**HOLY CHILD SENIOR SECONDARY SCHOOL**

**TAGORE GARDEN**



ACADEMIC YEAR:-2021-22

**PROJECT REPORT ON LAUNDRY AND WARDROBE SERVICES**

NAME:-SIMRAN SARDANA

CBSE ROLL NO.:- 26633772

CLASS:-XII D

SUBJECT: - COMPUTER SCIENCE

SUBJECT CODE: - 083

**Table of Contents:-**

[ACKNOWLEDGEMENT 3](#_Toc96181652)

[INTRODUCTION 4](#_Toc96181653)

[APPLICATIONS 5](#_Toc96181654)

[CERTIFICATE 6](#_Toc96181655)

[HARDWARE AND SOFTWARE REQUIREMENTS 7](#_Toc96181656)

[HARDWARE REQUIREMENTS: 7](#_Toc96181657)

[SOFTWARE REQUIREMENTS: 8](#_Toc96181658)

[PYTHON AS FRONT END 8](#_Toc96181659)

[DATABASE (MySQL AS BACKEND) 9](#_Toc96181660)

[Er diagram 12](#_Toc96181661)

[FLOWCHART 13](#_Toc96181662)

[PYTHON MODULES & INBUILT FUNCTIONS 14](#_Toc96181663)

[UDF AND GLOSSARY 15](#_Toc96181664)

[SOURCE CODE 16](#_Toc96181665)

[OUTPUT SCREENSHOTS 38](#_Toc96181666)

[BIBLIOGRAPHY 51](#_Toc96181667)

hr

**index:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Description** | **Page No.** | **Sign.** |
| 1. | ACKNOWLEGEMENT |  |  |
| 2. | INTRODUCTION |  |  |
| 3. | APPLICATIONS |  |  |
| 4. | CERTIFICATE |  |  |
| 5. | HARDWARE AND SOFTWARE REQUIREMENTS |  |  |
| 6. | PYTHON AS FRONT END |  |  |
| 7. | DATABASE(MySql AS BACK END) |  |  |
| 8. | ER DIAGRAM |  |  |
| 9. | FLOWCHART |  |  |
| 10. | PYTHON MODULES AND IN-BULIT FUNCTIONS |  |  |
| 11. | UDF AND GLOSSARY |  |  |
| 12. | SOURCE CODE |  |  |
| 13. | OUTPUT SCREENSHOTS |  |  |
| 14. | BIBLIOGRAPHY |  |  |

# 

# ACKNOWLEDGEMENT

Accomplishments require collective efforts of many people and this work is no different. I am delighted to take this opportunity to express my heartfelt gratitude towards all those who made the successful completion of this project possible.

I would like to thank **Ms. Bhawna Sachdev** (Computer Science Teacher, Holy Child School) for her purposeful guidance and constant encouragement which made this seemingly difficult task a wonderful learning experience.

I am grateful to my team members Neetu & Garima(Students, Holy Child School), who helped me in getting the required information and contributed towards the perfection of the project.

Last but not the least, the project has been a great learning experience and I am indebted to Holy Child Sr. Sec. School for providing me with an opportunity to do the same.

# INTRODUCTION

The main aim of the project is to create a program using the interface of MySQL and python that can help in management of a Laundry and Wardrobe Store. This project will make the work of an employee as well as the owner easier for an instance accessing the details of the customers as well as modifying or updating or altering them to some extent. The primary reason to opt or chose this particular topic is to make a part of the work easier for people working hard by making an efficient use of technology as well as learning a lot through it. Moreover, it is a step towards developing a whole system of data handling. There are a lot of facilities provided by the project. They are as follows:-

1) The insertion of the customer details for the different services they chose in different tables to keep it simple, clean and non-confusing.

2) The services included are:-

(i) Dry Cleaning

(ii) Ironing

(iii) Dyeing

(iv) Darning

(v) Wardrobe Services

Each service will have no specific rates, there are generalised prices to keep it moderate and not sophisticated.

3)Invoice will be generated at the end of the insertion , with the ecode for the customer based on their insertion of details for different services, along with applicable rate of GST already included and total amount and the date of return for their belongings.

4)At the end, when the belongings are to be returned to the customer their record will be deleted from the tables and a final bill will be generated to them.

# APPLICATIONS

1) People running the store/organization will benefit from this for their organizational and legal purposes. It also helps as time savior as the manual data entry and deletion takes time to manage as well as to keep it a well-mannered way is not an easy task.

2) This program will ensure to provide services without any complaints and keeping a record of the efficiency, potential, productivity of the store. Moreover, it will make the work of the employee easier as the manual entering and discarding the records after the return of belongings is a tiring and laborious job. Relying on manual processes to manage data can result in many inaccuracies. Tracking information by hand not only increases the likelihood or human errors, but it exposes you to certain compliance risks too.

3) It will help in the enhancement of the business. Nonetheless, it will also provide much less effort in calculating the total price for a customer chosen service.

4) It will only be accessible to the employee and the owner and the customer wouldn’t be able to have access to all the information within the program and tables therefore the question for the privacy also vanishes out.

