Parking Space Management System Project Documentation

**Overview**

The Parking Space Management System is a software application developed to efficiently manage parking space records. It provides functionalities for adding new parking details, viewing existing parking records, removing records, and exiting the program. The system is designed to interact with a MySQL database to store and retrieve parking data.

**Purpose**

The purpose of this project is to create a user-friendly interface for managing parking space records effectively. By automating the process of recording parking details and providing easy access to this information, the system aims to streamline parking management operations and improve overall efficiency.

**Scope**

The scope of the project includes:

* Adding new parking records with details such as slot number, name, vehicle number, mobile number, date, time, and duration of parking.
* Viewing existing parking records based on slot number, mobile number, or vehicle number.
* Removing parking records based on vehicle number.
* Exiting the program gracefully.

**2. Project Objectives**

* Develop a user-friendly interface for interacting with the parking management system.
* Implement functionalities for adding, viewing, and removing parking records.
* Ensure seamless integration with a MySQL database for data storage and retrieval.
* Test the system thoroughly to ensure reliability and accuracy.
* Provide comprehensive documentation for users and developers.

**3. Project Dependencies**

The project dependencies include:

* Python programming language
* MySQL database server
* MySQL Connector/Python library

**4.Project Code**

import mysql.connector

mydb=mysql.connector.connect(host="127.0.0.1",user="root",password="root",database='parkingproject')

mycursor=mydb.cursor()

print(mydb)

def Menu():

"""

Use Case:

Displays a menu of options for the user to choose from and calls the corresponding function

based on the user's input.

"""

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

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

print("Enter 3 : To Remove Parking Record")

print("Enter 4 : To exit")

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:

remove()

elif input\_dt== 4:

exit()

else:

print("Invalid option entered...")

runAgain()

def Add\_Record():

"""

Use Case:

Allows users to add parking details such as slot number, name, vehicle number, mobile number,

date, time, and duration of parking. Calculates the payment based on the number of days and

inserts the record into the MySQL database.

"""

L =[]

slot=int(input("Enter the parking Slot number : "))

L.append(slot)

name=str(input("Enter Your Name: "))

L.append(name)

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

L.append(vehicleno)

Mobile=int(input("Enter Your PhoneNumber:"))

L.append(Mobile)

date=input("Enter the date your are parking in the format of dd-mm-yyyy:")

L.append(date)

time=input("Enter the Current Time in the format of HH:MM :")

L.append(time)

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

L.append(nod)

payment = 20\*nod

print("Payment: ",payment)

L.append(payment)

sql='insert into parkingspace(slot ,name ,vehicleno , Mobile, date, time, nod, payment) values(%s,%s,%s,%s,%s,%s,%s,%s)'

mycursor.execute(sql,L)

mydb.commit()

del(L)

if\_record = input("If you want to add more parking details(yes/no):")

if if\_record.lower()=='yes':

Add\_Record()

else :

print("Thank You")

def Rec\_View():

"""

Use Case:

Allows users to view parking details based on different search criteria such as slot number,

mobile number, or vehicle number. Retrieves the records from the database and displays them.

"""

print("Select the search criteria : ")

print("1. Slot Number")

print("2. Your Mobile Number")

print("3. Your Vehicle Number")

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

if ch==1:

s=int(input("Enter Parking Slot ID : "))

rl=[s]

sql="select \* from parkingspace where slot=%s"

mycursor.execute(sql,rl)

res=mycursor.fetchall()

elif ch==2:

s=input("Enter Your Mobile Number : ")

rl=[s]

sql="select \* from parkingspace where Mobile=%s"

mycursor.execute(sql,rl)

res=mycursor.fetchall()

elif ch==3:

s=int(input("Enter Your Vehicle Number : "))

rl=[s]

sql="select \* from parkingspace where vehicleno=%s"

mycursor.execute(sql,rl)

res=mycursor.fetchall()

for x in res:

print('Parking Slot Id :',x[0])

print('Your Name :',x[1])

print('Vehicle Number :',x[2])

print('Mobile Number :',x[3])

print('Date when you Parked :',x[4])

print('Time when you Parked :',x[5])

print('Number of days for parking :',x[6])

print('Payment :',x[7])

print('Thank You')

rl.pop()

if\_view = input("If you want to view more parking details(yes/no):")

if if\_view.lower()=='yes':

Rec\_View()

def remove():

"""

Use Case:

Allows users to remove parking records based on the vehicle number. Deletes the record from

the MySQL database.

"""

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

Y = [vid]

sql="Delete from parkingspace where vehicleno=%s"

mycursor.execute(sql,Y)

mydb.commit()

print('The Vehicle is gone from Parking space')

Y.pop()

if\_remove = input("If you want to delete more vehicle details(yes/no):")

if if\_remove.lower()=='yes':

remove()

def exit():

"""

Use Case:

Closes the cursor and the database connection when the user chooses to exit the program.

