**IPAD SYSTEM MANGEMENT**

***DONE BY - STEIVE JAMES S***

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **S NO** | **DESCRIPTION** |
| 1 | INTRODUCTION TO PYTHON |
| 2 | INTRODUCTION TO MYSQL (RDBMS) |
| 3 | PROJECT SYNOPSIS |
| 4 | MODULES AND DESCRIPTIONS |
| 5 | PYTHON SOURCE CODE |
| 6 | SCREENSHOTS OF EXECUTION |
| 7 | SYSTEM REQUIREMENTS |
| 8 | BIBLIOGRAPHY |

**INTRODUCTION**

**TO**

**PYTHON**



**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; make it very attractive for Rapid Application Development, as well as for use as a scripting language to connect existing components together. Python's simple and easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**Python's features include −**

* **Easy to code:** Python is a high-level programming language and it is very easy to code. It is also a developer-friendly language.
* **Free and Open Source:** Python is freely available. You can download it from the [Python Official Website](https://www.python.org/downloads/). Secondly, it is open-source. This means that its source code is available to the public. You can download it, change it, use it, and distribute it.
* **Object-Oriented Language:** One of the key features of python is Object-Oriented programming. Python supports object-oriented language and concepts of classes, objects, modularization etc.
* **GUI Programming Support:** Graphical User interfaces can be made using a module such as PyQt5, PyQt4, wxPython, or Tk in python.
* **High-Level Language:** Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.
* **Extensible feature:** If needed, you can write some of your Python code in other languages like**C++**. This makes Python an extensible language, meaning that it can be extended to other languages.
* **Python is Portable language:** Python language is also a portable language. For example, if we have python code for windows and if we want to run this code on other platforms such as Linux, Unix, and Mac then we do not need to change it, we can run this code on any platform.
* **Python is Integrated language:** Python is also an Integrated language because we can easily integrated python with other languages like c, c++, etc.
* **Interpreted Language:** Python is an Interpreted Language because Python code is executed line by line at a time. Like other languages C, C++, Java, etc. there is no need to compile entire python code, this makes it easier to debug the code.
* **Large Standard Library**: Python has a large standard library which provides a rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, web browsers, etc.
* **Dynamically Typed Language:** Python is dynamically-typed. This means that the type for a value is decided at runtime, not in advance. This is why we don’t need to specify the type of data while declaring it.

**INTRODUCTION**

**TO**

**MYSQL (RDBMS)**



A **Database** is a collection of information related to a particular subject or purpose, such as tracking customer orders or maintaining a product collection. Using any RDBMS application software like MS SQL Server, MySQL, Oracle, Sybase etc, all information can be managed from a single database file. Within the file, data can be divided into separate storage containers called tables. Data can be retrieved using queries.

A table is a collection of data about a specific topic, such as products or suppliers. Using a separate table for each topic means you can store that data only once, which makes your database more efficient and reduces data-entry errors. Table organises data into columns (called fields) and rows (called records).

A **Primary key** is one or more fields whose value or values uniquely identify each record in a table. In a relationship, a primary key is used to refer to specific record in one table from another table. A foreign key is a column in a relational database table that provides a link between data in two tables. It acts as a cross-reference between tables because it references the primary key of another table, thereby establishing a link between them.

### Role of RDBMS Application Program:

A computer database works as a electronic filing system, which has a large number of ways of cross-referencing, and this allows the user many different ways in which to re-organize and retrieve data. A database can handle business inventory, accounting and filing and use the information in its files to prepare summaries, estimates and other reports. The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available DBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase. A database management system, therefore, is a combination of hardware and software that can be used to set up and monitor a database, and can manage the updating and retrieval of database that has been stored in it. Most of the database management systems have the following capabilities:

* **Creating of a table, addition, deletion, modification of records.**
* **Retrieving data collectively or selectively.**
* **The data stored can be sorted or indexed at the user's discretion and direction.**
* **Various reports can be produced from the system. These may be either standardized report or that may be specifically generated according to specific user definition.**
* **Mathematical functions can be performed and the data stored in the database can be manipulated with these functions to perform the desired calculations.**
* **To maintain data integrity and database use.**

