

**ALL INDIA SENIOR SECONDARY CERTIFICATE
EXAMINATION-2022**

Central Board of Secondary Education, New Delhi

A Project Report On

RAJIV CHOWK PARKING SYSTEM

Submitted By-

Name: Khushal Jhingan (21665094)

Class – XII, Section – A

Khaitan Public School, Sahibabad

Session – 2021-2022

Under the guidance of – Mr. S. S. Das

(PGT – Computer Science)

Department of Computer Science

Khaitan Public School, Sahibabad

CERTIFICATE

(By Internal Examiner)

This is to certify that **Khushal Jhingan** of class XII-A have prepared the report on the project entitled "Rajiv Chowk Parking System". The report is result of their efforts and endeavours. The report is found worthy of acceptance as final project report for the subject Computer Science (083) of class XII. They have prepared the report under my guidance.

(S. S. Das)

PGT (Computer Science)
Department of Computer
Science Khaitan Public School
Sahibabad

CERTIFICATE

(By External Examiner)

The project report entitled
“Rajiv Chowk Parking System”

Submitted by **Khushal Jhingan** of class XII-A for the CBSE Senior Secondary Examination 2022 for the subject Computer Science (083) has been examined.

Signature of External Examiner

Examiner No:

ACKNOWLEDGEMENT

We would like to express a deep sense of thanks & guidance to our project guide **Mr. S. S. Das** Sir for guiding us immensely through the course of the project. He always evinced keen interest in our work. His constructive advice & constant motivation have been responsible for the successful completion of this project.

Last but not the least; We would like to thank all those who had helped directly or indirectly towards the completion of this project.

Name: Khushal Jhingan (21665094)

Class: XII-A

Khaitan Public School, Sahibabad Session – 2021-2022

Table of contents

Sl.No	Topic	Page No.
1	PYTHON Overview	6
2	Object Oriented Programming – An Introduction	6
3	Project Synopsis	7-11
4	Hardware Requirements	12
5	Software Requirements	12
6	Installation	13
7	Sample Output	14
8	Coding & Output	15-28

PYTHON Overview

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, making it very attractive for Rapid Application Development and used as a scripting or glue language to connect existing components. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

Object-Oriented Programming

PYTHON fully supports object-oriented programming, including the four pillars of object-oriented development:

1. **Encapsulation:** It describes the idea of wrapping data and the methods that work on data within one unit. This puts restrictions on accessing variables and methods directly and can prevent the accidental modification of data.
2. **Abstraction:** Abstraction is used to hide the internal functionality of the function from the users. The users only interact with the basic implementation of the function, but inner working is hidden. User is familiar with that "**what function does**" but they don't know "**how it does.**"
3. **Inheritance:** Inheritance allows us to define a class that inherits all the methods and properties from another class
4. **Polymorphism:** The word polymorphism means having many forms. In programming, polymorphism means the same function

Synopsis:

i) Proposed System Overview:

The proposed system is designed for Rajiv Chowk Parking System in a **semi-Automated** way that requires minimal human usage, that too only for data entry, which rules out the chances of **human error**. The data entry operator has the whole System under his control. When a vehicle reaches the entry gate, the operator first checks the total number of spaces available; if spaces are available, the operator enters the vehicle registration number and name of the driver in the System. He also gives a receipt with a **code** on it required during exit; the details of the same are stored in the database at the **backend**. During exit, the vehicle owner will give the receipt back to the operator. The operator will enter the code in the System. **If the code and all details match**, then only the Vehicle will be allowed to exit. The parking fee will also be given at the exit gate to the operator, which the **System will calculate** based on the time the Vehicle has spent in the parking. All the data will be saved in a **binary file** and also in the database. These files can be accessed by the **administrator**, for which he needs to login into the admin account by entering the user Id and Password provided. The administrator can see the daily report in a tabular form. He can also get information about the total revenue generated the whole day and access many other features.

ii) Current System In use:

The current System used is manually operated by a person in which a handwritten receipt is given at the entry point, which needs to be deposited at the exit. The revenue is both collected and maintained by the administrator of the parking at that time. At the exit point, the receipt is returned to the person along with parking fees.

iii) Challenges in the Current System:

- 1) This System always requires at least two people, which is a pretty tiresome job.
- 2) The receipt is handwritten, and mostly all the Vehicle details are not filled in it.
- 3) The parking fee is the same whether the Vehicle is parked for an hour or more than that.
- 4) The Vehicle owners are unaware of whether the parking space is available or not.
- 5) The daily records are not maintained at the backend.
- 6) The total revenue generated in a day is not recorded anywhere, due to which there are chances of cash embezzlement.
- 7) There are high possibilities that the in-charge may forget to update the records at his end.
- 8) There are chances that the resource person may miss out on accepting payment at the exit
- 9) Data is not encrypted and can lead to data redundancy

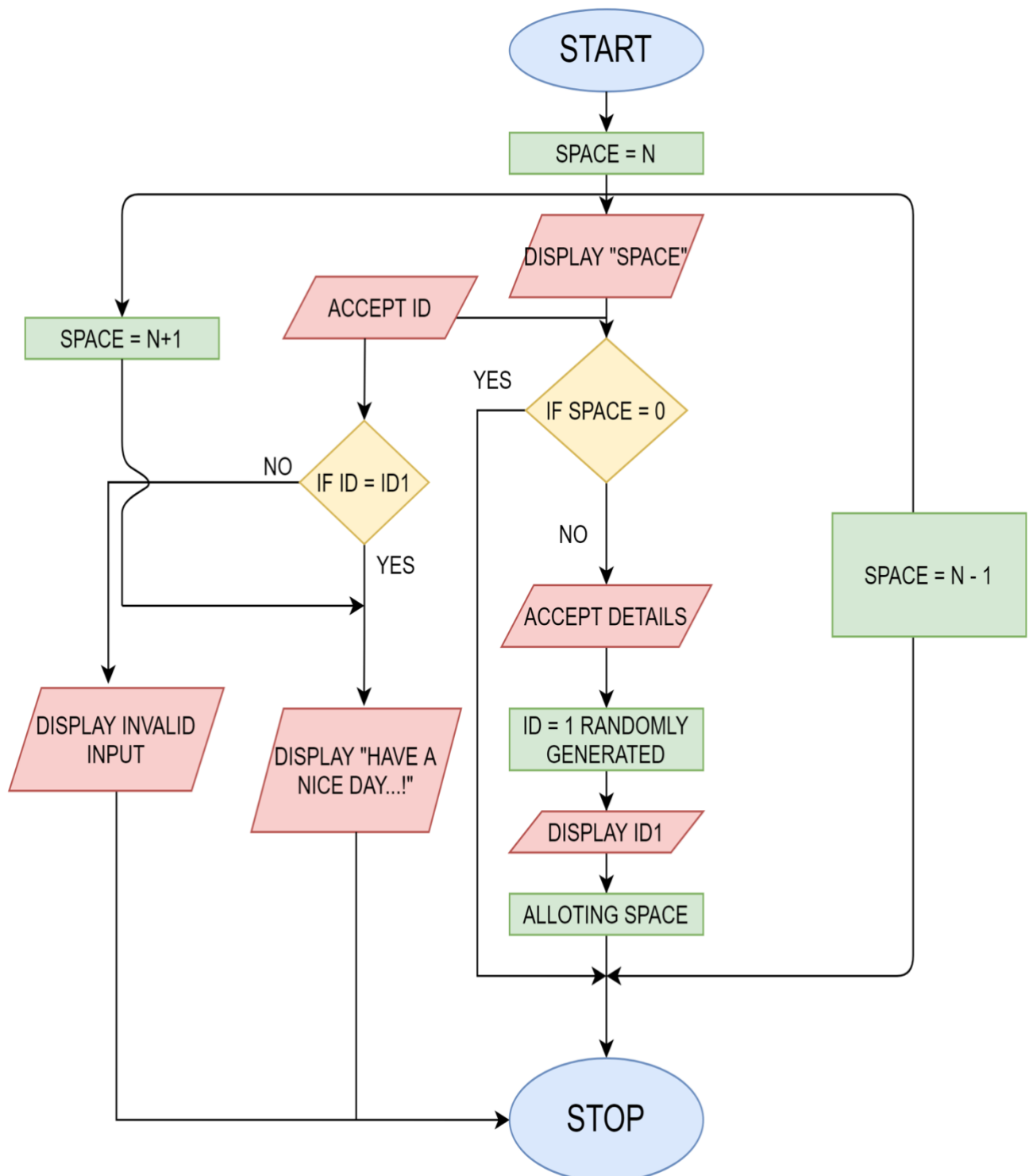
iv) Advantages Of Proposed System:

The proposed System of Rajiv Chowk Parking System has been designed to facilitate the computerized **Semi-automated** System over the current manual System. This has been observed that the present system has many loopholes, leading to faulty system output and **data redundancy**. Many manual/handwritten processes were involved in the current system, which has been found and converted into error-less output.

The proposed system has the following salient features:

- i) All the modules are **loosely coupled** and highly **cohesive**.
- ii) Less human intervention.
- iii) Hassle-free data entry
- iv) Back-up of data in **binary** files
- v) **Secure** Log In System
- vi) Computer **Proficiency** not required to operate the System
- vii) **Automatic** report generation
- viii) **Environment** friendly (Reduces use of paper)
- ix) Cost-Effective

v) Data flow Diagram of the System:



vi) Files to be Used in Proposed System:

- 1) parking_data_file.dat(For storing daily data at Backend)

vii) Structure of Database Files:

Database Name: Parking_System (In MySQL)

- 1) Table Name: floor_1(For telling current available spaces on floor 1)
Structure:

Field Name	Type	Null	Default
Pillars	varchar(2)	YES	NULL
Reserved	varchar(7)	YES	NOT RES

- 2) Table Name: floor_2(For telling current available spaces on floor 2)
Structure:

Field Name	Type	Null	Default
Pillars	varchar(2)	YES	NULL
Reserved	varchar(7)	YES	NOT RES

- 3)** Table Name: floor_3(For telling current available spaces on floor 3)
Structure:

Field Name	Type	Null	Default
Pillars	varchar(2)	YES	NULL
Reserved	varchar(7)	YES	NOT RES

- 4)** Table Name: logfiles(For storing Vehicle Details)
Structure:

Field Name	Type	Null	Default
VehicleClass	varchar(12)	NO	NULL
NumPlate	varchar(10)	NO	NULL
Floor	Int	NO	NULL
Pillar	char(1)	NO	NULL
EntryTime	Datetime	NO	NULL
ExitTime	Datetime	YES	NULL
AmountPaid			

Hardware Requirements:

CPU	:	Core i5 or higher processor
RAM	:	1GB or higher
Hard Disk	:	1GB or higher
Keyboard	:	Required for data entry
Mouse	:	Not required
CD Drive	:	Not required
Monitor	:	VGA (800x600) or higher resolution monitor.

Software Requirements:

Operating System	:	Windows 7/8 or any higher
PYTHON Compiler	:	PYTHON
MYSQL	:	MYSQL installer Community MYSQL Connector Python MYSQL Web Installer Community

Installation Process:

The Rajiv Chowk Parking System software is very easy to install and use.

The following steps need to follow to use this software:

- 1)** First of all, install PYTHON software from Python.org
- 2)** Install MYSQL and Python Connector correctly from dev.mysql.com
- 3)** Copy all the .py files to PYTHON installation folder of C: drive.
- 4)** Open the project file named Database.py to create the MYSQL databases with all tables automatically
- 5)** Open the main project file named Parking_System.py to compile and run to execute.
- 6)** Follow the instructions and use main menu for necessary actions.