# CERTIFICATE



This is a bona fide project done satisfactorily at Holy Child Senior Secondary School by Garima Angelina, Neetu Yadav and Simran Sardana in fulfillment of the same.

This report or a similar project on the topic **LAUNDRY AND WARDROBE SERVICES** has not been submitted for any other examination and does not include any part of any other course undergone by the candidate.

DATE:-\_\_\_/\_\_\_/\_\_\_\_\_\_\_

TEACHER’S SIGNATURE:-\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

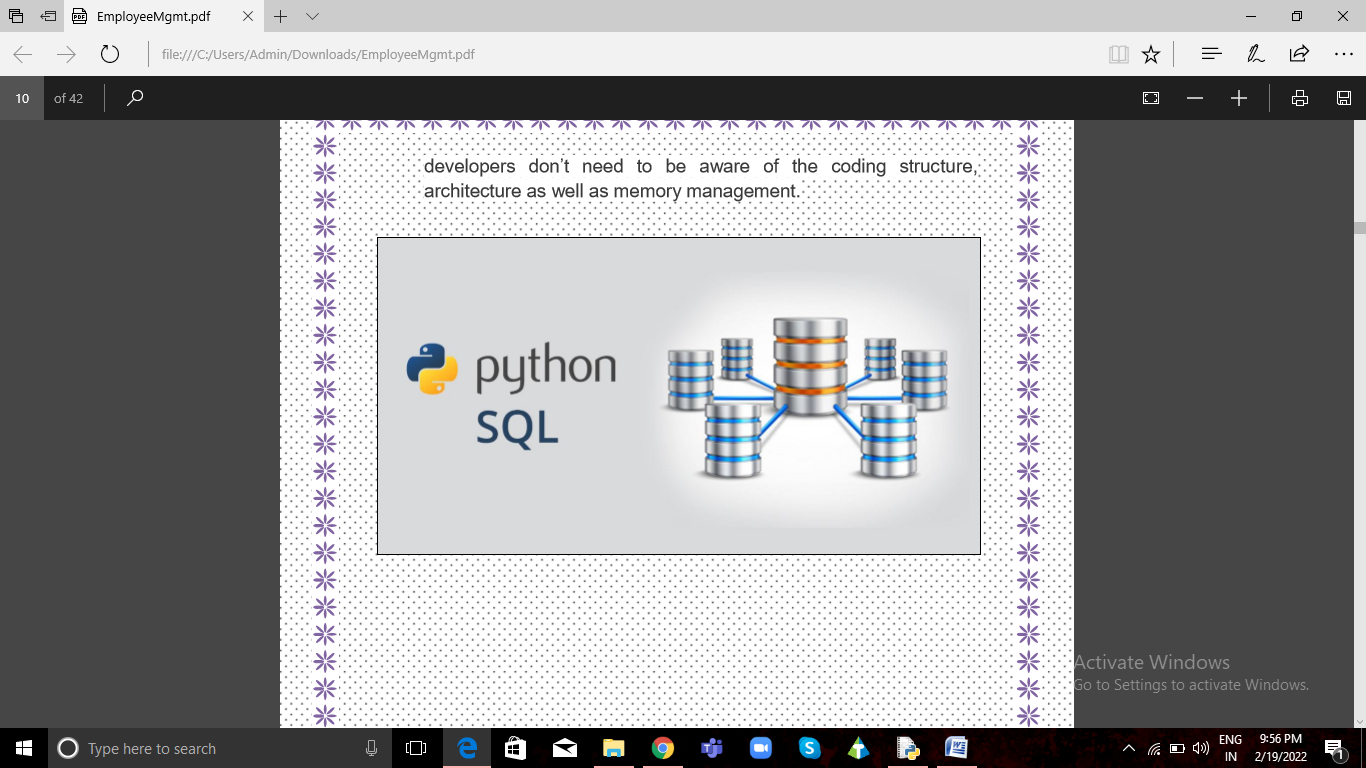
# HARDWARE AND SOFTWARE REQUIREMENTS

## HARDWARE REQUIREMENTS:

* OPERATING SYSTEM :Windows 10 Pro
* PROCESSOR :Intel(R) Pentium(R) CPU @ 1.99GHz 1.99 GHz
* MOTHERBOARD :Dell Inc. 0M78VH
* RAM :4.00 GB
* HARD DISK :KBG40ZNS256G NVMe KIOXIA 256GB
* MONITOR:13-17 inch
* KEYBOARD AND MOUSE:-WIRED
* PRINTER:-REQUIRED

## SOFTWARE REQUIREMENTS:

* WINDOWS 10/MAC OS AND ABOVE
* PYTHON
* MYSQL
* PYTHON AND MYSQL CONNECTIVITY



# PYTHON AS FRONT END

Python is undoubtedly the one of the popular programming language of the present IT world. Learning Python programming language is a wonderful experience and being a Python developer opens up the door for number of Python developer job opportunities. One way to learn Python and gain expertise in this programming language is to start working on Python projects. Moreover, project-based learning helps to improve student’s knowledge.

