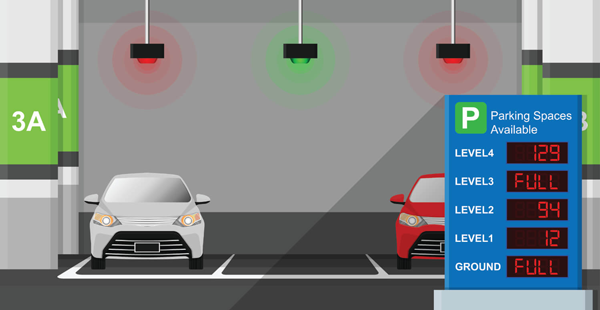
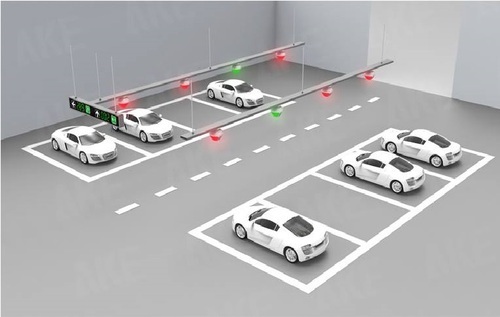


**PYTHON PROJECT**

**PARKING MANAGEMENT SYSTEM**



SUBMITTED BY- Mr.Alok Patil

Mr.Mohit Patel

**Acknowledgement**

I ,…………………………… of class XIITH A roll no………………

would like to express my sincere gratitude to my computer science teacher Ms.APARNA DHIRDE,PGT COMPUTER SCIENCE, for her vital support, guidance and encouragement – without which this project would not have come forth.

I would also like to express my gratitude to my school KENDRIYA VIDYALAYA AMBARNATH for letting me use the school laboratory.

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr. No | Particular | Page No. |
| 1 | Brief overview of project |  |
| 2 | Need of computerization |  |
| 3 | Software and hardware requirement |  |
| 4 | Advantages of project |  |
| 5 | Limitations of project |  |
| 6 | Source code of project |  |
| 7 | Output screening |  |
| 8 | Future enhancement of project |  |
| 9 | Bibliography |  |

**Introduction**

This is a project based on Parking Management. The program helps us to enter, display or alter the details of vehicles in parking records.

Moreover & most importantly the program helps us The program also helps us to know the present status of a parking detail, vehicle detail etc.

It includes various function programs to do the above mentioned tasks.

Data file handling has been effectively used in the program.

The database is a collection of interrelated data to serve multiple applications. That is database programs create files of information. So we see that files are worked with most, inside the program.

DBMS

The software required for the management of data is called as DBMS. It has 3 models

• Relation model

• Hierarchical model

• Network model

RELATIONAL MODEL It’s based on the concept on relation. Relation is the table that consists of rows and columns. The rows of the table are called tuple and the columns of the table are called attribute. Numbers of rows in the table is called as cardinality. Number of columns in the table is called as degree.

HIERARCHICAL MODEL: In this type of model, we have multiple records for each record. A particular record has one parent record. No chide record can exist without parent record. In this, the records are organized in tree (like structure

NETWORK MODEL:- In this, the data is represented by collection of records and relationship is represented by (ink or association.

CHARACTERISTICS OF DB MS: -

• It reduces the redundancy

• Reduction of data in inconsistency

• Data sharing

• Data standardization

DIFFERENT TYPES OF FILES: -BASED ON ACCESS:-

• Sequential file

• Serial file

• Random (direct access) file BASED ON STORAGE:-

• Text file

• Binary File

**Need Of Computerization**

****

No one can easily plan the journey comfortably as the process is efficient and fast with being easy to access. Reservations can be made through the Indian railways site or at the ample reservation centers all over the country. Also now there are authorized agencies which provide reservation facility on behalf of India railways and without waiting in long line one can easily book a ticket. The booking is done through an E- Ticket issue which has a PNR number of which one has to take a print and just have to show at the station.

It not only provides reservation but cancellation can also be done through this system at ease and one can use a credit card to complete the process. This being a big step in terms of improvement in the railway system it is widely accepted across the country.

**Software & Hardware Requirement**

**SOFTWARE SPECIFICATION: -**

**Operating System: Windows 7**

**Platform: Python IDLE 3.7**

**Database: MySQL**

**Language: Python**

**HARDWARE SPECIFICATION: -**

**Processor: Dual Core & Above**

**Hard Disk: 40GB**

**RAM: 1024MB**

**Note: For Python-MySQL connectivity, following data have been used:-**

**Host- localhost, user- root, password- root, database- school**

**Advantages Of Project**

* There is a greater sense of security due to the fact that patrons do not actually walk to and from their own space.
* It is highly feasible for extremely small sites that are unable to accommodate a conventional ramped parking structure.
* There is high parking efficiency (i.e. sf/space and cf/space).
* There is no need for driving while looking for an available space.
* Emissions are greatly brought down and reduced.
* The patrons wait for their car in a highly controlled environment.
* There are less chances for vehicle vandalism.
* There is a minimal staff requirement if it is used by known parkers.
* It is possible that the retrieval time is lower than the combined driving/parking/walking time in conventional ramped parking structures.  
  There is an easier facade integration since there are no ramping floors or openings in exterior walls.