SAMPLE OUTPUT OF THE PROPOSED SYSTEM:

**Welcome To Rajiv Chowk
Parking**

Date:2021-09-05

Vehicle Number: DLAB1k8888

Vehicle Class: Four-wheeler

Entry Time: 11:08:26

Parking Space Allotted: 1A

Unique Code:220011

Parking Fee:

i)4 Wheeler: Rs100

ii)2Wheeler:Rs60

Note: Above fee is only for 60 minutes
and thereafter Rs 1 will be charged for
each extra minute

CODING:

i) File name: User.py

```
import mysql.connector
import datetime
import random

print(
    """
Welcome to Parking System
This system has facility of maintaining realtime Parking data
This system facilitates Entry/Exit of Vehicles and also stores data locally.
""")

db=mysql.connector.connect(user="root",password="admin",host="localhost",data
base="parking")
mycursordb=db.cursor()
now = datetime.datetime.now()
x = int(input("Enter 1:For Entering Parking zone 2:Leaving Parking zone
3:Misplaced Parking Slip  "))
if x == 1:
    print("Welcome to Rajiv Chowk")
    FName=input("Enter First Name  ")
    while FName.isalpha() != True:
        FName=input("Enter Again should be alphabets only" )
    LName = input("Enter Last Name")
    while LName.isalpha() != True:
        LName=input("Enter Again should be alphabets only")
    VehicleNum=input("Enter The Number Plate  ")
    while len(VehicleNum)<10:
        VehicleNum=input("Should be 10 Characters Enter again")
    mycursordb.execute("select count(*) from floor_1 where Reserved='NOT RES'")
    r1=mycursordb.fetchone()
```

```

mycursordb.execute("select count(*) from floor_2 where Reserved='NOT RES'")
r2=mycursordb.fetchone()
mycursordb.execute("select count(*) from floor_3 where Reserved='NOT RES'")
r3=mycursordb.fetchone()
if r1[0] != 0:
    mycursordb.execute("select Pillars from floor_1 where      Reserved ='NOT
RES' limit 3")
    print("\nOn floor 1 pillar nearest to you are\n")
    for i in mycursordb:
        stripi=str(i)
        print(stripi.strip("(",)+"'\t')
    pillarin=input("\nPlease choose a pillar: ").upper()
    pillar=pillarin[1]
    mycursordb.execute ("""
        UPDATE floor_1
        SET Reserved="RES"
        WHERE Pillars='%s'
        """ %(pillar))
    db.commit()

```

```

elif r2[0] !=0:
    print("\nSorry Floor 1 is fully Booked. Nearest Floor empty is 2\n")
    mycursordb.execute("select Pillars from floor_2 where Reserved ='NOT RES'
limit 3")
    print("\nOn floor 2 pillar nearest to you are\n")
    for i in mycursordb:
        stripi=str(i)
        print(stripi.strip("(",)+"'\n')
    pillarin=input("\nPlease choose a pillar: ").upper()
    pillar=pillarin[1]
    mycursordb.execute ("""
        UPDATE floor_2

```



```

        SET Reserved="RES"
        WHERE Pillars='%s'
        """ %(pillar))
    db.commit

elif r3[0] != 0:
    print("\nSorry Floor 1 and 2 are fully Booked. Nearest Floor empty is 3\n")
    mycursordb.execute("select Pillars from floor_3 where Reserved ='NOT RES'
limit 3")
    print("\nOn floor 3 pillar nearest to you are\n")
    for i in mycursordb:
        stripi=str(i)
        print(stripi.strip('(')+'\n')
    pillarin=input("\nPlease choose a pillar: ").upper()
    pillar=pillarin[1]
    mycursordb.execute ("""
        UPDATE floor_3
        SET Reserved="RES"
        WHERE Pillars='%s'
        """ %(pillar))
else:
    print("Sorry we are fully booked no parking space is available.")

temp=random.randint(1,1000000)
print("Your Unique Identification Code is",temp)
floor=1
EntryTime=now.strftime("%Y-%m-%d %H:%M:%S")
mycursordb.execute("insert into
logfile(Name,VehicleNum,Floor,Pillar,EntryTime,code) values
(%s,%s,%s,%s,%s,%s)" , (FName+'
'+LName,VehicleNum,floor,pillar,EntryTime,temp))
    db.commit()
if x==2:

```

```

a=int(input("Enter The Unique Code"))
b=input("Enter Number Plate")
mycursordb.execute("select code from logfiles where VehicleNum='%s'" % (b))
for i in mycursordb:
    d=i[0]
if a==d:
    Extime=now.strftime("%Y-%m-%d %H:%M:%S")
    mycursordb.execute("""
        UPDATE logfiles
        SET ExitTime=%s
        WHERE code=%s
        """, (Extime,a))
    db.commit()
    mycursordb.execute("select
TIMESTAMPDIFF(MINUTE,EntryTime,ExitTime)from logfiles where code =
'%s'%a)
    g=mycursordb.fetchall()
    for i in g:
        totaltime=i[0]
        print("Total Time Spent in parking is",totaltime,"minutes")
        totalamount=80
    if totaltime>60:
        extratime=totaltime-60
        totalamount=totalamount+extratime*1
        print("Amount to be Paid:Rs",amount)
        print("Have A Nice Day")
    elif totaltime<60:
        print("Amount to be Paid:Rs",totalamount)
        print("Have A Nice Day")
        mycursordb.execute ("""
            UPDATE logfiles
            SET AmountPaid=%s
            WHERE code=%s

```

```

        """, (totalamount,a))
    db.commit()
else:
    print("Incorrect code")
if x==3:
    num=input("Enter Vehicle Number")
    fullname=input("Enter full Name")
    mycursordb.execute("select Name from logfiles where VehicleNum = '%s'"
%num )
    p=mycursordb.fetchall()
    for i in p :
        z=i[0]
        if z==fullname:
            print("Record Found")
            mycursordb.execute("select code from logfiles where VehicleNum = '%s'"
% num)
            yz=mycursordb.fetchall()
            for i in yz:
                print("Your Code is:",i[0])
                print("Try exiting the parking again")
            else:
                print("No record found")
                print("Contact Authorities")
mycursordb.close()
mycursordb.close()
db.close()
db.close()

