VISWAPRAKASH CENTRAL SCHOOL

(Affiliated to CBSE, New Delhi. Affiliation No.930625)
Mangattukadavu, Thirumala P.O, Thiruvananthapuram-695006

A Project Report On

HOTEL MANAGEMENT SYSTEM

Submitted in partial fulfillment for the requirement of
the award of AISSCE certified by the
Central Board of Secondary Education

NAME: THOMAS BIJU CHEERAMVELIL

CLASS: XII - A

Register No. ....

AISSCE PRACTICAL EXAMINATION

2021-2022

CERTIFICATE

This is to certify that the Project/Dissertation entitled **HOTEL MANAGEMENT SYSTEM** is a bonafide work done by THOMAS BIJU CHEERAMVELIL of XII A Science Session 2021-22 in partial fulfillment of CBSE's AISSCE Examination, 2022 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form a part of any other course undergone by the candidate. The student has done this project with his own effort and the guidance of the teacher.

……………………	…………………..	…………………
SIGNATURE OF	**SIGNATURE OF**	**SIGNATURE OF**
EXTERNAL EXAMINER	**TEACHER**	**PRINCIPAL**
	MRS. SOORYA NAIR M. S.	**MRS. SMITHA S.**

DECLARATION

I, **THOMAS BIJU CHEERAMVELIL,** do hereby declare that this project report on **Hotel Management System** is a record of study and work carried out by me and no part of this has been submitted by me for the award of any certificate or recognition before.

Thiruvananthapuram Signature:

Date THOMAS BIJU CHEERAMVELIL

Register No.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

Apart from my efforts, the success of any project depends largely on the encouragement and guidance of many others. I take this opportunity to express my gratitude to all who have been instrumental in the successful completion of this project.

I express my deep sense of gratitude to the almighty God for giving me strength for the successful completion of the project.

I express my heartfelt gratitude to my parents for their constant encouragement while carrying out this project.

I gratefully acknowledge the contribution of the individuals who contributed in bringing this project up to this level.

I express my deep sense of gratitude to **Mrs. Smitha S**, The Principal, Viswaprakash Central School, who has motivated and helped us throughout the course.

I express my thanks to our Manager **Rev.Fr.Gladin Alex** for providing the infrastructure and moral support to carry out this project in the school.

I express my sincere and heartfelt gratitude to **Mrs. Soorya Nair M.S**, teacher in-charge, my guide, mentor and above all a friend, who critically reviewed my project and helped in solving each and every problem that occurred during the implementation of the project.

The guidance and support received from all the members who contributed to this project was vital for the success of the project. I am grateful for their constant support and help.

PROJECT ON HOTEL MANAGEMENT SYSTEM

INTRODUCTION

This program helps users in selecting and booking rooms from the various types of rooms available, specifying the days for which it is required. The bill of this booking is based on the type of room they have selected. Then, the total bill amount, which also includes the restaurant bill, is displayed along with the customer’s name, check-in date, check-out date, and the room in which the user was staying.

OBJECTIVES OF THE PROJECT

The objective of this project is to let the students apply the programming knowledge into a real-world situation/problem and expose the students to how programming skills helps in developing a good software.

1. Write programs utilizing modern software tools.
2. Apply object-oriented programming principles effectively when developing small to medium sized projects.
3. Write effective procedural code to solve small to medium sized problems.
4. Students demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development. Students demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.

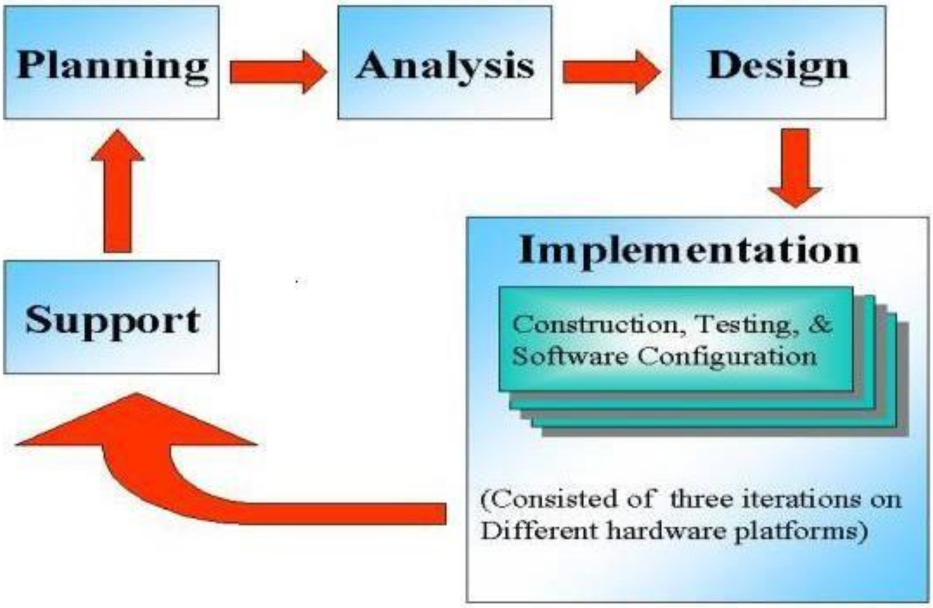
PROPOSED SYSTEM

The wise saying **"To err is human"** is no longer valid, it's out-dated to rationalize your mistake. One cannot afford to rely on fallible human beings if one really wants to withstand today's merciless competition. So, to keep pace with time, to bring about the best result without malfunctioning and for greater efficiency we can replace the heaps of files with a more sophisticated hard disk of the computer.

One has to use the data management software. Software has been an asset in automation of various organisations. Many software products available now in markets, have helped in making the organization's work easier and efficient. Data management initially had to maintain a lot of ledgers and a lot of paperwork. But now, software has helped this organization make their work faster and easier. Now, only this software has to be loaded on the computer and the work can be done.

This prevents a loss of time and money. Work becomes fully automated and any information regarding the organization can be obtained by clicking the button. Moreover, now it's an age of computers and automating such an organization gives it a better look.

SYSTEM DEVELOPMENT LIFE CYCLE (SDLC)


Planning
Analysis
Design
Implementation
Construction, Testing, & Software Configuration
(Consisted of three iterations on Different hardware platforms)
Support

The systems development life cycle is a project management technique that divides complex projects into smaller, more easily managed segments or phases. Segmenting projects allows managers to verify the successful completion of project phases before allocating resources to subsequent phases.

Software development projects typically include initiation, planning, design, development, testing, implementation, and maintenance phases. However, the phases may be divided differently depending on the organization involved.

For example, initial project activities might be designated as request, requirements definition, and planning phases, or initiation, concept development, and planning phases. End users of the system under development should be involved in reviewing the output of each phase to ensure the system is being built to deliver the needed functionality.

PHASES OF SYSTEM DEVELOPMENT LIFE CYCLE

INITIATION PHASE

The Initiation Phase begins when a business sponsor identifies a need or an opportunity.

The purpose of the Initiation Phase is to:

- Identify and validate an opportunity to improve business accomplishments of the organization or a deficiency related to a business need.
- Identify significant assumptions and constraints on solutions to that need.
- Recommend the exploration of alternative concepts and methods to satisfy the need including questioning the need for technology, i.e., will a change in the business process offer a solution?
- Assure executive business and executive technical sponsorship. The Sponsor designates a Project Manager and the business need is documented in a Concept Proposal. The Concept Proposal includes information about the business process and the relationship to the Agency/Organization.
- Infrastructure and the Strategic Plan. A successful Concept Proposal results in a Project Management Charter which outlines the authority of the project manager to begin the project.

Careful oversight is required to ensure projects support strategic business objectives and resources are effectively implemented into an organization's enterprise architecture. The initiation phase begins when an opportunity to add, improve, or correct a system is identified and formally requested through the presentation of a business case. The business case should, at a minimum, describe a proposal’s purpose, identify expected benefits, and explain how the proposed system supports one of the organization’s business strategies.