**Limitations Of Project**

* There is a greater construction cost per space (but this may be offset by the chance for lesser land costs per space and the system manufacturers say that the operating and maintenance cost will be lower as compared to a conventional ramped parking structure).
* Use of redundant systems will result in a greater cost.
* It may be a bit confusing for unfamiliar users.
* It is not recommended for high peak hour volume facilities.
* There may be a fear of breakdown (How do I get my car out?).
* There is an uncertain building department review and approval process.
* It requires a maintenance contract with the supplier.

**Source Code Of Project**

import os

import platform

import mysql.connector

mydb=mysql.connector.connect(host="localhost",user="root",password="root",database='parking')

mycursor=mydb.cursor()

defAdd\_Record():

L=[]

id1=int(input("Enter the parking number : "))

L.append(id1)

pname1=input("Enter the Parking Name: ")

L.append(pname1)

level1=input("Enter level of parking : ")

L.append(level1)

freespace1=input("Is there any freespace or not :YES/NO ")

L.append(freespace1)

vehicleno1=input("Enter the Vehicle Number : ")

L.append(vehicleno1)

nod1=int(input("Enter total number of days for parking: "))

L.append(nod1)

if nod1==1:

Payment1=20

elif nod1==2:

Payment1=40

elif nod1==3:

Payment1=60

elif nod1==4:

Payment1=80

elif nod1==5:

Payment1=100

elif nod1==6:

Payment1=120

L.append(Payment1)

stud=(L)

sql='insert into parkmaster12(pid,pnm,level,freespace,vehicleno,nod,payment) values(%s,%s,%s,%s,%s,%s,%s)'

mycursor.execute(sql,stud)

mydb.commit()

defRec\_View():

print("Select the search criteria : ")

print("1. Parking Number")

print("2. Parking Name")

print("3. Level No")

print("4. All")

ch=int(input("Enter the choice : "))

if ch==1:

s=int(input("Enter Parking no : "))

rl=(s,)

sql="select \* from parkmaster12 where pid=%s"

mycursor.execute(sql,rl)

res=mycursor.fetchall()

elifch==2:

s=input("Enter Parking Name : ")

rl=(s,)

sql="select \* from parkmaster12 where pnm=%s"

mycursor.execute(sql,rl)

res=mycursor.fetchall()

elifch==3:

s=int(input("Enter Level of Parking : "))

rl=(s,)

sql="select \* from parkmaster12 where level=%s"

mycursor.execute(sql,rl)

res=mycursor.fetchall()

elifch==4:

sql="select \* from parkmaster12"

mycursor.execute(sql)

res=mycursor.fetchall()

print("Details about Parking are as follows : ")

print("(Parking Id,ParkingName,Level,FreeSpace(Y/N),Vehicle No,No of days for parking,Payment)")

for x in res:

print(x)

print('Task comple ted')

defVehicle\_Detail():

L=[]

vid1=int(input("Enter Vehicle No : "))

L.append(vid1)

vnm1=input("Enter Vehicle Name/Model Name : ")

L.append(vnm1)

dateofpur1=input("Enter Year-Month-date of purchase : ")

L.append(dateofpur1)

vdt=(L)

sql="insert into vehicle(pid,vnm,dateofpur) values(%s,%s,%s)"

mycursor.execute(sql,vdt)

mydb.commit()

defVehicle\_View():

vid1=int(input("Enter the vehicle number of the vehicle whose details is to be viewed : "))

sql='select parkmaster12.pid,parkmaster12.pnm,parkmaster12.vehicleno, vehicle.pid,vehicle.vnm from parkmaster12 INNER JOIN vehicle ON parkmaster12.pid=vehicle.pid and vehicle.pid=%s'

rl=(vid1,)

print('The following are the detailes you wanted:')

mycursor.execute(sql,rl)

res=mycursor.fetchall()

for x in res:

print(x)

print('Task compelted')

defremove():

vid1=int(input("Enter the vehicle number of the vehicle to be deleted : "))

rl=(vid1,)

sql="Delete from vehicle where pid=%s"

mycursor.execute(sql,rl)

mydb.commit()

print('Removed as per the command')

defMenu():

print("Enter 1 : To Add Parking Detail")

print("Enter 2 : To View Parking Detail ")

print("Enter 3 : To Add Vehicle Detail ")

print("Enter 4 : To Remove Vehicle Record")

print("Enter 5 : To see the details of Vehicle")

input\_dt = int(input("Please Select An Above Option: "))

if(input\_dt== 1):

Add\_Record()

elif (input\_dt==2):

Rec\_View()

elif (input\_dt==3):

Vehicle\_Detail()

elif (input\_dt==4):

remove()

elif (input\_dt==5):

Vehicle\_View()

else:

print("Enter correct choice....")