The management of data in a database system is done by means of a general-purpose software package called a Database Management System (DBMS). Some commercially available RDBMS are MS SQL Server, MS ACCESS, INGRES, ORACLE, and Sybase. MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

**The Main Features of MySQL**

* MySQL is written in C and C++.
* **Easy to use:** MySQL is easy to use. We have to get only the basic knowledge of SQL. We can build and interact with MySQL by using only a few simple SQL statements.
* **Secure:** It consists of a solid data security layer that protects sensitive data from intruders. Also, passwords are encrypted in MySQL.
* **Client/ Server Architecture:** It follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they can query data, save changes, etc.
* **Free to download:** It is free to use so that we can download it from MySQL official website without any cost.
* **Speed:** It is considered to be one of the very fast database languages, backed by a large number of the benchmark test.
* **High Flexibility:** It supports a large number of embedded applications, which makes it very flexible.
* **Compatible on many operating systems:** It is compatible to run on many operating systems, like Novell NetWare, Windows, Linux, many varieties of UNIX etc.
* **Allows roll-back:** It allows transactions to be rolled back, commit, and crash recovery.
* **Memory efficiency:** It’s efficiency is high because it has a very low memory leakage problem.
* **High Performance:** It is faster, more reliable, and cheaper because of its unique storage engine architecture. It provides very high-performance results in comparison to other databases without losing an essential functionality of the software. It has fast loading utilities because of the different cache memory.
* **High Productivity:** MySQL uses Triggers, Stored procedures, and views that allow the developer to give higher productivity.
* **Platform Independent:** It can be downloaded, installed, and executed on most of the available operating systems.
* **Supports large databases:** MySQL Server can be used with databases that contain 50 million records. There are users who use MySQL Server with 200,000 tables and about 50,000,000 rows.

**PROJECT SYNOPSIS**

IPAD System

1. **Introduction** :

This project is aimed at developing a software application that depicts ipad system management for managing their subscribers list. Using this software, companies can improve the efficiency of those services.

1. **Objective of the Project:**

This software helps customers to find Payment Updates easily and Administrator to check the List of Subscribers .The software will help in easy Maintaining and Updating Subscribers Details with their Payment Details, website for the administrator, also for quick and easy identification the outstanding bill of the subscribers.

1. **Scope of the Project:**

This system will reduce the Manual Operation required to maintain all the Records of Subscriber’s Information. And also generates the various reports for Analysis. Main concept of the project is to enter Transaction Reports and to maintain Subscriber’s Records.

**MODULES**

**AND**

**DESCRIPTION**

**The Modules used in this Software are as follows:**

1. **ADMIN :** For accessing this module you need to fill the password correctly. This module has 2 options - Subscriber Login and Payment Login . We can create , modify , search and delete Subscriber and Payment details in this login.
2. **USER:** This module is for the Users who can access only Payment Login. Here the user is allowed to create, modify and search payment details. This login doesn’t need password. But this module can’t delete any records.
3. **LASTDAY:** This module is used to search the subscriber’s Plan details and fetch the details of the Subscribers with Outstanding bills every month end.

**PYTHON**

**SOURCE CODE**



SOURCE CODE

import mysql.connector as mc

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

if s.is\_connected():

print("Connected Sucessfully")

cur.execute("create table if not exists Subscriber(SubscriberNo varchar(25) primary key,Name varchar(25) not null,Mobile varchar(25) not null,ProductType varchar(25) not null,CreditCardNo varchar(25) not null,Address varchar(100) not null,State varchar(25) not null,EmailId varchar(25) not null,Scheme varchar(25) not null,FacilitiesReq varchar(25) not null,Validity date not null,BillReceivingMode varchar(25) not null,PaymentMode varchar(25) not null,Status varchar(25) not null)")