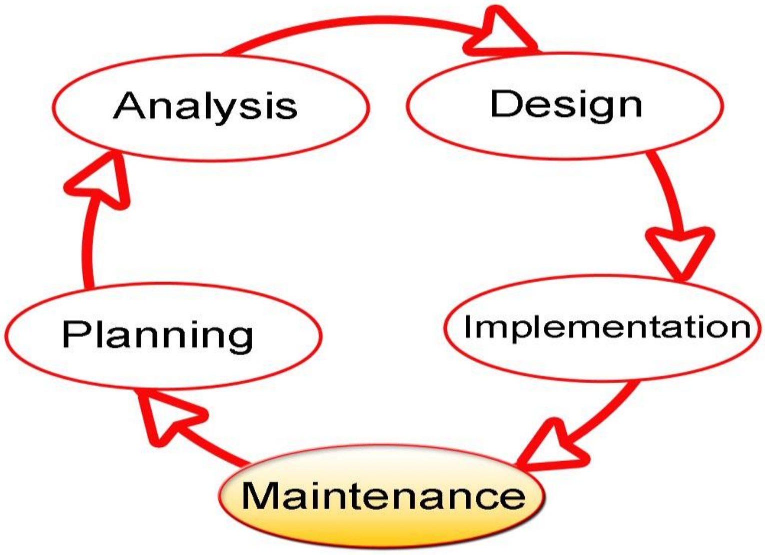
SYSTEM CONCEPT DEVELOPMENT PHASE

The System Concept Development Phase begins after a business need or opportunity is validated by the Agency/Organization Program Leadership and the Agency/Organization CIO.

The purpose of the System Concept Development Phase is to:

- Determine the feasibility and appropriateness of the alternatives.
- Identify system interfaces.
- Identify basic functional and data requirements to satisfy the business need.
- Establish system boundaries; identify goals, objectives, critical success factors, and performance measures.
- Evaluate costs and benefits of alternative approaches to satisfy the basic functional requirements
- Assess project risks
- Identify and initiate risk mitigation actions, and develop high-level technical architecture, process models, data models, and a concept of operations. This phase explores potential technical solutions within the context of the business need.
- It may include several trade-off decisions such as the decision to use COTS software products as opposed to developing custom software or reusing software components, or the decision to use an incremental delivery versus a complete, onetime deployment.
- Construction of executable prototypes is encouraged to evaluate technology to support the business process. The System Boundary Document serves as an important reference document to support the Information Technology Project Request (ITPR) process.
- The ITPR must be approved by the State CIO before the project can move forward.

PICTORIAL REPRESENTATION OF SDLC:



PLANNING PHASE

The planning phase is the most critical step in completing development, acquisition, and maintenance projects. Careful planning, particularly in the early stages of a project, is necessary to coordinate activities and manage project risks effectively. The depth and formality of project plans should be commensurate with the characteristics and risks of a given project. Project plans refine the information gathered during the initiation phase by further identifying the specific activities and resources required to complete a project.

A critical part of a project manager’s job is to coordinate discussions between user, audit, security, design, development, and network personnel to identify and document as many functional, security, and network requirements as possible. During this phase, a plan is developed that documents the approach to be used and includes a discussion of methods, tools, tasks, resources, project schedules, and ‘user input. Personnel assignments, costs, project schedule, and target dates are established.

A Project Management Plan is created with components related to acquisition planning, configuration management planning, quality assurance planning, concept of operations, system security, verification and validation, and systems engineering management planning.

REQUIREMENTS ANALYSIS PHASE

This phase formally defines the detailed functional user requirements using high-level requirements identified in the Initiation, System Concept, and Planning phases. It also delineates the requirements in terms of data, system performance, security, and maintainability requirements for the system. The requirements are defined in this phase to a level of detail sufficient for systems design to proceed. They need to be measurable, testable, and relate to the business need or opportunity identified in the Initiation Phase. The requirements that will be used to determine acceptance of the system are captured in the Test and Evaluation Master Plan.

The purposes of this phase are to:

- Further define and refine the functional and data requirements and document them in the Requirements Document,
- Complete business process reengineering of the functions to be supported (i.e., verify what information drives the business process, what information is generated, who generates it, where does the information go, and who processes it),
- Develop detailed data and process models (system inputs, outputs, and the process.
- Develop the test and evaluation requirements that will be used to determine acceptable system performance**.**

DESIGN PHASE

The design phase involves converting the informational, functional, and network requirements identified during the initiation and planning phases into unified design specifications that developers use to script programs during the development phase. Program designs are constructed in various ways. Using a top-down approach, designers first identify and link major program components and interfaces, then expand design layouts as they identify and link smaller subsystems and connections. Using a bottom-up approach, designers first identify and link minor program components and interfaces, then expand design layouts as they identify and link larger systems and connections. Contemporary design techniques often use prototyping tools that build mock-up designs of items such as application screens, database layouts, and system architectures. End users, designers, developers, database managers, and network administrators should review and refine the prototyped designs in an iterative process until they agree on an acceptable design. Audit, security, and quality assurance personnel should be involved in the review and approval process. During this phase, the system is designed to satisfy the functional requirements identified in the previous phase. Since problems in the design phase could be very expensive to solve in the later stage of the software development, a variety of elements are considered in the design to mitigate risk.

These include:

- Identifying potential risks and defining mitigating design features.
- Performing a security risk assessment.
- Developing a conversion plan to migrate current data to the new system.
- Determining the operating environment.
- Defining major subsystems and their inputs and outputs.
- Allocating processes to resources.
- Preparing detailed logic specifications for each software module. The result is a draft System Design Document which captures the preliminary design for the system.

- Everything requiring user input or approval is documented and reviewed by the user. Once these documents have been approved by the Agency CIO and Business Sponsor, the final System Design Document is created to serve as the Critical/Detailed Design for the system.
- This document receives a rigorous review by Agency technical and functional representatives to ensure that it satisfies the business requirements. Concurrent with the development of the system design, the Agency Project Manager begins development of the Implementation Plan, Operations and Maintenance Manual, and the Training Plan.

DEVELOPMENT PHASE

The development phase involves converting design specifications into executable programs. Effective development standards include requirements that programmers and other project participants discuss design specifications before programming begins. The procedures help ensure programmers clearly understand program designs and functional requirements. Programmers use various techniques to develop computer programs. The large transaction oriented programs associated with financial institutions have traditionally been developed using procedural programming techniques. Procedural programming involves the line-by-line scripting of logical instructions that are combined to form a program. Effective completion of the previous stages is a key factor in the success of the Development phase. The Development phase consists of:

- Translating the detailed requirements and design into system components.
- Testing individual elements (units) for usability.
- Preparing for integration and testing of the IT system.

INTEGRATION AND TEST PHASE

- Subsystem integration, system, security, and user acceptance testing is conducted during the integration and test phase. The user, with those responsible for quality assurance, validates that the functional requirements, as defined in the functional requirements document, are satisfied by the developed or modified system. OIT Security staff assesses the system security and issue a security certification and accreditation prior to installation/implementation.

Multiple levels of testing are performed, including:

- Testing at the development facility by the contractor and possibly supported by end users

- Testing as a deployed system with end users working together with contract personnel

- Operational testing by the end user alone performing all functions.

Requirements are traced throughout testing, a final Independent Verification & Validation evaluation is performed and all documentation is reviewed and accepted prior to acceptance of the system.

IMPLEMENTATION PHASE

This phase is initiated after the system has been tested and accepted by the user. In this phase, the system is installed to support the intended business functions. System performance is compared to performance objectives established during the planning phase. Implementation includes user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes. This phase continues until the system is operating in production in accordance with the defined user requirements.

OPERATIONS AND MAINTENANCE PHASE

The system operation is on-going. The system is monitored for continued performance in accordance with user requirements and needed system modifications are incorporated. Operations continue as long as the system can be effectively adapted to respond to the organization’s needs. When modifications or changes are identified, the system may re-enter the planning phase.

The purpose of this phase is to:

- Operate, maintain, and enhance the system.
- Certify that the system can process sensitive information.
- Conduct periodic assessments of the system to ensure the functional requirements continue to be satisfied.
- Determine when the system needs to be modernized, replaced, or retired.