"""

mycursor.close()

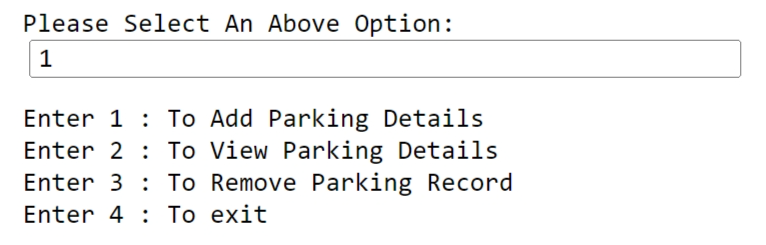
mydb.close()

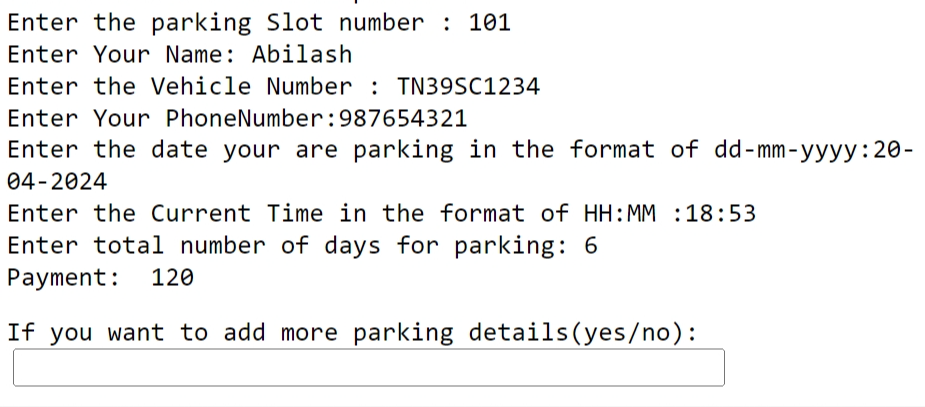
print("Thankyou!!!")

Menu()

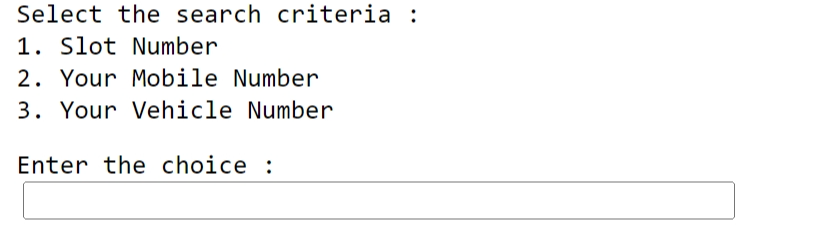
**Output Screenshot :**

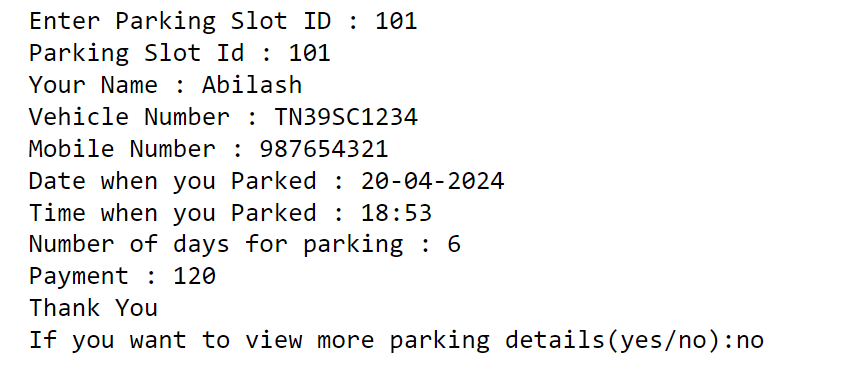
**To Add Parking Details:**



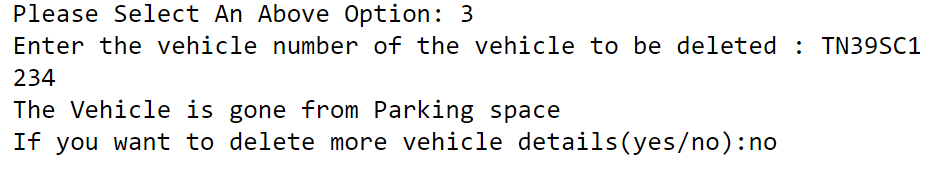


**To View Parking Details:**

****

****

**To Remove Parking Record:**

****