**Did You Know?***Python is the preferred programming language for working on Artificial Intelligence and Machine Learning projects.*

**Some of the features of Python programming language are:-**

* **Easy to learn and use:** Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block.
* **Expressive Language**: python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print(“Hello World”). It will take only one to execute, while Java and C takes multiple lines.
* **Object-Oriented language:** Python supports Object-Oriented programming concept. This feature helps programmers to write reusable code and develop applications in lesser code.
* **Provides large standard library:** Python provides large number of libraries that has some handy codes and functions making it easy for programmers to build projects.
* **Cross-platform language:** Python can run equally on different platforms such as Windows, Linux, Unix and Macintosh etc. So, we can say that python is a portable language.It enables programmers to develop the software for several competing platforms by writing a program only once
* **Interpreted Language:** Python is an interpreted programming language. This means Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.
* **High-Level Language:** Python has been designed to be a high-level programming language. It means while writing Python codes developers don’t need to be aware of the coding structure, architecture as well as memory management.



# DATABASE (MySQL AS BACKEND)

MySQL is a free-to-use, open-source database that facilitates effective management of databases by connecting them to the software. It is a stable, reliable and powerful solution with advanced features including:

* **Data Security**

MySQL is globally renowned for being the most secure and reliable database management system used in popular web applications including WordPress, Facebook and Twitter. The data security and support for transactional processing that accompany the recent version of MySQL can greatly benefit any business , especially if it is an eCommerce business that involves frequent money transfers.

* **High Performance**

Whether it is an eCommerce website that receives a million queries every single day or a high-speed transactional processing system, MySQL is designed to meet even the most demanding applications while ensuring optimum speed for enhanced flawless performance.

* **Comprehensive Transactional Support**

MySQL tops the list of robust transactional database engines available on the market. It is the go-to solution for full data integrity. It guarantees instant deadlock identification through server-enforced referential integrity.

* **Complete Workflow Control**

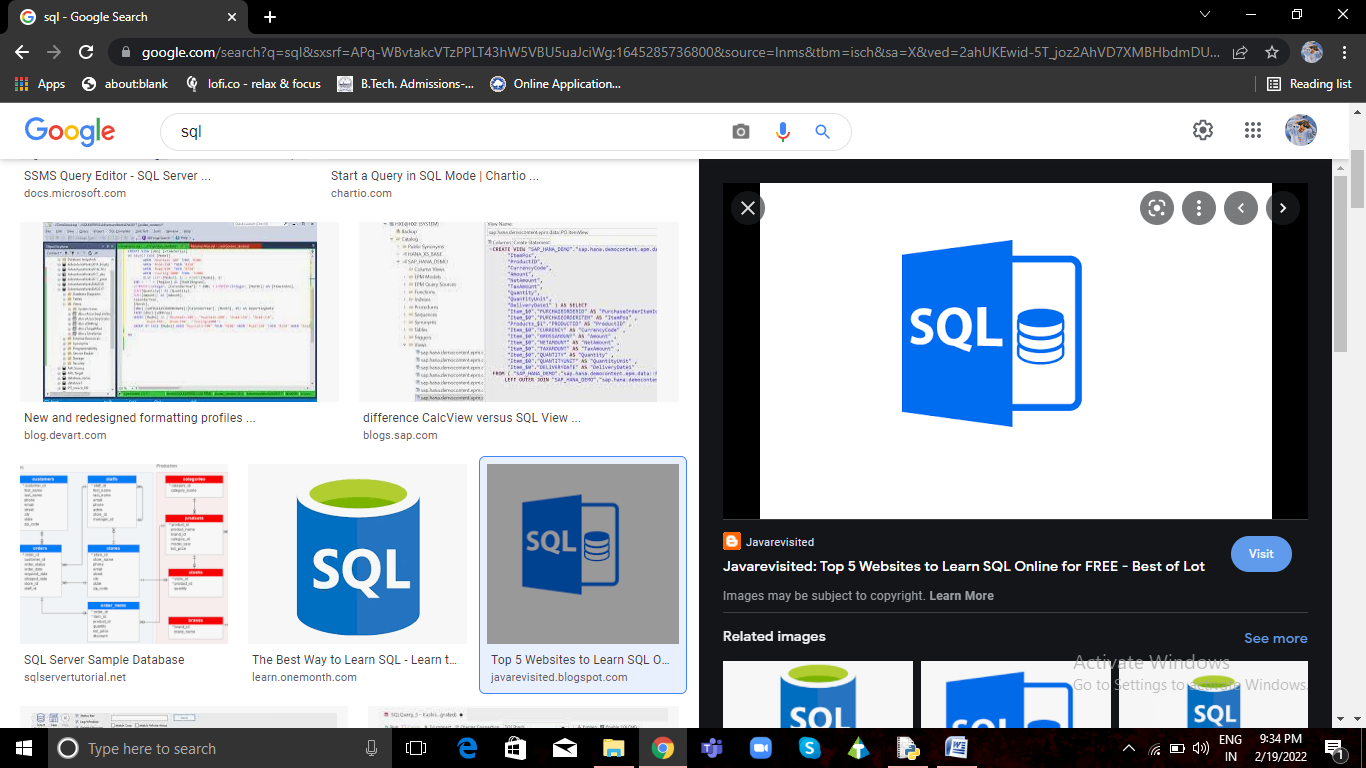
With an average download and installation time of less than 30 minutes, MySQL means usability from day one. Whether your platform is Linux, Microsoft, Macintosh or UNIX, MySQL is a comprehensive solution with self-management features.