SOURCE CODE

```
import mysql.connector, datetime

mycon =
mysql.connector.connect(host='localhost',user='root',passwd='
Thomas@123',database='hotel')

if mycon.is_connected():

    print('Successfully connected.')


cursor = mycon.cursor()


# CREATING TABLES

createTable ='''CREATE TABLE IF NOT EXISTS C_DETAILS(CID
VARCHAR(20) PRIMARY KEY,C_NAME VARCHAR(30),C_ADDRESS
VARCHAR(100),C_AGE INT,

C_COUNTRY VARCHAR(30) ,P_NO VARCHAR(30),C_EMAIL
VARCHAR(30))'''

cursor.execute(createTable)


createTable ="CREATE TABLE IF NOT EXISTS BOOKING_RECORD(CID
VARCHAR(20),CHECK_IN DATE ,CHECK_OUT DATE, ROOM_CHOICE INT,
ROOMNO INT, ROOMRENT INT)"

cursor.execute(createTable)


createTable = '''CREATE TABLE IF NOT EXISTS RESTAURANT(ROOMNO
INT, CUISINE INT, QUANTITY INT, RESTAURANTBILL INT)'''

cursor.execute(createTable)


createTable ='''CREATE TABLE IF NOT EXISTS
ENTERTAINMENT(ROOMNO INT, GAMES INT,HOURS
INT,ENTERTAINMENT_BILL INT)'''
```

```
cursor.execute(createTable)


createTable = '''CREATE TABLE IF NOT EXISTS TOTAL(CID
VARCHAR(20), C_NAME VARCHAR(30), ROOMNO INT, CHECK_IN DATE,
CHECK_OUT DATE, ROOMRENT INT,

RESTAURANTBILL INT, ENTERTAINMENTBILL INT, TOTALAMOUNT INT)'''

cursor.execute(createTable)

mycon.commit()


def days(a,b):

    l1,l2 = a.split('-'), b.split('-')

    y1, m1, d1 = int(l1[0]),int(l1[1]),int(l1[2])

    y2, m2, d2 = int(l2[0]),int(l2[1]),int(l2[2])

    day1 = datetime.date(y1,m1,d1)

    day2 = datetime.date(y2,m2,d2)

    delta = day2 - day1

    return delta.days


def newuser():

    if mycon.is_connected():

        cursor=mycon.cursor()

        cid = input("Enter Customer Identification Number : ")

        if searchcid(cid) != True:

            name = input("Enter Customer Name : ")

            address = input("Enter Customer Address : ")

            age= input("Enter Customer Age : ")
            nationality = input("Enter Customer Country : ")
            phoneno= input("Enter Customer Contact Number : ")
            email = input("Enter Customer Email : ")
            sql = "INSERT INTO C_Details
VALUES(%s,%s,%s,%s,%s,%s,%s)"
            values=
(cid,name,address,age,nationality,phoneno,email)
            cursor.execute(sql,values)
            mycon.commit()
            print("\nNew Customer Entered In The System
Successfully !")
            cursor.close()
        else:
            print("This customer ID already exists. Please
allot another ID.")


def searchcid(x):
    if mycon.is_connected():
        cursor = mycon.cursor()
        query = "SELECT * FROM C_DETAILS WHERE CID = %s"
        cursor.execute(query,(x,))
        data = cursor.fetchall()
        if data:
            return True
        else:
            return False
```

```
def searchuser():
    if mycon.is_connected():
        cursor = mycon.cursor()
        print('''\n\tSEARCH BY: 1)Name    2)Phone no
3)Customer ID\n''')
        choice3 = int(input('Enter your choice: '))
        if choice3 == 1:
            name = input('Enter Customer Name: ')
            query = "SELECT * FROM C_DETAILS WHERE C_NAME =
%s"
            cursor.execute(query,(name,))
            data = cursor.fetchall()

        elif choice3 == 2:
            phno = input('Enter Customer Contact Number: ')
            query = "SELECT * FROM C_DETAILS WHERE P_NO = %s"
            cursor.execute(query,(phno,))
            data = cursor.fetchall()

        elif choice3 == 3:
            cid = input('Enter Customer ID: ')
            query = "SELECT * FROM C_DETAILS WHERE CID = %s"
            cursor.execute(query,(cid,))
            data = cursor.fetchall()
        else:
            print('Invalid Input. Going Back to First Menu.')
            return

        if data:
            for row in data:
                print('\nID\t :',row[0])
                print('NAME\t :',row[1])
                print('ADDRESS  :',row[2])
                print('AGE\t :',row[3])
                print('COUNTRY  : ',row[4])
                print('PHONE NO :',row[5])
                print('EMAIL\t :',row[6])
        else:
            print("Record Not Found.")



def userEntry():
    print("""
        1---> New Customer
        2---> Existing Customer
    """)
    choice2 = int(input("Enter your choice: "))
    if choice2 == 1:
        newuser()
    elif choice2 == 2:
        searchuser()
```

```
def searchroom(x):
    if mycon.is_connected():
        cursor = mycon.cursor()
        query = "SELECT * FROM BOOKING_RECORD WHERE ROOMNO =
%s"
        cursor.execute(query,(x,))
        data = cursor.fetchall()
        if data:
            return True
        else:
            return False


def bookingRecord():
    cid = input("Enter Customer ID: ")
    if searchcid(cid):
        cursor=mycon.cursor()
        checkin=input("\nEnter Customer CheckIN Date [ YYYY-
MM-DD ] : ")
        checkout=input("Enter Customer CheckOUT Date [ YYYY-
MM-DD ] : ")
        print ("\n##### We have The Following Rooms For You
#####")
        print ("1. Ultra Royal ----> 5000 Rs.")
        print ("2. Royal ----> 4000 Rs. ")
        print ("3. Elite ----> 3000 Rs. ")
        print ("4. Budget ----> 2000 Rs. ")
        roomchoice=int(input("\nEnter Your Choice : "))
        while True:
            roomno=int(input("Enter Allotted Room No : "))
            if searchroom(roomno) != True:
                break
            else:
                print("Room is already reserved. Please allot
another room.")


        query="INSERT INTO BOOKING_RECORD
VALUES(%s,%s,%s,%s,%s,NULL)"
        values=(cid,checkin,checkout,roomchoice,roomno)
        cursor.execute(query,values)
        mycon.commit()
        noofdays = days(checkin,checkout)


        if roomchoice==1:
            roomrent = noofdays * 5000
            print("\nUltra Royal Room Rent : ",roomrent)
        elif roomchoice==2:
            roomrent = noofdays * 4000
            print("\nRoyal Room Rent : ",roomrent)
        elif roomchoice==3:
            roomrent = noofdays * 3000
            print("\nElite Royal Room Rent : ",roomrent)
        elif roomchoice==4:
            roomrent = noofdays * 2000
            print("\nBudget Room Rent : ",roomrent)
        else:
            print("Invalid Input. Going Back To First Menu.")
            return

        print("Thank You , Your Room Has Been Booked For : 
",noofdays , "Days" )
        print("Your Total Room Rent is : Rs. ",roomrent)

        query3 = "UPDATE BOOKING_RECORD SET ROOMRENT = %s 
WHERE ROOMNO = %s"
        cursor.execute(query3,(roomrent,roomno))
        mycon.commit()

        print("\nCHECK-IN AND CHECK-OUT ENTRY MADE 
SUCCESSFULLY !")
        cursor.close()
    else:
        print("Customer ID does not exist. Kindly enter the 
correct customer ID.")

def Restaurant():
    if mycon.is_connected():
        cursor = mycon.cursor()
        roomno = int(input("Enter Room No: "))
        if searchroom(roomno):
            print("1. Vegetarian Combo                  ----->
300 Rs.")
            print("2. Non-Vegetarian Combo              ----->
500 Rs.")
            print("3. Vegetarian & Non-Vegetarian Combo ----->
750 Rs.")
            choice_dish = int(input("\nEnter Your Choice : "))
            quantity=int(input("Enter Quantity : "))
            if choice_dish==1:
                print("\nYou Have Ordered: Vegetarian Combo ")
                restaurantbill = quantity * 300
            elif choice_dish==2:
                print("\nYou Have Ordered: Non-Vegetarian
Combo ")
                restaurantbill = quantity * 500
            elif choice_dish==3:
                print("\nYou Have Ordered: Vegetarian & Non-
Vegetarian Combo ")
                restaurantbill= quantity * 750
            else:
                print("Invalid Input. Going Back To First
Menu.")
                return

            sql= "INSERT INTO RESTAURANT VALUES(%s,%s,%s,%s)"
            values=
(roomno,choice_dish,quantity,restaurantbill)
            cursor.execute(sql,values)
            mycon.commit()

            print("Your Total Bill Amount Is : Rs.
",restaurantbill)

            print("**** WE HOPE YOU WILL ENJOY YOUR MEAL
***\n" )

        else:

            print("Invalid Room Number.")


def Entertainment():

    if mycon.is_connected():

        cursor = mycon.cursor()

        roomno = int(input("Enter room no: "))

        if searchroom(roomno):

            print("""

            1. Table Tennis -----> 150 Rs./HR

            2. Bowling -----> 100 Rs./HR

            3. Snooker -----> 250 Rs./HR

            4. VR World Gaming -----> 400 Rs./HR

            5. Video Games -----> 300 Rs./HR

            6. Swimming Pool Games -----> 350 Rs./HR

            7. Exit

            """)

            game=int(input("Enter Game You Want To Play : "))

            hours=int(input("Enter No Of Hours : "))

print("\n\n#################################################")

            if game==1:
                print("YOU HAVE SELECTED TO PLAY : Table
Tennis")
                entertainmentbill = hours * 150
            elif game==2:
                print("YOU HAVE SELECTED TO PLAY : Bowling")
                entertainmentbill = hours * 100
            elif game==3:
                print("YOU HAVE SELECTED TO PLAY : Snooker")
                entertainmentbill = hours * 250
            elif game==4:
                print("YOU HAVE SELECTED TO PLAY : VR World
Gaming")
                entertainmentbill = hours * 400
            elif game==5:
                print("YOU HAVE SELECTED TO PLAY : Video
Games")
                entertainmentbill = hours * 300
            elif game ==6:
                print("YOU HAVE SELECTED TO PLAY : Swimming
Pool Games")
                entertainmentbill = hours * 350
            else:
                print("Invalid Input. Going Back To First
Menu.")
                return
            query= "INSERT INTO ENTERTAINMENT
VALUES(%s,%s,%s,%s)"
            values= (roomno,game,hours,entertainmentbill)
            cursor.execute(query,values)

            mycon.commit()

            print("FOR : ",hours," HOURS")

            print("Your Total Entertainment Bill Is : Rs. 
",entertainmentbill)

        else:

            print("Invalid Room Number.")


def totalAmount():

    if mycon.is_connected():

        cursor=mycon.cursor()

        roomno = int(input("Enter room no: "))

        if searchroom(roomno):

            query1 = "SELECT BOOKING_RECORD.CID, 
C_DETAILS.C_NAME, CHECK_IN, CHECK_OUT, ROOMRENT FROM 
BOOKING_RECORD, C_DETAILS WHERE BOOKING_RECORD.ROOMNO = %s AND 
C_DETAILS.CID = BOOKING_RECORD.CID"

            cursor.execute(query1,(roomno,))

            data = cursor.fetchone()

            cid,name,checkin,checkout,roomrent = data


            query2 = "SELECT SUM(RESTAURANTBILL) FROM 
RESTAURANT WHERE ROOMNO = %s"

            cursor.execute(query2,(roomno,))

            data = cursor.fetchone()

            restaurantbill = data[0]

            if restaurantbill == None:
                restaurantbill = 0

            query3 = "SELECT SUM(ENTERTAINMENT_BILL) FROM ENTERTAINMENT WHERE ROOMNO = %s"
            cursor.execute(query3,(roomno,))
            data = cursor.fetchone()
            entertainmentbill = data[0]
            if entertainmentbill == None:
                entertainmentbill = 0

            query4 = "INSERT INTO TOTAL VALUES(%s,%s,%s,%s,%s,%s,%s,%s,%s)"
            grandTotal = roomrent + restaurantbill + entertainmentbill
            values = (cid,name,roomno,checkin,checkout,roomrent,restaurantbill,entertainmentbill,grandTotal)
            cursor.execute(query4,values)
            mycon.commit()
            print("\n **** HOTEL AMBROSIA, TVM **** CUSTOMER BILLING ****")
            print("\nCUSTOMER NAME:",name)
            print("ROOMNO: ",roomno)
            print("CHECKIN DATE:",checkin)
            print("CHECKOUT DATE:",checkout)
            print("ROOM RENT: Rs.",roomrent)
            print("RESTAURANT BILL: Rs.",restaurantbill)
            print("ENTERTAINMENT BILL: Rs.",entertainmentbill)
```