Menu()

defrunAgain():

runAgn=input('\nwant to run Again Y/n:')

while(runAgn.lower()=='y'):

if(platform.system()=='Windows'):

print(os.system('cls'))

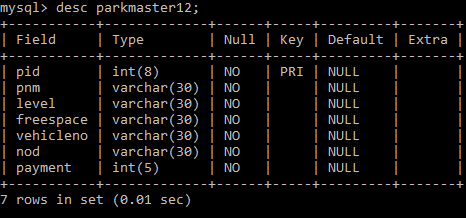
else:

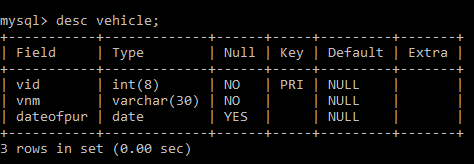
print(os.system('clear'))

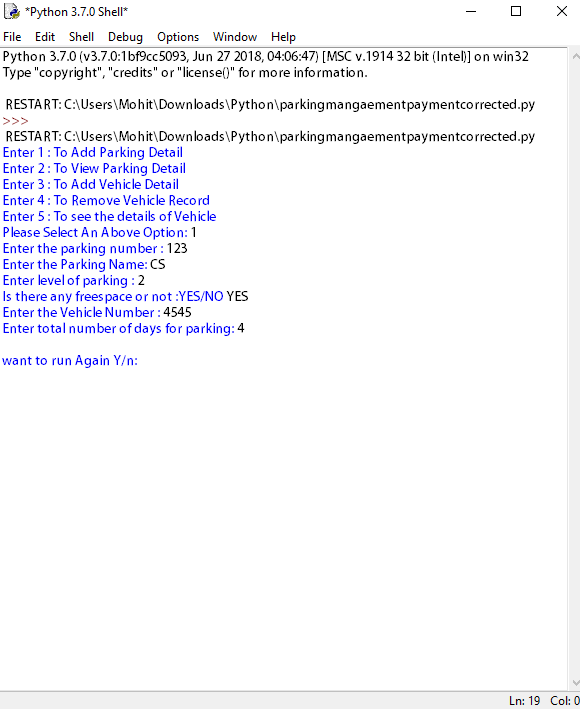
Menu()

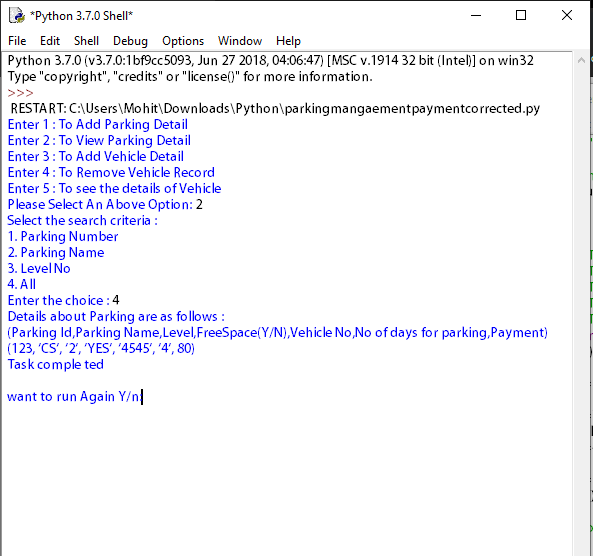
runAgn=input('\nwant to run Again Y/n:') runAgain()

**Output Screening**

****

****

****

****

**Future Enhancement Of Project**

The smart parking industry continues to evolve as an increasing number of cities struggle with traffic congestion and inadequate parking availability. While the deployment of sensor technologies continues to be core to the development of smart parking, a wide variety of other technology innovations are also enabling more adaptable systems—including cameras, wireless communications, data analytics, induction loops, smart parking meters, and advanced algorithms.

The future of the smart parking market is expected to be significantly influenced by the arrival of automated vehicles (AVs). Several cities around the world are already beginning to trial self-parking vehicles, specialized AV parking lots, and robotic parking valets.

**Bibliography**

[www.google.com](http://www.google.com)

[https://pythontrends.wordpress.com](https://pythontrends.wordpress.com/)

BOOK – Python by Sumita Arora

CERTIFICATE

CLASS: Xii-a YEAR:2019-20

**This is to certify that Investigatory Project is**

**successfully completed by …..………………… of Class:XII ,Division: A .Roll no. : ……………….. for the academic year 2019-2020 in the School Computer lab.**

**Head Teacher External Internal Examiner Signature: Examiner (Subject Teacher)**

**Date: / / 20 Department of COMPUTER SCIENCE**

**PRINCIPAL**