* **Reduced Total Cost Of Ownership**

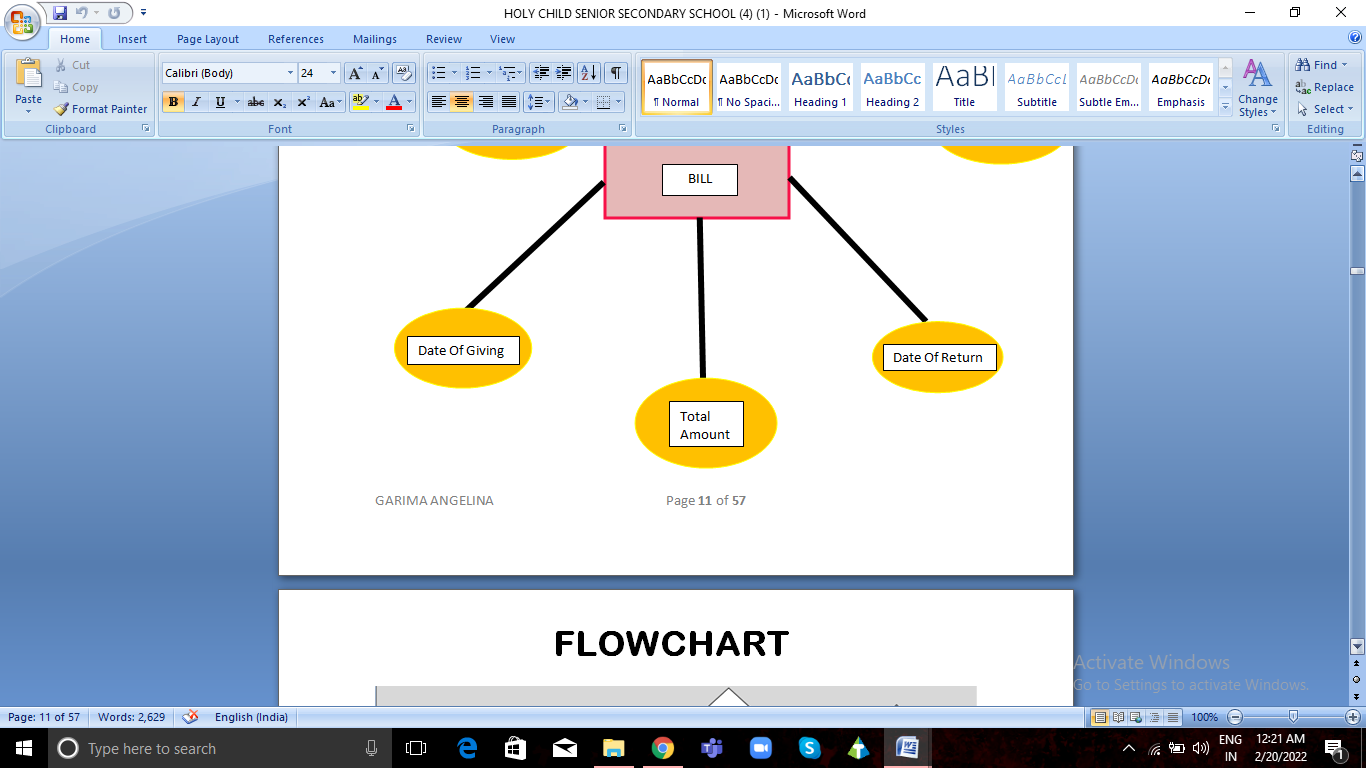
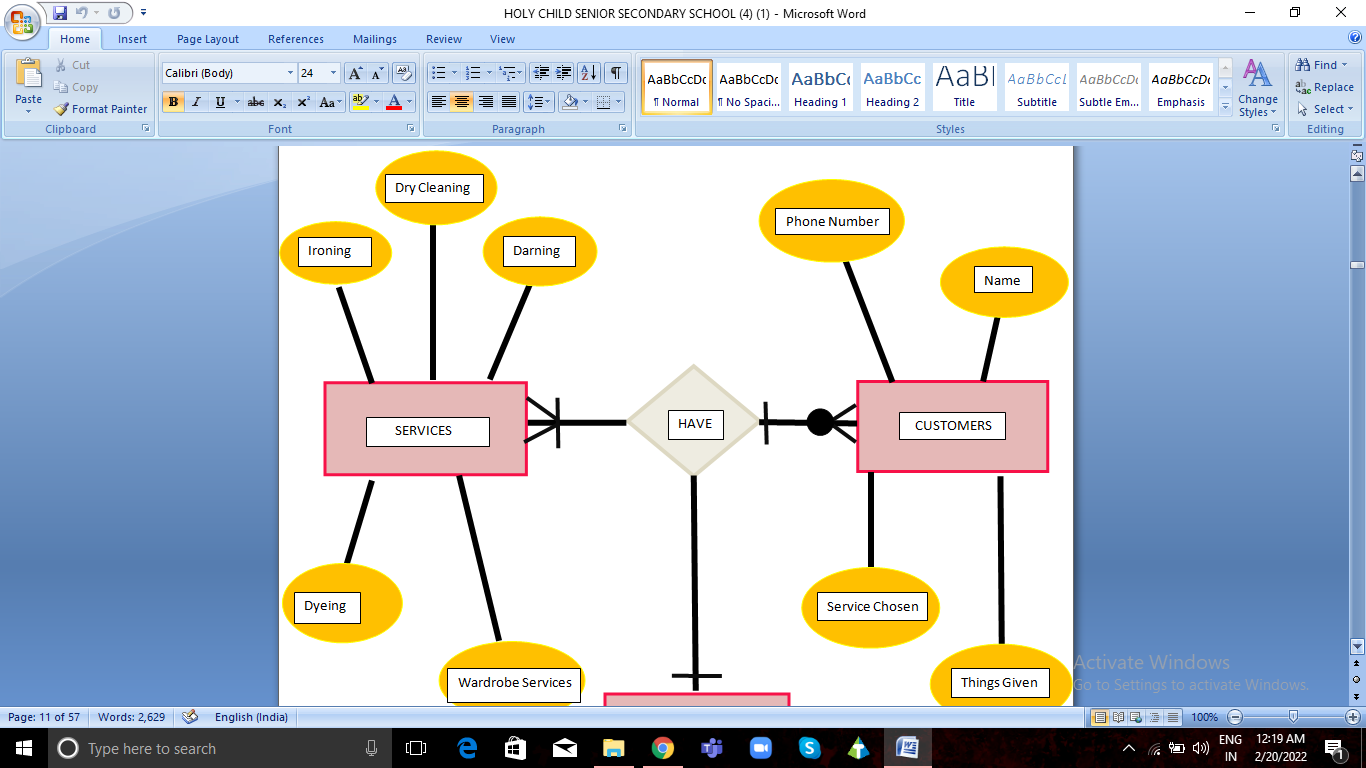
By migrating current database apps to MySQL, enterprises enjoy significant cost savings on new projects. The dependability and ease of management can save fixing downtime issues and performance problems.