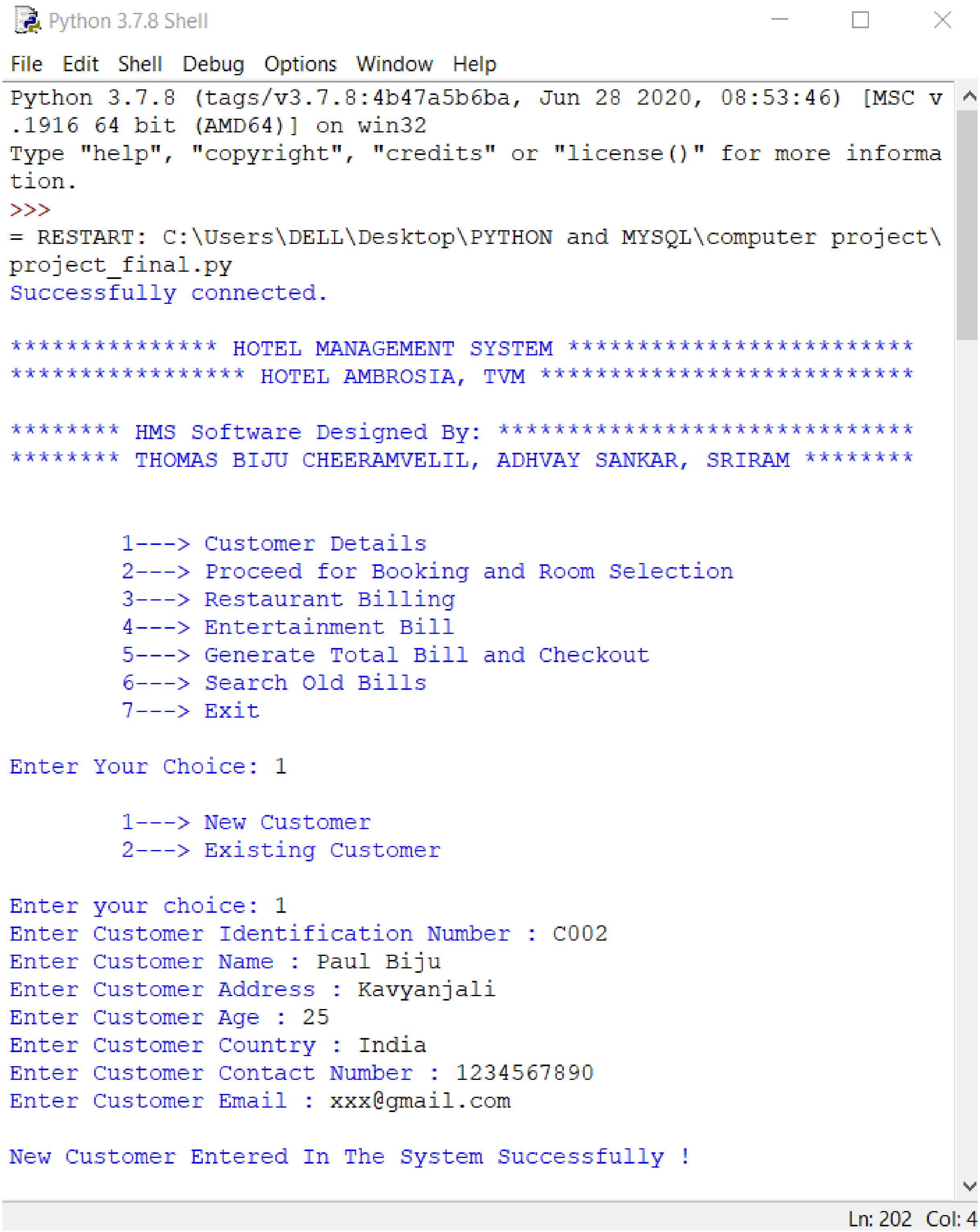
```
print("__________________________________________________")
            print("TOTAL AMOUNT : Rs.",grandTotal)
            query5 = "DELETE FROM BOOKING_RECORD WHERE ROOMNO
= %s"
            query6 = "DELETE FROM RESTAURANT WHERE ROOMNO =
%s"
            query7 = "DELETE FROM ENTERTAINMENT WHERE ROOMNO =
%s"
            cursor.execute(query5,(roomno,))
            cursor.execute(query6,(roomno,))
            cursor.execute(query7,(roomno,))
            mycon.commit()
            cursor.close()
        else:
            print("Invalid Room Number.")

def searchOldBill():
    if mycon.is_connected():
        cursor = mycon.cursor()
        print('''\n\tSEARCH BY: 1)Customer ID    2)Name\n''')
        choice = int(input('Enter your choice: '))
        if choice == 1:
            cid = input('Enter Customer ID: ')
            query = "SELECT * FROM TOTAL WHERE CID = %s"
            cursor.execute(query,(cid,))
            data = cursor.fetchall()
```

```
elif choice == 2:
    name = input('Enter Customer Name: ')
    query = "SELECT * FROM TOTAL WHERE C_NAME = %s"
    cursor.execute(query,(name,))
    data = cursor.fetchall()
else:
    print('Invalid Input. Please Try Again.')
    return

if data:
    for row in data:
        print('\nID\t\t   :',row[0])
        print('NAME\t\t   :',row[1])
        print('ROOMNO\t\t   :',row[2])
        print('CHECKIN DATE\t   :',row[3])
        print('CHECKOUT DATE\t   :',row[4])
        print('ROOM RENT\t   :',row[5])
        print('RESTAURANT BILL    :',row[6])
        print('ENTERTAINMENT BILL :',row[7])
        print('TOTAL AMOUNT\t   :',row[8])

else:
    print("Record Not Found. Please Try Again.")
```