cur.execute("create table if not exists Payment(BillNo varchar(25) not null,BillingDate date not null,SubscriberNo varchar(25) primary key,Name varchar(25) not null,Mobile varchar(25) not null,ProductType varchar(25) not null,Recharge varchar(25) not null,DueDate date not null,Dateofpayment date not null,PaidAmount varchar(25) not null,DueTransaction varchar(25) not null,TransactionStatus varchar(25) not null)")

s.commit()

def Add\_Subs():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a=int(input("Enter Subscriber Number:"))

b=input("Enter Subscriber Name:")

c=int(input("Enter Mobile no.:"))

d=input("Enter Product Type [Prepaid/Postpaid]:")

e=input("Enter Credit card No.:")

f=input("Enter Address:")

h=input("Enter State:")

i=input("Enter E-Mail:")

j=input("Enter Scheme/Plan:")

k=input("Enter Facilities required [SMS/STD/International/Conferencing/Forwarding]:")

l=input("Enter Validity:")

n=input("Enter Bill receiving mode [Post/SMS/E-Mail]:")

o=input("Enter Payment mode [Cash/DD/Online]:")

p=input("Enter Status:")

q0="insert into Subscriber values('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(a,b,c,d,e,f,h,i,j,k,l,n,o,p)

cur.execute(q0)

s.commit()

print("---Record Saved---")

def Add\_Pay():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a0=input("Enter Billing Date:")

a1=int(input("Enter Bill No:"))

a=int(input("Enter Subscriber Number:"))

b=input("Enter Subscriber Name:")

c=int(input("Enter Mobile no.:"))

d=input("Enter Product Type [Prepaid/Postpaid]:")

m1=int(input("Enter Ammount to be Recharge:"))

o1=input("Enter Due Date:")

o2=input("Enter Date of Payment:")

o3=int(input("Enter Paid Amount:"))

o4=input("Enter Due Transaction:")

p=input("Enter Transaction Status:")

q1="insert into Payment values('{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(a1,a0,a,b,c,d,m1,o1,o2,o3,o4,p)

cur.execute(q1)

s.commit()

print("---Record Saved---")

def Modify\_Subs():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a=input("Enter Subscriber Number:")

cur.execute("select \* from Payment where SubscriberNo='{}'".format(a))

d=cur.fetchall()

if d==[]:

print("Sorry No Record Found!..")

else:

a1=input("Enter Product Type:")

a2=input("Enter Scheme:")

a3=input("Enter Facilities Required:")

a4=input("Enter Validity:")

a6=input("Enter Bill Received Mode:")

a7=input("Enter Payment Mode:")

a8=input("Enter Status:")

q2="update Subscriber set ProductType='{}',Scheme='{}',FacilitiesReq='{}',Validity='{}',BillReceivingMode='{}',PaymentMode='{}',Status='{}' where SubscriberNo='{}'".format(a1,a2,a3,a4,a6,a7,a8,a)

cur.execute(q2)

s.commit()

print("---Record Updated---")

def Modify\_Pay():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a=input("Enter Subscriber Number:")

cur.execute("select \* from Payment where SubscriberNo='{}'".format(a))

d=cur.fetchall()

if d==[]:

print("Sorry No Record Found!..")

else:

a1=input("Enter Product Type:")

a2=input("Enter Recharge:")

a3=input("Enter Due Amount:")

a4=input("Enter Transaction Status:")

q2="update Payment set ProductType='{}',Recharge='{}',DueTransaction='{}',TransactionStatus='{}' where SubscriberNo='{}'".format(a1,a2,a3,a4,a)

cur.execute(q2)

s.commit()

print("---Record Updated---")

def Delete\_Subs():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a=input("Enter Subscriber No to Delete:")

cur.execute("select \* from Subscriber where SubscriberNo='{}'".format(a))

d=cur.fetchall()

if d==():

print("Sorry No Record Found!..")

else:

q9="delete from Subscriber where SubscriberNo='{}'".format(a)

cur.execute(q9)

s.commit()

print("---Record Deleted---")

def Delete\_Pay():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a=input("Enter Subscriber No to delete:")

cur.execute("select \* from Payment where SubscriberNo='{}'".format(a))

d=cur.fetchall()

if d==():

print("Sorry No Record Found!..")