```

ii) File Name:Administration.py

```
#program to access the reports and other observatory options.
#requires the administrative access using an ID and a password.
import mysql.connector
from tabulate import tabulate

db=mysql.connector.connect(user="root",password="admin",host="localhost",data
base="parking")
mycursordb=db.cursor()
initial_count = 1
chances = 5
input_id = str(input("Enter the login id here: "))
password=input("Enter the Password")
mycursordb.execute("select ID from login where ID = '%s'" %(input_id))
idcheck=mycursordb.fetchall()
for i in idcheck:
    c=i[0]
mycursordb.execute("select Password from login where ID = '%s'" % (input_id))
passcheck=mycursordb.fetchall()
for a in passcheck:
    d=a[0]
if input_id==c and password==d:
    Options=int(input("Enter 1:For Tabular Report 2:For Revenue collected 3: No.
of cars 4: Get Log Files of Specified Date 5: Current status of Pillars 6: Cars
Present Inside Currently 7:Frequency Of a Car"))
    if Options==1:
        mycursordb.execute("select
VehicleNum,Floor,Pillar,EntryTime,ExitTime,code,AmountPaid from logfiles")
        result=mycursordb.fetchall()

print(tabulate(result,headers=["VehicleNum", "Floor", "Pillar", "EntryTime", "ExitTim
e", "Code", "Amount"],tablefmt="psql"))
    if Options==2:
```

```

mycursordb.execute("select SUM(AmountPaid)from logfiles")
total=mycursordb.fetchall()
for q in total:
    print("Total Revenue collected:",q[0])
if Options==3:
    mycursordb.execute("select COUNT(VehicleNum)from logfiles")
    count=mycursordb.fetchall()
    for w in count:
        print("Total Number of Cars:",w[0])
if Options==4:
    date=input("Enter the Date(YYYY/MM/DD)")
    mycursordb.execute("select
VehicleNum,Floor,Pillar,EntryTime,ExitTime,code,AmountPaid from logfiles
where cast(EntryTime as date)= '%s'"%(date))
    res=mycursordb.fetchall()

print(tabulate(res,headers=["VehicleNum","Floor","Pillar","EntryTime","ExitTime","
Code","Amount"],tablefmt="psql"))

if Options==5:
    floor=int(input("Enter Floor Number"))
    if floor==1:
        mycursordb.execute("select *from floor_1")
        floor_1=mycursordb.fetchall()
        print(tabulate(floor_1,headers=["Floor","Pillars"],tablefmt="psql"))

    if floor==2:
        mycursordb.execute("select *from floor_2")
        floor_2=mycursordb.fetchall()
        print(tabulate(floor_2,headers=["Floor","Pillars"],tablefmt="psql"))

    if floor==3:
        mycursordb.execute("select *from floor_3")
        floor_3=mycursordb.fetchall()

```

```

        print(tabulate(floor_3,headers=["Floor","Pillars"],tablefmt="psql"))
    if floor>3:
        print("Invalid Input")
    if Options==6:
        mycursordb.execute("select COUNT(VehicleNum) from logfiles where
AmountPaid=0")
        count1=mycursordb.fetchall()
        for z in count1:
            print("Total Number of Cars in Parking are:",z[0])

    if Options==7:
        Car=input("Enter Vehicle Registration Number")
        mycursordb.execute("select COUNT(VehicleNum) from logfiles where
VehicleNum= '%s'" % (Car))
        count2=mycursordb.fetchall()
        for xz in count2:
            print("Frequency Of Car is :",xz[0])
    else:
        print("Incorrect UserID or Password")
    if input_id in idcheck :
        input_pass = input("Enter the password here: ")
    db.close()

```

iii) **File Name:Database.py**

#This python program creates database for the system

```
import mysql.connector
```

```
db=mysql.connector.connect(user="root",password="admin",host="localhost")
mycursordb=db.cursor()
```

```
mycursordb.execute("create database parking")
```

```
mycursordb.execute("USE parking")
```

```
mycursordb.execute("create table floor_1 ( Pillars varchar(2), Reserved varchar(7)
Default 'NOT RES');")
```

```
mycursordb.execute("create table floor_2 ( Pillars varchar(2), Reserved varchar(7)
Default 'NOT RES');")
```

```
mycursordb.execute("create table floor_3 ( Pillars varchar(2), Reserved varchar(7)
Default 'NOT RES');")
```

```
mycursordb.execute("create table logfiles (Name varchar(255) not null,
VehicleNum varchar(10) not null, Floor int not null, Pillar char(1) not null,
EntryTime datetime not null,ExitTime datetime ,AmountPaid int DEFAULT
0,code int);")
```

```
mycursordb.execute("insert into floor_1 (Pillars) values
('A'),('B'),('C'),('D'),('E'),('F'),('G'),('H'),('I'),('J'),('K'),('L'),('M'),('N'),('O'),('P'),('Q'),('R'),('S')")
db.commit()
```

```
mycursordb.execute("insert into floor_2 (Pillars) values
('A'),('B'),('C'),('D'),('E'),('F'),('G'),('H'),('I'),('J'),('K'),('L'),('M'),('N'),('O'),('P')")
db.commit()
```

```
mycursordb.execute("insert into floor_3 (Pillars) values
('A'),('B'),('C'),('D'),('E'),('F'),('G'),('H'),('I'),('J'),('K'),('L'),('M')")
db.commit()
```