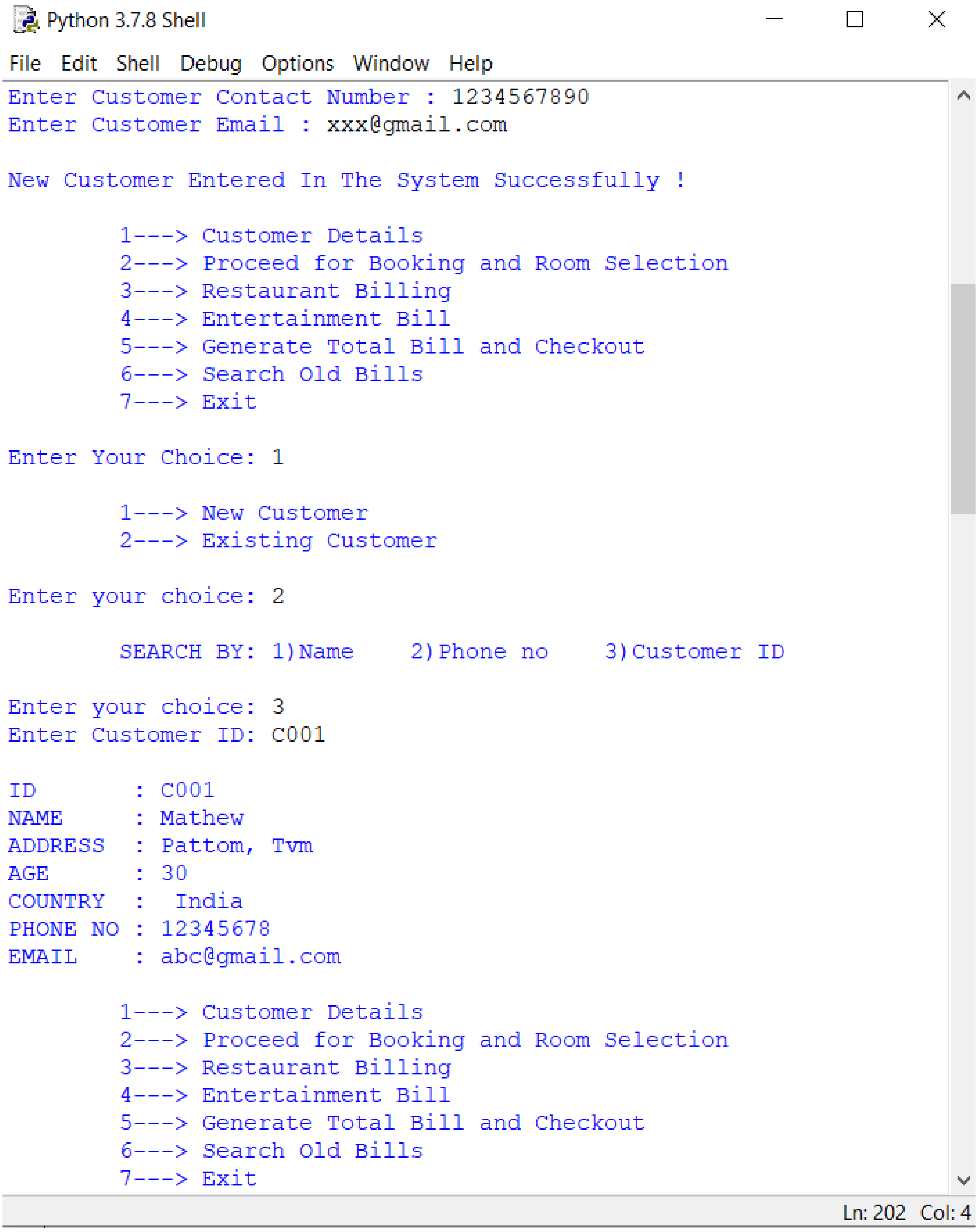
```
# Main program


print("""
*************** HOTEL MANAGEMENT SYSTEM
*************************
***************** HOTEL AMBROSIA, TVM
***************************


******** HMS Software Designed By:
******************************
******** THOMAS BIJU CHEERAMVELIL, ADHVAY SANKAR, SRIRAM
********
""")


while True:
    print("""
        1---> Customer Details
        2---> Proceed for Booking and Room Selection
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit \n""")


    choice = int(input("Enter Your Choice: "))
    if choice == 1:
        userEntry()
```

```
elif choice ==2:
    bookingRecord()
elif choice ==3:
    Restaurant()
elif choice ==4:
    Entertainment()
elif choice ==5:
    totalAmount()
elif choice ==6:
    searchOldBill()
elif choice ==7:
    break
else:
    print("Invalid Input.")
```