else:

q9="delete from Payment where SubscriberNo='{}'".format(a)

cur.execute(q9)

s.commit()

print("---Record Deleted---")

def Search\_Subs():

def name():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a=input("Enter Subscriber Name to search:")

q5="select \* from Subscriber where Name='{}'".format(a)

cur.execute(q5)

d=cur.fetchall()

if d==[]:

print("Sorry No Record Found!..")

else:

print(d)

def mobile():

s=mc.connect(host="localhost",user="root",passwd=” ",database="ios")

cur=s.cursor()

a5=input("Enter Mobile No to Search:")

q4="select \* from Subscriber where Mobile='{}'".format(a5)

cur.execute(q4)

d=cur.fetchall()

if d==[]:

print("Sorry No Record Found!..")

else:

print(d)

def subsno():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a3=input("Enter Subscriber No to Search:")

q2="select \* from Subscriber where SubscriberNo='{}'".format(a3)

cur.execute(q2)

d=cur.fetchall()

if d==[]:

print("Sorry No Record Found!..")

else:

print(d)

while True:

print("\n")

print("\*\*\*WEOLCOME TO SUBSCRIBER SEARCH\*\*\*")

print("\n")

print("1--->To Search on Name")

print("2--->To Search on Subscriber No")

print("3--->To Search on Mobile No")

print("4--->Exit")

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

if ch==1:

name()

elif ch==2:

subsno()

elif ch==3:

mobile()

elif ch==4:

break

else:

print("invalid choice")

def Search\_Pay():

def subsno():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a3=input("Enter Subscriber No to Search:")

q2="select \* from Payment where SubscriberNo='{}'".format(a3)

cur.execute(q2)

d=cur.fetchall()

if d==[]:

print("Sorry No Record Found!..")

else:

print(d)

def billno():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a1=input("Enter Bill No to Search:")

q1="select \* from Payment where BillNo='{}'".format(a1)

cur.execute(q1)

d=cur.fetchall()

if d==[]:

print("Sorry No Record Found!..")

else:

print(d)

def mobile():

s=mc.connect(host="localhost",user="root",passwd=" ",database="ios")

cur=s.cursor()

a5=input("Enter Mobile No to Search:")

q4="select \* from Payment where Mobile='{}'".format(a5)

cur.execute(q4)

d=cur.fecthall()

if d==[]:

print("Sorry No Record Found!..")

else:

print(d)

while True:

print("\n")

print("\*\*\*WEOLCOME TO PAYMENTS SEARCH\*\*\*")

print("\n")

print("1--->To Search on Billno")

print("2--->To Search on Subscriber No")

print("3--->To Search on Mobile No")

print("4--->Exit")

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

if ch==1:

billno()

elif ch==2:

subsno()

elif ch==3:

mobile()

elif ch==4:

break

else:

print("invalid choice")

while True:

print("\n")

print("\*\*\*\*WELCOME TO IPAD SYSTEM MANAGEMENT\*\*\*\*")

print("\n")

print("1-->Administrator")

print("2-->User")

print("3-->Exit")

ch=int(input("Enter your Choice:"))

if ch==1:

ad=int(input("Enter Pin:"))

p=12345

if ad==p:

while True:

print("\n")

print("\*\*\*\*WELCOME TO ADMIN\*\*\*\*")

print("\n")

print("1-->Subscriber Details")

print("2-->Payment Details")

print("3-->Exit")

c=int(input("Enter your Choice:"))

if c==1:

print("\n")

print("\*\*\*\*WELCOME TO SUBSCRIBER MODULE\*\*\*\*")

print("\n")

print("1-->Add Subscriber")

print("2-->Modify Subscriber")

print("3-->Delete Subscriber")

print("4-->Search Subscriber")

print("5-->Exit")

c=int(input("Enter your Choice:"))

if c==1:

Add\_Subs()

elif c==2:

Modify\_Subs()

elif c==3:

Delete\_Subs()

elif c==4:

Search\_Subs()

elif c==5:

break

else:

print("Invalid Choice")

elif c==2:

print("\n")

print("\*\*\*\*WELCOME TO PAYMENT MODULE\*\*\*\*")

print("\n")

print("1-->Add Payment")

print("2-->Modify Payment")

print("3-->Delete Payment")

print("4-->Search Payment")

print("5-->Exit")

c=int(input("Enter your Choice:"))

if c==1:

Add\_Pay()

elif c==2:

Modify\_Pay()

elif c==3:

Delete\_Pay()

elif c==4:

Search\_Pay()

elif c==5:

break

else:

print("Invalid Choice")

elif c==3:

break

else:

print("Invalid Choice")

else:

print("Access Denied - Admin Only...")

elif ch==2:

while True:

print("\n")

print("\*\*\*\*WELCOME TO USER\*\*\*\*")

print("\n")

print("1-->Payment Details")

print("2-->Exit")

c=int(input("Enter your Choice:"))

if c==1:

print("\n")

print("\*\*\*\*WELCOME TO PAYMENT MODULE\*\*\*\*")

print("\n")

print("1-->Add Payment")

print("2-->Modify Payment")

print("3-->Search Payment")

print("4-->Exit")

c=int(input("Enter your Choice:"))

if c==1:

Add\_Pay()

elif c==2:

Modify\_Pay()

elif c==3:

Search\_Pay()

elif c==4:

break

else:

print("Invalid Choice")

elif c==2:

break

else:

print("Invalid Choice")

elif ch==3:

break

else:

print("Invalid Choice")

cur.execute("select last\_day(curdate());")

r=cur.fetchone()

cur.execute("select curdate()")

y=cur.fetchone()

if r==y:

cur.execute("select SubscriberNo,Name,ProductType,Scheme,FacilitiesReq from Subscriber")

f=cur.fetchall()

print("Subscriber's Plan")

for row in f:

print(row)

cur.execute("select SubscriberNo,Name from Subscriber where Status='Pending'")

f=cur.fetchall()

if f==[]:

print("No Subscribers with Outstanding Bill")

else:

print("Subscriber - Outstanding Bills")

for row in f:

print(row)

s.commit()

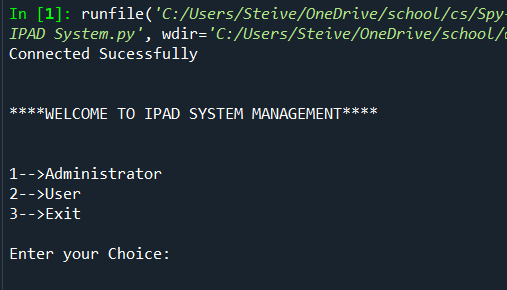
s.close()