* **The Flexibility Of Open Source**

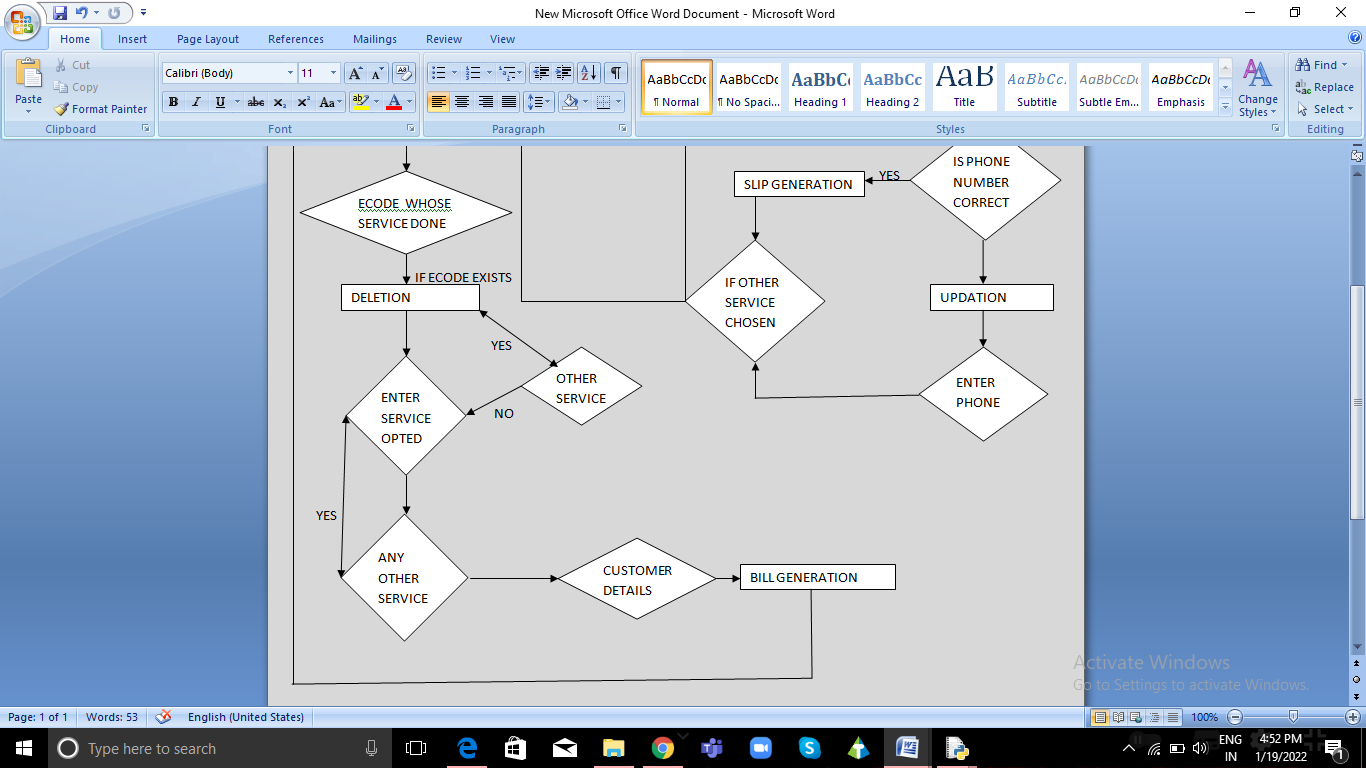
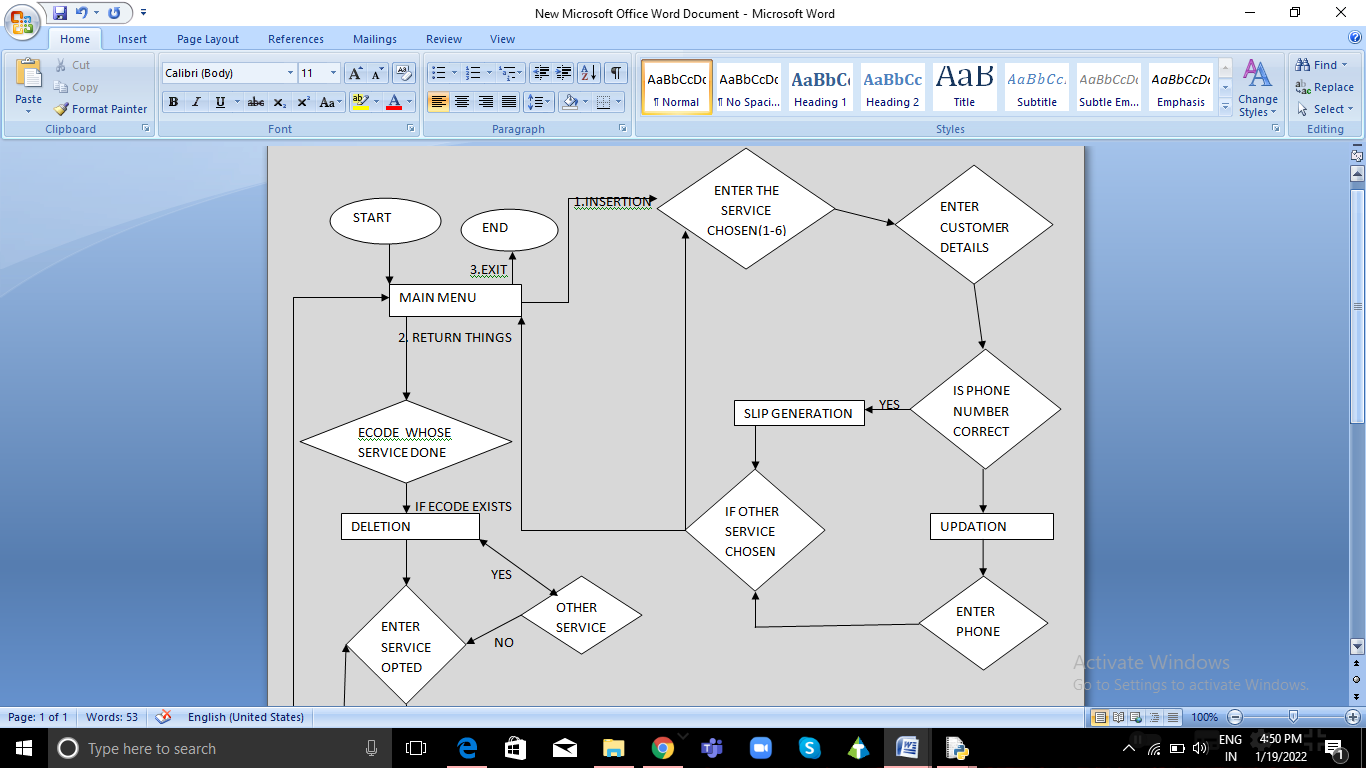
By migrating current database apps to MySQL, enterprises enjoy significant cost savings on new projects. The dependability and ease of management can save fixing downtime issues and performance problems.