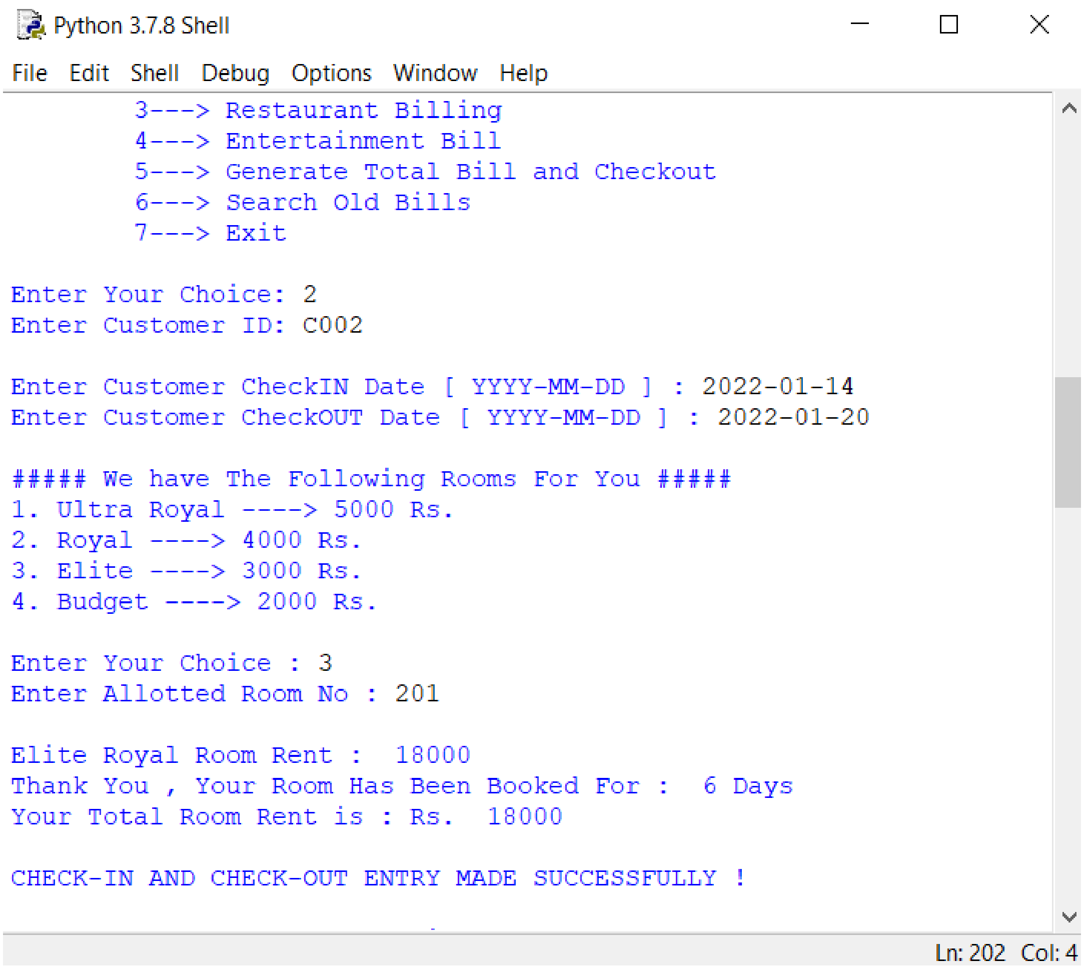
OUTPUT



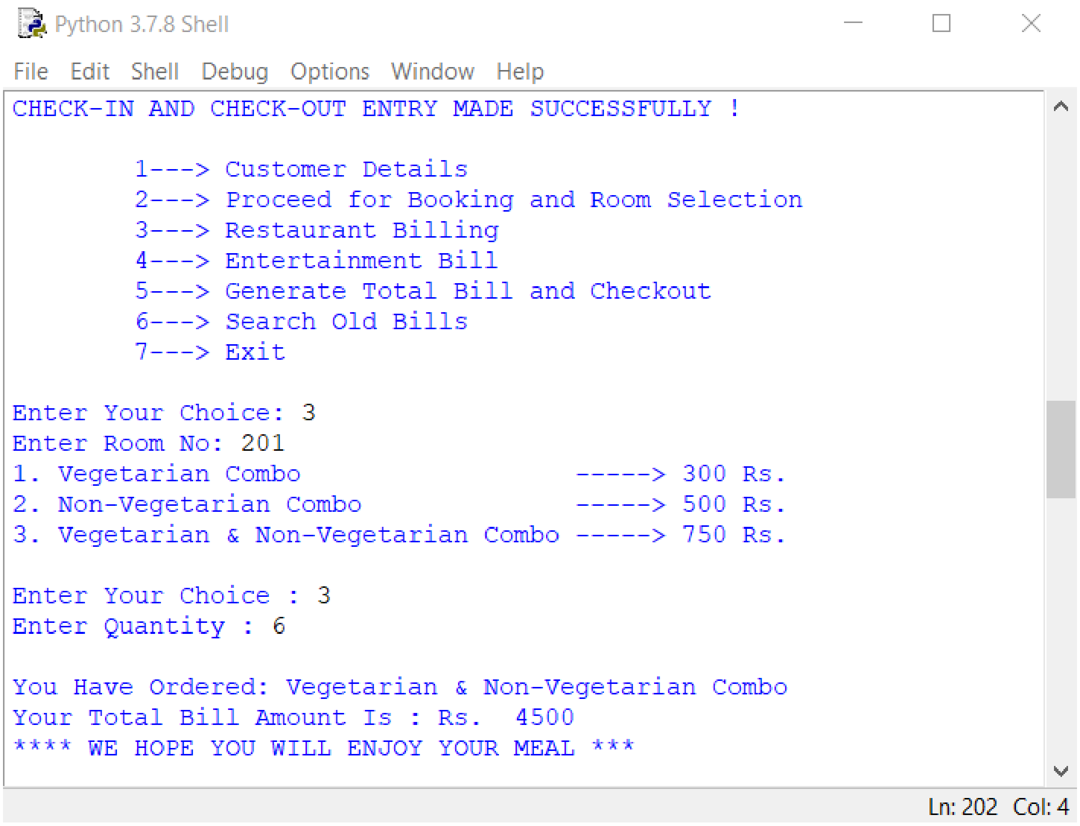
```
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v
.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informa
tion.
>>>
= RESTART: C:\Users\DELL\Desktop\PYTHON and MYSQL\computer project\
project_final.py
Successfully connected.

*************** HOTEL MANAGEMENT SYSTEM *************************
***************** HOTEL AMBROSIA, TVM ***************************

******** HMS Software Designed By: ******************************
******** THOMAS BIJU CHEERAMVELIL, ADHVAY SANKAR, SRIRAM ********


        1---> Customer Details
        2---> Proceed for Booking and Room Selection
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit

Enter Your Choice: 1

        1---> New Customer
        2---> Existing Customer

Enter your choice: 1
Enter Customer Identification Number : C002
Enter Customer Name : Paul Biju
Enter Customer Address : Kavyanjali
Enter Customer Age : 25
Enter Customer Country : India
Enter Customer Contact Number : 1234567890
Enter Customer Email : xxx@gmail.com

New Customer Entered In The System Successfully !
```



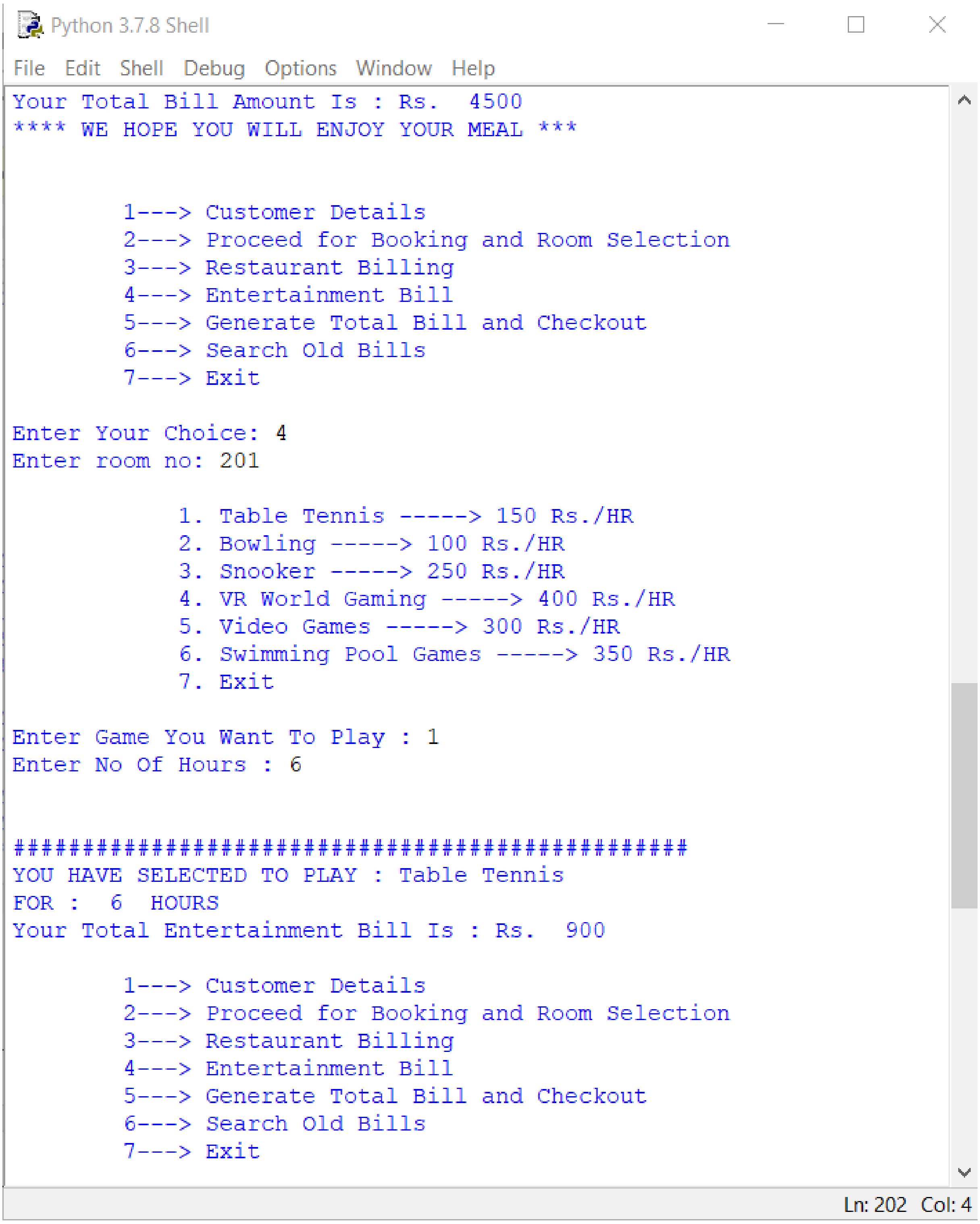
```
Enter Customer Contact Number : 1234567890
Enter Customer Email : xxx@gmail.com

New Customer Entered In The System Successfully !

        1---> Customer Details
        2---> Proceed for Booking and Room Selection
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit

Enter Your Choice: 1

        1---> New Customer
        2---> Existing Customer

Enter your choice: 2

        SEARCH BY: 1)Name    2)Phone no    3)Customer ID

Enter your choice: 3
Enter Customer ID: C001

ID         : C001
NAME       : Mathew
ADDRESS    : Pattom, Tvm
AGE        : 30
COUNTRY    :  India
PHONE NO   : 12345678
EMAIL      : abc@gmail.com

        1---> Customer Details
        2---> Proceed for Booking and Room Selection
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit
```