OUTPUT:

```
Welcome to Parking System
This system has facility of maintaining realtime Parking data
This system facilitates Entry/Exit of Vehicles and also stores data.
```

Snapshot of welcome screen

```
Enter 1:For Entrying Parking zone
Enter 2:Leaving Parking zone
Enter 3:Misplaced Parking Slip
```

Snapshot of Main Menu(User.py)

```
Welcome to Rajiv Chowk
Enter First Name  Khushal
Enter Last NameJhingan
Enter The Number Plate  DL2CAK8997

On floor 1 pillar nearest to you are

C
D
E

Please choose a pillar: C
Your Unique Identification Code is 831871
```

Snapshot of Vehicle Entry Screen


```
Enter The Unique Code831871
Enter Number Platedl2cak8997
Total Time Spent in parking is 2 minutes
Amount to be Paid:Rs 80
Have A Nice Day
```

Snapshot of Vehicle Exit Screen

```
Enter Vehicle NumberDL2CAK8994
Enter full NameVarun Marwah
Record Found
Your Code is: 214774
Try exiting the parking again
```

Snapshot of Misplaced Parking Slip

```
Welcome to Rajiv Chowk Parking Administrative system
Enter the login id here: Khushal
Enter the Passwordkps1234
```

Snapshot of Login Screen for Administrative system

```
Enter 1:For Tabular Report
Enter 2:For Revenue collected
Enter 3: No. of cars
Enter 4: Get Log Files of Specified Date
Enter 5: Current status of Pillars
Enter 6: Cars Present Inside Currently
Enter 7:Frequency Of a Car|
```

Snapshot of Main Menu(Administration.py)

VehicleNum	Floor	Pillar	EntryTime	ExitTime	Code	Amount
dl2cak8997	1	B	2021-09-13 16:48:00		9453	0
dl2cak8991	1	B	2021-09-13 16:48:42		909323	0
dl2cak8997	1	A	2021-09-13 16:49:00		757035	0
DL2CAK8997	1	B	2021-09-08 17:01:22	2021-09-13 17:01:27	654794	80
DL1AC88882	1	B	2021-09-13 16:49:34	2021-09-13 16:52:51	685413	80
dl2cak8997	1	B	2021-09-14 10:49:30		406118	0
DL2CAK8997	2	B	2021-09-14 11:20:50		107475	0
dl2cak8997	2	A	2021-10-08 10:09:50		282426	0

Snapshot of Tabular Report

Total Revenue Collected:Rs160

Snapshot of Revenue Collected

Total No. of Cars:8

>>>

Snapshot of Total Number of cars

```
Enter the Date(YYYY/MM/DD)2021/09/13
```

VehicleNum	Floor	Pillar	EntryTime	ExitTime	Code	Amount
dl2cak8997	1	B	2021-09-13 16:48:00		9453	0
dl2cak8991	1	B	2021-09-13 16:48:42		909323	0
dl2cak8997	1	A	2021-09-13 16:49:00		757035	0
DL1AC88882	1	B	2021-09-13 16:49:34	2021-09-13 16:52:51	685413	80

Snapshot of log files on specified date

```

Total Number of Cars in Parking currently are:6
>>>

```

Snapshot of No. of cars present inside parking currently

Enter Floor Number1	
Floor	Pillars
A	RES
B	RES
C	RES
D	NOT RES
E	NOT RES
F	RES
G	NOT RES
H	NOT RES
I	NOT RES
J	NOT RES
K	NOT RES
L	NOT RES
M	NOT RES
N	NOT RES
O	NOT RES
P	NOT RES
Q	NOT RES
R	NOT RES
S	NOT RES

Snapshot of Current Status of Pillars

Enter Vehicle Number:DL2CAK8997
Frequency of Car:6

Snapshot of Frequency of a Vehicle screen