highlightcolor="black")
self.Frame1.configure(width=925)

self.Label1 = Label(self.Frame1)
self.Label1.place(relx=0.14, rely=0.12, height=46,
width=442)
self.Label1.configure(activebackground="#ffffff")
self.Label1.configure(activeforeground="black")
self.Label1.configure(background="#ffffff")
self.Label1.configure(disabledforeground="#bfbfbf")
self.Label1.configure(font=font11)
self.Label1.configure(foreground="#000000")
self.Label1.configure(highlightbackground="#ffffff")
self.Label1.configure(highlightcolor="black")
self.Label1.configure(text='''ENTER THE ROOM NO. : ''')

self.Entry1 = Entry(self.Frame1)
self.data=StringVar()
self.Entry1.place(relx=0.67, rely=0.12, height=44,
relwidth=0.07)
self.Entry1.configure(background="white")
self.Entry1.configure(disabledforeground="#bfbfbf")
self.Entry1.configure(font=font10)
self.Entry1.configure(foreground="#000000")
self.Entry1.configure(highlightbackground="#ffffff")

```



```

        self.Entry1.configure(highlightcolor="black")
        self.Entry1.configure(insertbackground="black")
        self.Entry1.configure(selectbackground="#e6e6e6")
        self.Entry1.configure(selectforeground="black")
        self.Entry1.configure(textvariable=self.data)

self.Text1 = Text(self.Frame1)
self.Text1.place(relx=0.05, rely=0.54, relheight=0.4,
relwidth=0.89)
self.Text1.configure(background="white")
self.Text1.configure(font=font9)
self.Text1.configure(foreground="black")
self.Text1.configure(highlightbackground="#ffffff")
self.Text1.configure(highlightcolor="black")
self.Text1.configure(insertbackground="black")
self.Text1.configure(selectbackground="#e6e6e6")
self.Text1.configure(selectforeground="black")
self.Text1.configure(width=824)
self.Text1.configure(wrap=WORD)

self.Button1 = Button(self.Frame1)
self.Button1.place(relx=0.34, rely=0.28, height=93,
width=286)
self.Button1.configure(activebackground="#ffffff")
self.Button1.configure(activeforeground="#000000")
self.Button1.configure(background="#ffffff")
self.Button1.configure(disabledforeground="#bfbfbf")
self.Button1.configure(font=font12)
self.Button1.configure(foreground="#000000")
self.Button1.configure(highlightbackground="#ffffff")
self.Button1.configure(highlightcolor="black")
self.Button1.configure(pady="0")
self.Button1.configure(text='''CHECK OUT''')
self.Button1.configure(command=check_room)
root.mainloop()

if __name__ == '__main__':
    out=New_Toplevel()

```



Get Info:-

```
class HOTEL_MANAGEMENT:
    def __init__(self):
        def gotinfo():
            self.gettininfo = str(self.gather.get())
            print(self.gettininfo)
            print("\n")
            if self.gettininfo.isdigit() == True and
len(self.gettininfo) != 0:
                self.Text1.insert(INSERT, " valid room number
"""\n""")
            else :
                self.Text1.insert(INSERT,"invalid room
number"""\n")

        try:
            n = 0
            f2 = open("hotel.dat", "rb")
            while True:
                s = pickle.load(f2)
                a = str(s.room_no)
                print (a)
                if self.gettininfo == a:
                    n = 1
                    print("NAME-", "\t", "\t", s.name)
                    print("\n")
                    print("ADDRESS-", "\t", s.address)
                    print("\n")
                    print("MOBILE NO.-", " ", s.mobile_no)
                    print("\n")
                    print("HIS TOTAL BILL IS Rs.", s.price)
            elif EOFError:
                if n == 0:
                    print("NO GUEST IN ROOM ",
self.gettininfo)
                else:
                    n = 0
                    continue
            except EOFError:
                pass
            f2.close()

        root = Tk()
        '''This class configures and populates the toplevel
window.
        top is the toplevel containing window.'''
        _bgcolor = '#d9d9d9' # X11 color: 'gray85'
        _fgcolor = '#000000' # X11 color: 'black'
```