****

# Er diagram



# FLOWCHART



# PYTHON MODULES & INBUILT FUNCTIONS

* **PYTHON MODULES USED:**
* **MySQL Connector as sql**
* MySql.connector-helps in connecting or establishing a connection between sql and python
* mycursor=mydb.cursor-Creates cursor object that helps to traverse through the data by different queries and code.
* mycursor.execute()-Helps in passing queries in MySql through python and to make changes in the tables associated with them.
* mydb.commit()- Helps in making the changes permanently in MySql tables.
* mycursor.fetchall- Helps in fetching all the records of the query passed
* **IN-BUILT FUNCTIONS USED:**

**len():-** len() is a built-in function in python. We can use the len() to get the length of the given string, array, list, tuple, dictionary, etc. len helps provide the number of elements in an object.

Syntax:-len(value)

**print():-** print() is a built-in function in python. We can use the print() to get data required directly on the screen or any other standard output device.

Syntax:-print(data)

**int():-** The int() function converts the specified value into an integer number.

Syntax:-int(value)

**str():-**The str() function converts the specified value into a string.

Syntax:-str(value)

**upper():-** the upper() function converts the case specified string into upper case

**lower():-** the upper() function converts the case specified string into lower case. The original string doesn’t get altered.

**split():-**The split() function separates a string into multiple strings, arranges them in a list, and returns the list.

**append():-** The append() method appends an element to the end of the list.

**format():-** The format() method returns a formatted representation of the given value controlled by the format specifier.

# 

# UDF AND GLOSSARY

1)insert\_user\_records():-

It is used to insert the records of the customer into the laundry and wardrobe services. It also generates a slip at last which is used when the customer comes back to collect their belongings.

2)name():-

It is used to display the different or a single service chosen by the customer when the bill generation takes place.

3)pricing():-

It is used for bill generation including the total amount to be paid along with the details of the customer.

4)update\_ph():-

It is used to update the phone number of the customer just after insertion of the customer details takes place.

5)delete():-

When a customer comes back to collect their belongings, initially their records are deleted to make sure there are no belongings of some customer left for the services and to avoid confusion at any ground.

6)check1() : To check the existence of table 1.

7)check2() : To check the existence of table 2.

8)check3() : To check the existence of table 3.

9)check4() : To check the existence of table 4.

10)check5() : To check the existence of table 5.

# SOURCE CODE

'''Board Project

Team no.:-3

Simran Sardana , Garima Angelina , Neetu Yadav'''

import mysql.connector

mydb=mysql.connector.connect(host="localhost",user="root",password="admin")

mycursor=mydb.cursor()

##mycursor.execute("CREATE DATABASE LAUNDRY")

mycursor.execute("USE LAUNDRY")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO LAUNDRY AND WARDROBE SERVICES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print('\n')

##string1="CREATE TABLE DRYCLEANING(SNO int(3) Primary key ,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3),ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

##mycursor.execute(string1)

##

##string2="CREATE TABLE IRONING(SNO int(3) Primary key,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3), ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

##mycursor.execute(string2)

##

##string3="CREATE TABLE DYEING(SNO int(3) Primary key,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3),ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

##mycursor.execute(string3)

##

##string4="CREATE TABLE DARNING(SNO int(3) Primary key ,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3),ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

##mycursor.execute(string4)

##

##string5="CREATE TABLE WARDROBESERVICES(SNO int(3) Primary key ,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3),ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

##mycursor.execute(string5)

##def check1():

## mycursor.execute("SHOW DATABASES")

## found=0

## for i in mycursor:

## if i==('LAUNDRY',):

## found=1

## if found==0:

## mycursor.execute("CREATE DATABASE LAUNDRY")

## mycursor.execute("USE LAUNDRY")

## mycursor.execute("SHOW TABLES")

## found=0

## for i in mycursor:

## if i==('DRYCLEANING',):

## found=1

## if found==0:

## string1="CREATE TABLE DRYCLEANING(SNO int(3) Primary key,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3), ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

## mycursor.execute(string1)

##def check2():

## mycursor.execute("SHOW DATABASES")

## found=0

## for i in mycursor:

## if i==('LAUNDRY',):

## found=1

## if found==0:

## mycursor.execute("CREATE DATABASE LAUNDRY")

## mycursor.execute("USE LAUNDRY")

## mycursor.execute("SHOW TABLES")

## found=0

## for i in mycursor:

## if i==('IRONING',):

## found=1

## if found==0:

## string2="CREATE TABLE IRONING(SNO int(3) Primary key,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3), ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

## mycursor.execute(string2)

##def check3():

## mycursor.execute("SHOW DATABASES")

## found=0

## for i in mycursor:

## if i==('LAUNDRY',):

## found=1

## if found==0:

## mycursor.execute("CREATE DATABASE LAUNDRY")

## mycursor.execute("USE LAUNDRY")

## mycursor.execute("SHOW TABLES")

## found=0

## for i in mycursor:

## if i==('DYEING',):

## found=1

## if found==0:

## string3="CREATE TABLE DYEING(SNO int(3) Primary key,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3), ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

## mycursor.execute(string3)

##def check4():