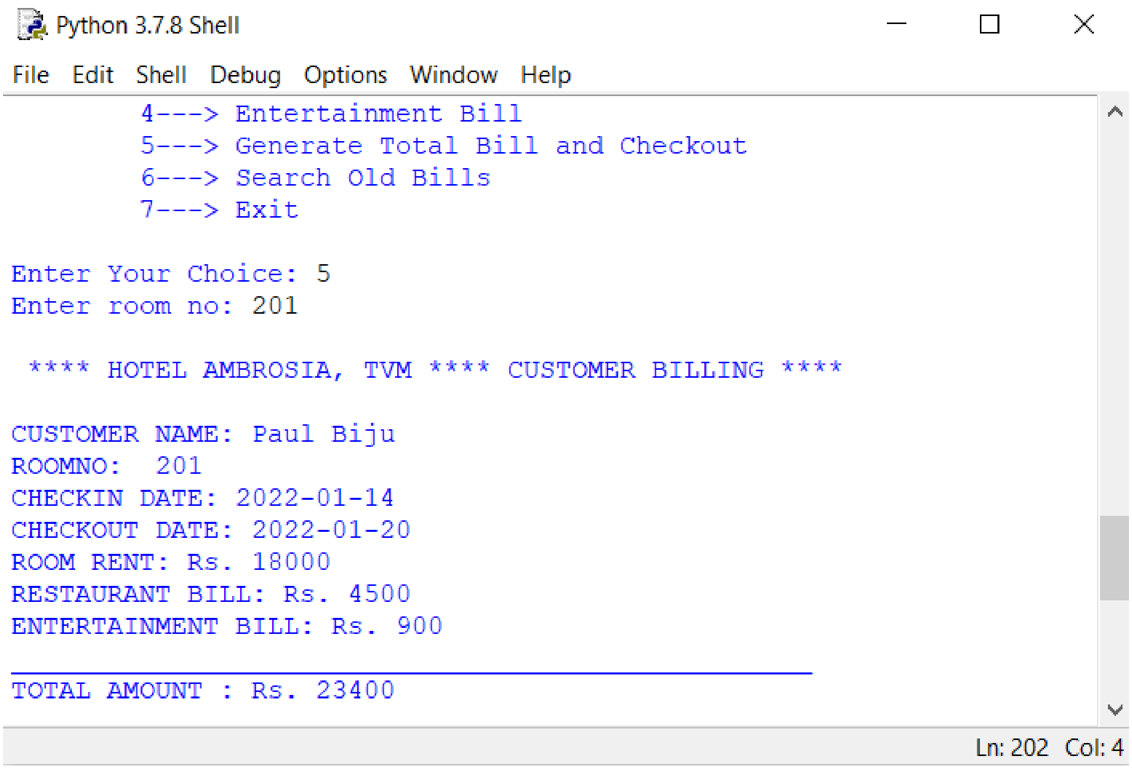
```
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit

Enter Your Choice: 2
Enter Customer ID: C002

Enter Customer CheckIN Date [ YYYY-MM-DD ] : 2022-01-14
Enter Customer CheckOUT Date [ YYYY-MM-DD ] : 2022-01-20

##### We have The Following Rooms For You #####
1. Ultra Royal ----> 5000 Rs.
2. Royal ----> 4000 Rs.
3. Elite ----> 3000 Rs.
4. Budget ----> 2000 Rs.

Enter Your Choice : 3
Enter Allotted Room No : 201

Elite Royal Room Rent :  18000
Thank You , Your Room Has Been Booked For :  6 Days
Your Total Room Rent is : Rs.  18000

CHECK-IN AND CHECK-OUT ENTRY MADE SUCCESSFULLY !
```



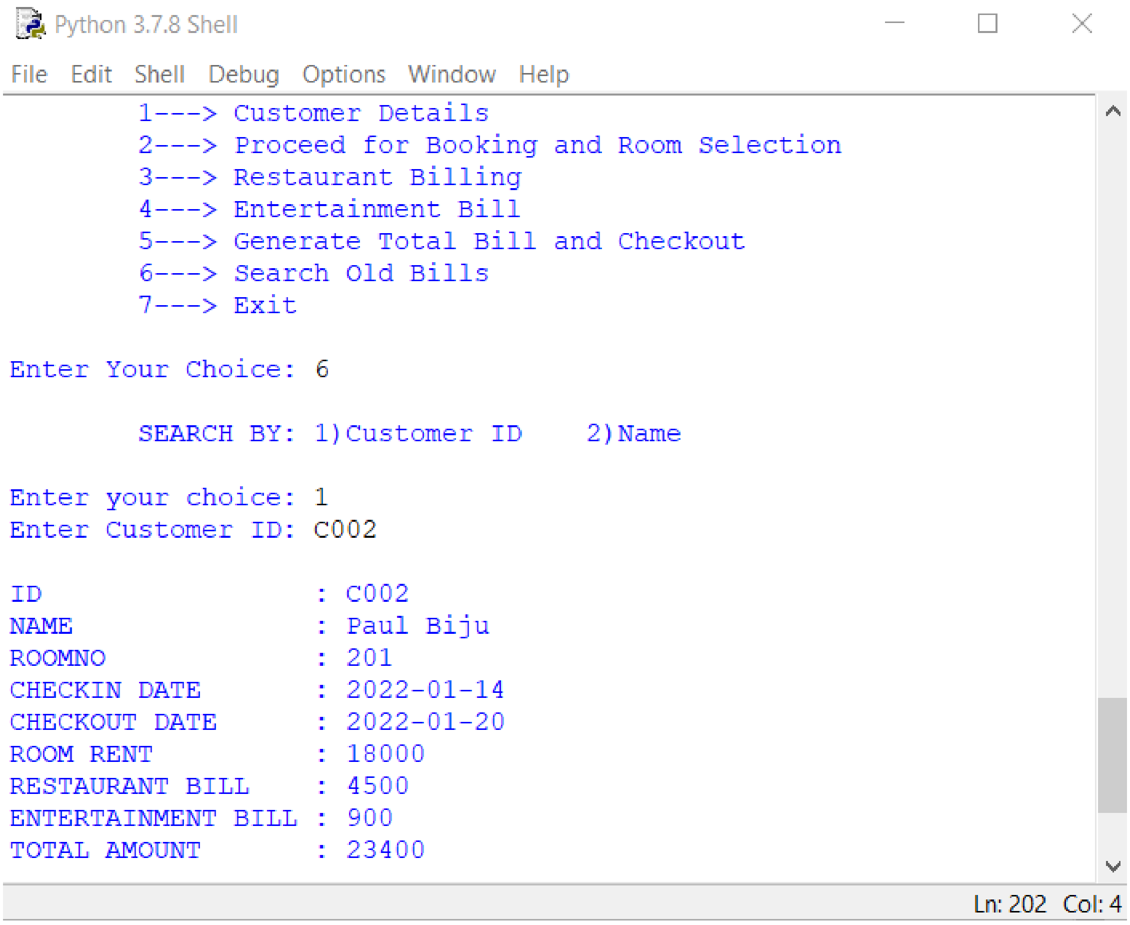
```
CHECK-IN AND CHECK-OUT ENTRY MADE SUCCESSFULLY !

        1---> Customer Details
        2---> Proceed for Booking and Room Selection
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit

Enter Your Choice: 3
Enter Room No: 201
1. Vegetarian Combo                -----> 300 Rs.
2. Non-Vegetarian Combo            -----> 500 Rs.
3. Vegetarian & Non-Vegetarian Combo -----> 750 Rs.

Enter Your Choice : 3
Enter Quantity : 6

You Have Ordered: Vegetarian & Non-Vegetarian Combo
Your Total Bill Amount Is : Rs.  4500
**** WE HOPE YOU WILL ENJOY YOUR MEAL ***
```