```

_compcolor = '#d9d9d9' # X11 color: 'gray85'
_ana1color = '#d9d9d9' # X11 color: 'gray85'
_ana2color = '#d9d9d9' # X11 color: 'gray85'
font10 = "-family {Segoe UI} -size 17 -weight bold -slant
"
  \
    "roman -underline 0 -overstrike 0"
font11 = "-family {Segoe UI} -size 28 -weight bold -slant
"
  \
    "roman -underline 0 -overstrike 0"
font9 = "-family {Segoe UI} -size 23 -weight bold -slant
\ "
  " -underline 0 -overstrike 0"

root.geometry("881x582+249+104")
root.title("RTT HOTEL AND RESORTS : GUEST INFO")
root.configure(background="#d9d9d9")

self.Frame1 = Frame(root)
self.Frame1.place(relx=0.02, rely=0.03, relheight=0.94,
relwidth=0.94)
self.Frame1.configure(relief=GROOVE)
self.Frame1.configure(borderwidth="2")
self.Frame1.configure(relief=GROOVE)
self.Frame1.configure(background="#d9d9d9")
self.Frame1.configure(width=825)

self.Text1 = Text(self.Frame1)
self.Text1.place(relx=0.04, rely=0.46, relheight=0.48,
relwidth=0.93)
self.Text1.configure(background="white")
self.Text1.configure(font="TkTextFont")
self.Text1.configure(foreground="black")
self.Text1.configure(highlightbackground="#d9d9d9")
self.Text1.configure(highlightcolor="black")
self.Text1.configure(insertbackground="black")
self.Text1.configure(selectbackground="#c4c4c4")
self.Text1.configure(selectforeground="black")
self.Text1.configure(width=764)
self.Text1.configure(wrap=WORD)

self.Label1 = Label(self.Frame1)
self.Label1.place(relx=0.12, rely=0.15, height=48,
width=377)
self.Label1.configure(background="#d9d9d9")
self.Label1.configure(disabledforeground="#a3a3a3")

```



```

        self.Label1.configure(font=font9)
        self.Label1.configure(foreground="#000000")
        self.Label1.configure(text=' ' 'ENTER THE ROOM NO. : ' ')

        self.Entry1 = Entry(self.Frame1)
        self.gather=StringVar()
        self.Entry1.place(relx=0.65, rely=0.17,height=40,
relwidth=0.1)
        self.Entry1.configure(background="white")
        self.Entry1.configure(disabledforeground="#a3a3a3")
        self.Entry1.configure(font="TkFixedFont")
        self.Entry1.configure(foreground="#000000")
        self.Entry1.configure(insertbackground="black")
        self.Entry1.configure(width=84)
        self.Entry1.configure(textvariable=self.gather)

        self.Button1 = Button(self.Frame1)
        self.Button1.place(relx=0.39, rely=0.29, height=74,
width=197)
        self.Button1.configure(activebackground="#d9d9d9")
        self.Button1.configure(activeforeground="#000000")
        self.Button1.configure(background="#d9d9d9")
        self.Button1.configure(disabledforeground="#a3a3a3")
        self.Button1.configure(font=font10)
        self.Button1.configure(foreground="#000000")
        self.Button1.configure(highlightbackground="#d9d9d9")
        self.Button1.configure(highlightcolor="black")
        self.Button1.configure(pady="0")
        self.Button1.configure(text=' ' 'SUBMIT' ')
        self.Button1.configure(width=197)
        self.Button1.configure(command=gotinfo)

        self.Message1 = Message(self.Frame1)
        self.Message1.place(relx=0.22, rely=0.02, relheight=0.12,
relwidth=0.56)
        self.Message1.configure(background="#d9d9d9")
        self.Message1.configure(font=font11)
        self.Message1.configure(foreground="#000000")
        self.Message1.configure(highlightbackground="#d9d9d9")
        self.Message1.configure(highlightcolor="black")
        self.Message1.configure(text=' ' 'GET INFO HERE ..!!' ')
        self.Message1.configure(width=460)
root.mainloop()

if __name__ == '__main__':
    GETINFO=HOTEL_MANAGEMENT()

```



FUTURE ENHANCEMENT

The future enhancement of this software, can link the software to online data base and provide a administrative control of this software through the internet. This will be helpful to the companies, which have a chain of hotels to manage and view the details of their hotels. Make the software a customizable software, so that companies can customize the software according to their requirements.



BIBLIOGRAPHY

- ❖ *Python.org*
- ❖ *JetBrains (PyCharm)*
- ❖ *Google*
- ❖ *Wikipedia*
- ❖ *Geeksforgeeks*
- ❖ *Tutorialspoint*



Conclusion

HOTEL MANAGEMENT SYSTEM has managed to successfully fulfils its objective of making easy of managing the records of any hotel. This application provides facility to automatically generating the receipt when any guest checked in. It saves time as the user (receptionist in hotel) have not to find manually which room is available for guests. It allows user to check how many guest are in hotels and are in which rooms. User can check details of any guest only with the room number.