**SCREENSHOTS**

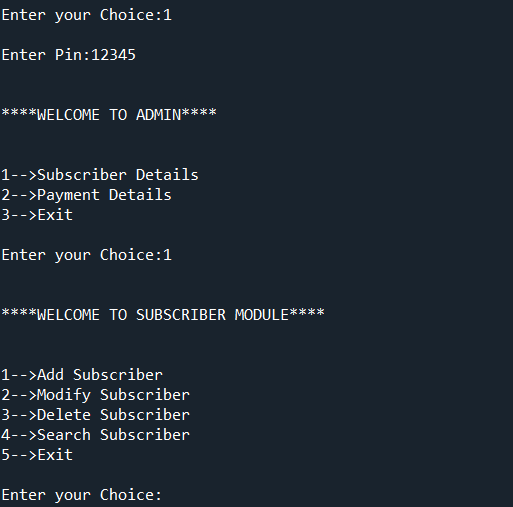
**OF**

**EXECUTION**

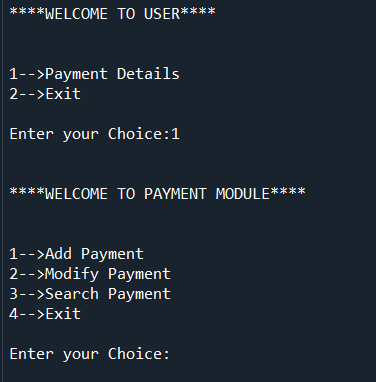
**MAIN MENU**



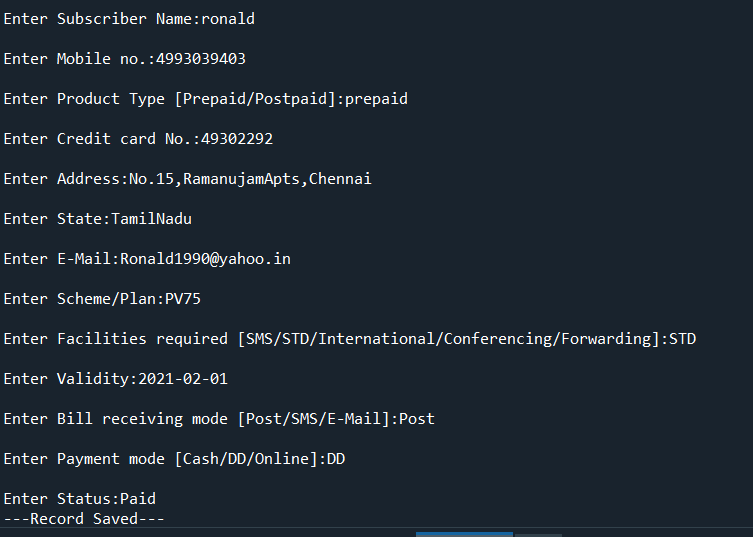
**ADMIN MODULE**



**USER MODULE**



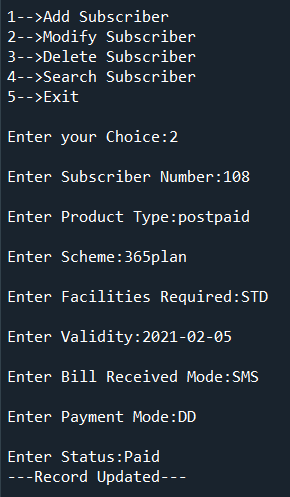
**ADDING RECORD**



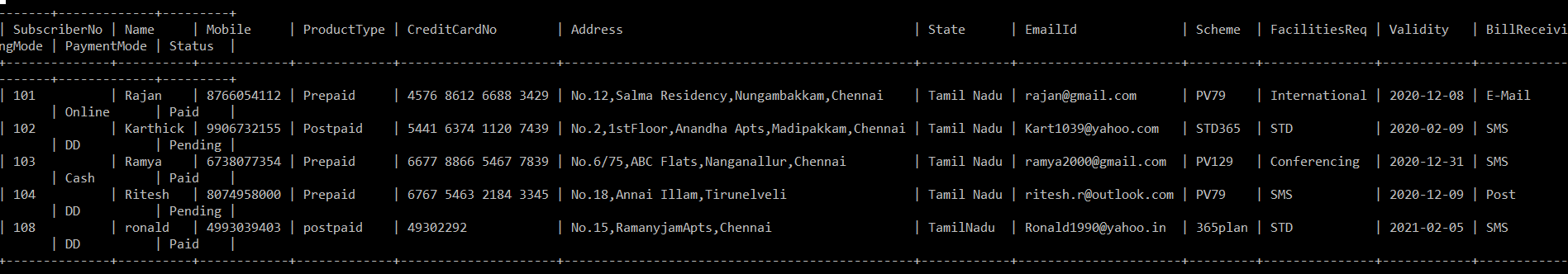
**RECORD ADDED**



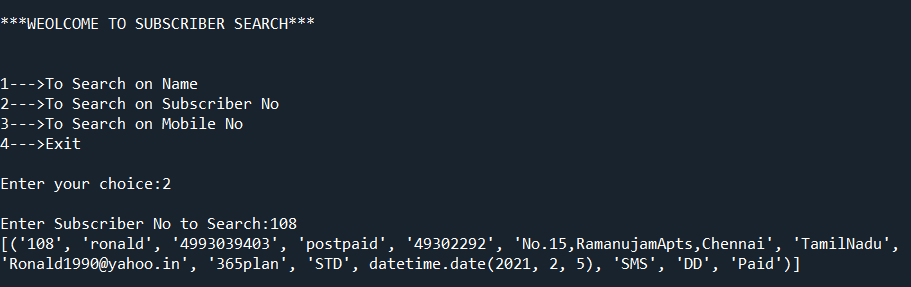
**MODIFYING RECORD**



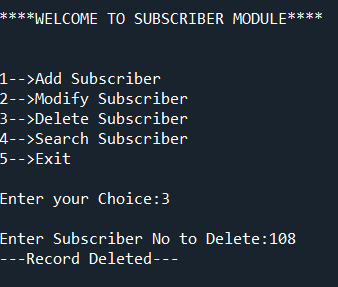
**RECORD MODIFIED**



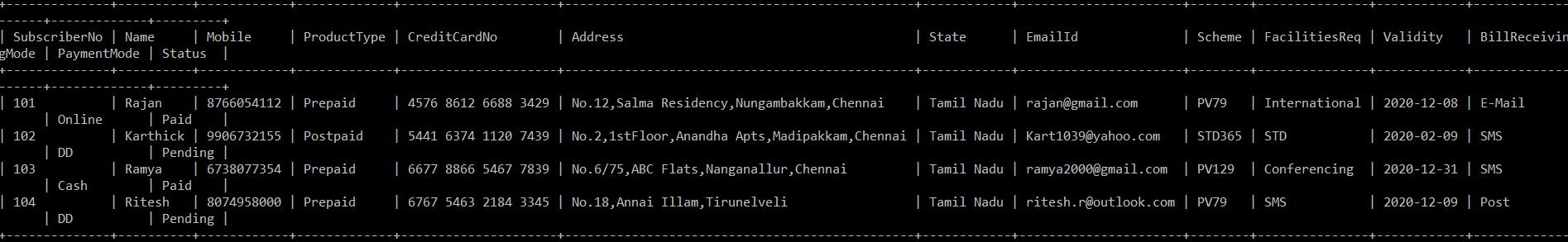
**SEARCHING RECORD**



**DELETING RECORD**



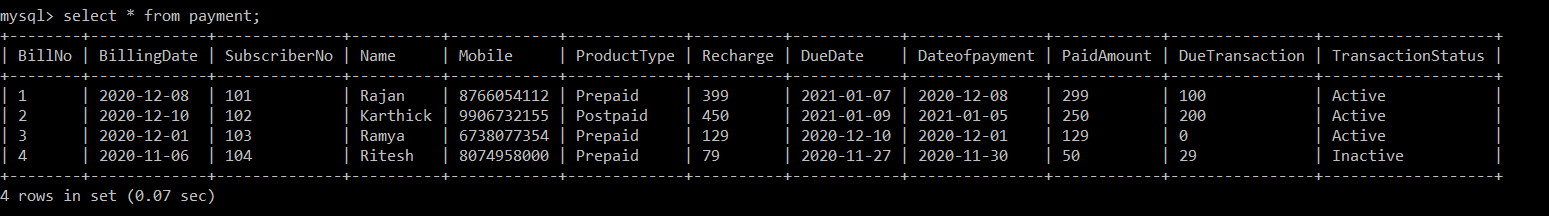
**RECORD DELETED**



**SUBSCRIBER TABLE**



**Payment Table**



**SYSTEM REQUIREMENTS**

**HARDWARE:**

* Proccesor: Pentium III and above
* Printer- to print the required documents of the project.
* Minimum memory - 2GB

**SOFTWARE:**

* Windows 7 or higher
* My-SQL server 5.5 or higher(as backend)
* Python idle 3.6 or higher or Spyder (as frontend).
* Microsoft Word 2010 or higher for documentation.

**BIBLIOGRAPHY**

In order to work on this project titled – Online Mobile Shopping, the following books and literature are referred by me during the various phases of development of the project.

* + Computer Science with python - by Sumita Arora
  + [www.python.org/download](http://www.python.org/download)
  + [www.py2exe.org](http://www.py2exe.org)
  + [www.mysql.org](http://www.mysql.org)