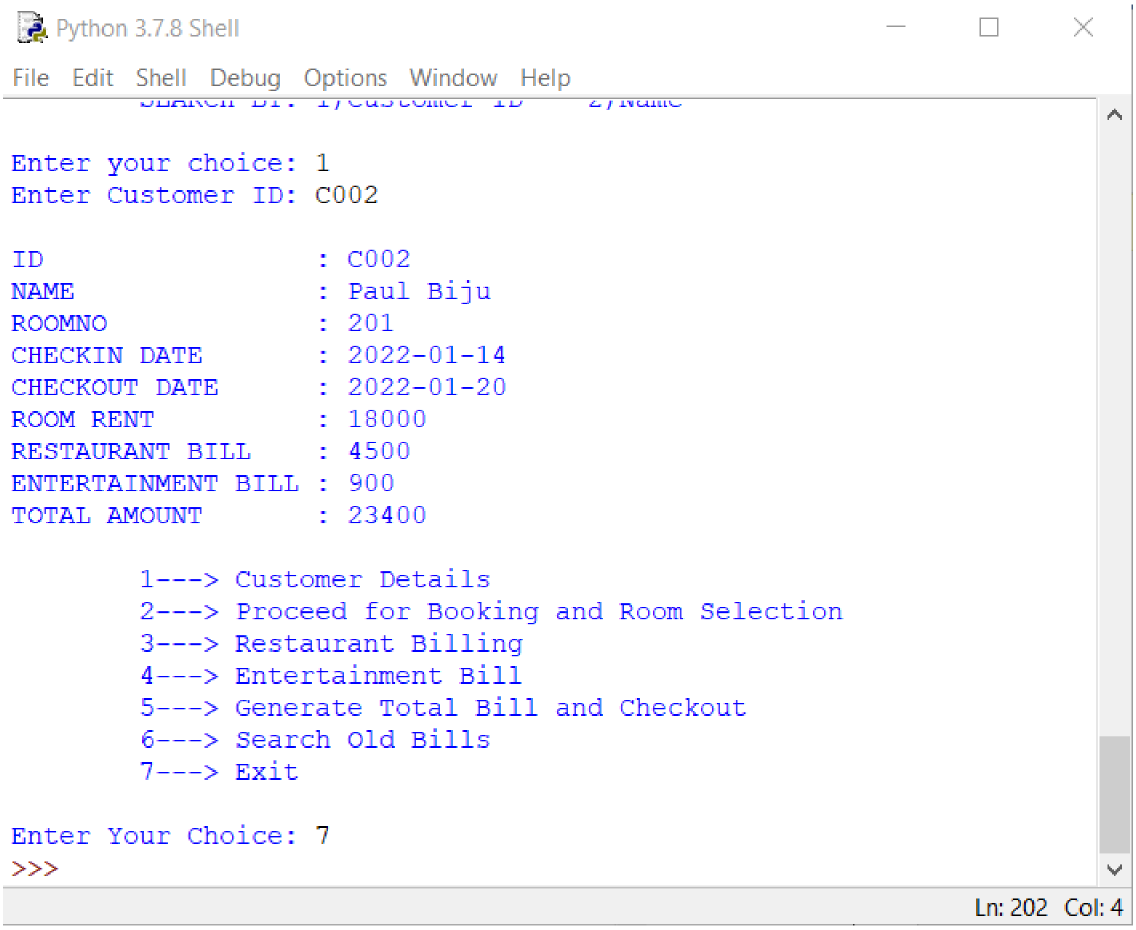
```
Your Total Bill Amount Is : Rs.  4500
**** WE HOPE YOU WILL ENJOY YOUR MEAL ***


        1---> Customer Details
        2---> Proceed for Booking and Room Selection
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit

Enter Your Choice: 4
Enter room no: 201

            1. Table Tennis -----> 150 Rs./HR
            2. Bowling -----> 100 Rs./HR
            3. Snooker -----> 250 Rs./HR
            4. VR World Gaming -----> 400 Rs./HR
            5. Video Games -----> 300 Rs./HR
            6. Swimming Pool Games -----> 350 Rs./HR
            7. Exit

Enter Game You Want To Play : 1
Enter No Of Hours : 6


#################################################
YOU HAVE SELECTED TO PLAY : Table Tennis
FOR :  6  HOURS
Your Total Entertainment Bill Is : Rs.  900

        1---> Customer Details
        2---> Proceed for Booking and Room Selection
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit
```



```
         4---> Entertainment Bill
         5---> Generate Total Bill and Checkout
         6---> Search Old Bills
         7---> Exit

Enter Your Choice: 5
Enter room no: 201

 **** HOTEL AMBROSIA, TVM **** CUSTOMER BILLING ****

CUSTOMER NAME: Paul Biju
ROOMNO:  201
CHECKIN DATE: 2022-01-14
CHECKOUT DATE: 2022-01-20
ROOM RENT: Rs. 18000
RESTAURANT BILL: Rs. 4500
ENTERTAINMENT BILL: Rs. 900
__________________________________________________
TOTAL AMOUNT : Rs. 23400
```



```
         1---> Customer Details
         2---> Proceed for Booking and Room Selection
         3---> Restaurant Billing
         4---> Entertainment Bill
         5---> Generate Total Bill and Checkout
         6---> Search Old Bills
         7---> Exit

Enter Your Choice: 6

         SEARCH BY: 1)Customer ID    2)Name

Enter your choice: 1
Enter Customer ID: C002

ID                   : C002
NAME                 : Paul Biju
ROOMNO               : 201
CHECKIN DATE         : 2022-01-14
CHECKOUT DATE        : 2022-01-20
ROOM RENT            : 18000
RESTAURANT BILL      : 4500
ENTERTAINMENT BILL   : 900
TOTAL AMOUNT         : 23400
```



```
        SEARCH BY: 1)Customer ID    2)Name

Enter your choice: 1
Enter Customer ID: C002

ID                   : C002
NAME                 : Paul Biju
ROOMNO               : 201
CHECKIN DATE         : 2022-01-14
CHECKOUT DATE        : 2022-01-20
ROOM RENT            : 18000
RESTAURANT BILL      : 4500
ENTERTAINMENT BILL : 900
TOTAL AMOUNT         : 23400

        1---> Customer Details
        2---> Proceed for Booking and Room Selection
        3---> Restaurant Billing
        4---> Entertainment Bill
        5---> Generate Total Bill and Checkout
        6---> Search Old Bills
        7---> Exit

Enter Your Choice: 7
>>>
```

TESTING

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test[1] , with respect to the context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

TESTING METHODS

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

BLACK BOX TESTING

Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

SPECIFICATION-BASED TESTING

Specification-based testing aims to test the functionality of software according to the applicable requirements.[16] Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behaviour), either "is" or "is not" the same as the expected value specified in the test case. Specification-based testing is necessary, but it is insufficient to guard against certain risks

ADVANTAGES AND DISADVANTAGES

The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code must have bugs. Using the principle, "Ask and you shall receive," black box testers find bugs where programmers don't. But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed.

That's why there are situations when (1) a black box tester writes many test cases to check something that can be tested by only one test case, and/or (2) some parts of the back end are not tested at all. Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

WHITE BOX TESTING

White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these)

Types of white box testing:-

The following types of white box testing exist:

- api testing - Testing of the application using Public and Private APIs.
- Code coverage - creating tests to satisfy some criteria of code coverage.

For example, the test designer can create tests to cause all statements in the program to be executed at least once. ✚ fault injection methods. ✚ mutation testing methods. ✚ static testing - White box testing includes all static testing.

CODE COMPLETENESS EVALUATION

White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.

Two common forms of code coverage are:

- Function Coverage: Which reports on functions executed and
- Statement Coverage: Which reports on the number of lines executed to complete the test.

They both return coverage metric, measured as a percentage

HARDWARE AND SOFTWARE REQUIREMENTS

I.	OPERATING SYSTEM	WINDOWS 7 AND ABOVE
II.	PROCESSOR	PENTIUM(ANY) OR AMD ATHALON(3800+- 4200+ DUALCORE)
III.	MOTHERBOARD	1.845 OR 915,995 FOR PENTIUM 0R MSI K9MM-V VIAK8M800+8237R PLUS CHIPSET FOR AMD ATHALON
IV.	RAM	512MB+
V.	Hard disk	SATA 40 GB OR ABOVE
VI.	CD/DVD r/w multi drive combo	(If back up required)
VII.	FLOPPY DRIVE 1.44 MB	(If Backup required)
VIII.	MONITOR 14.1 or 15 -17 inch	
IX.	Key board and mouse	
X.	Printer	(if print is required – [Hard copy])

SOFTWARE REQUIREMENTS:

1. Windows OS
2. Python
3. MySQL

BIBLOGRAPHY

1. ***Computer science With Python - Class XI/XII By : Sumita Arora***
2. ***A Project Report On Blood Bank Management System (BBMS) By : Praveen M Jigajinni***
3. ***Website:* https://python4csip.com/**

***