## mycursor.execute("SHOW DATABASES")

## found=0

## for i in mycursor:

## if i==('LAUNDRY',):

## found=1

## if found==0:

## mycursor.execute("CREATE DATABASE LAUNDRY")

## mycursor.execute("USE LAUNDRY")

## mycursor.execute("SHOW TABLES")

## found=0

## for i in mycursor:

## if i==('DARNING',):

## found=1

## if found==0:

## string4="CREATE TABLE DARNING(SNO int(3) Primary key,NAME VARCHAR(20),\

##DATE DATE, CONTENTS VARCHAR(25), QUANTITY INT(3), ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

## mycursor.execute(string4)

##def check5():

## mycursor.execute("SHOW DATABASES")

## found=0

## for i in mycursor:

## if i==('LAUNDRY',):

## found=1

## if found==0:

## mycursor.execute("CREATE DATABASE LAUNDRY")

## mycursor.execute("USE LAUNDRY")

## mycursor.execute("SHOW TABLES")

## found=0

## for i in mycursor:

## if i==('WARDROBESERVICES',):

## found=1

## if found==0:

## string5="CREATE TABLE WARDROBE SERVICES(SNO int(3) Primary key,NAME VARCHAR(20),\

##DATE DATE, CONTENT VARCHAR(25), QUANTITY INT(3), ECODE INT(4),DATEOFRETURN DATE,PHONE NUMERIC(10))"

## mycursor.execute(string5)

def insertion():

print("Following are the different services:-")

print("1. DRY CLEANING")

print("2. IRONING")

print("3. DYEING")

print("4. DARNING")

print("5. WARDROBE SERVICES")

print("6. EXIT")

table=''

t=int(input("Enter the service you want to proceed with(1-6)="))

if t==1:

table='drycleaning'

elif t==2:

table='ironing'

elif t==3:

table='dyeing'

elif t==4:

table='darning'

elif t==5:

table='wardrobeservices'

elif t==6:

print("Proceeding to the main menu")

return

mycursor.execute("SELECT \* FROM {}".format(table))

items={'saree':250,'blazer':300,'shawl':150,'blanket':350,'quilt':400,

'dress':400,'jeans':150,'suit':200,'curtain':100,'covers':400,

'shirt':100,'jacket':150}

for i in mycursor:

print(i)

print('\n')

sno=int(input("Enter Sno:-"))

nm=input("Enter Name:-")

dt=str(input("Enter Date(yyyy-mm-dd):-"))

contents=''

while True:

ct=input("Enter Contents:-")

contents+=(str(ct)+' ')

print(contents)

ans=input("Are there more items to give for service(y/n)=")

if ans in 'yY':

continue

elif ans in 'nN':

break

for x in (contents.split()):

qnt=int(input("Enter Quantity for {}:-".format(x)))

for x in (contents.split()):

if x.lower() in items:

pc=items[x.lower()]

else:

pc=int(input("Enter Price of {}:-".format(x)))

##ec=int(input("Enter Ecode:-"))

global ecode

ec=int(input("Enter the last ecode alotted="))

ec+=1

tt=qnt\*pc

dor=str(input("Enter Date of Return:-"))

ph=int(input("Enter Phone no:-"))

for x in (contents.split()):

mycursor.execute("INSERT INTO {} VALUES({},'{}','{}','{}',{},{},'{}',{})".format(table,sno,nm,dt,x,qnt,ec,dor,ph))

sno+=1

mydb.commit()

print("Record given by you is inserted....")

fr=input("Is the phone number inserted for the insertion correct=")

if fr in 'yY':

print("Ok")

else:

update()

mycursor.execute("SELECT \* FROM {}".format(table))

for i in mycursor:

print(i)

print('\n')

print("-------SLIP GENERATION---------")

print(tt)

print("Your invoice no=",ec)

print("Date of return:",dor)

print("Collect your belongings on",dor,".Have a nice day ahead.")

print('\n')

ans=input("Is there anything to be given for other services(y/n):")

if ans in 'yY':

insertion()

elif ans in 'nN':

print("-------------------------------------Thanks.Have a nice day ahead.-----------------------------------------------")

def name():

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("Following are the different services:-")

print("1. DRY CLEANING")

print("2. IRONING")

print("3. DYEING")

print("4. DARNING")

print("5. WARDROBE SERVICES")

ch=int(input("Enter the service chosen(1-5)?"))

if ch==1:

print("DRY CLEANING")

elif ch==2:

print("IRONING")

elif ch==3:

print("DYEING")

elif ch==4:

print("DARNING")

elif ch==5:

print("WARDROBE SERVICES")

c=input("Did you choose other services also(Y/N)=")

if c.upper()=='Y':

name()

elif c.upper()=="N":

print("Thanks...........")

def pricing():

items={'saree':250,'blazer':300,'shawl':150,'blanket':350,'quilt':400,

'dress':400,'jeans':150,'suit':200,'curtain':100,'covers':50,

'shirt':100,'jacket':150}

custname=input("Enter your name=")

print("Welcome to our store",custname)

cost=[]

while True:

things=input("Enter the things given for service=")

cost.append(things.lower())

print(cost)

ans=input("Are there more things(y/n)=")

if ans in 'yY':

continue

elif ans in 'nN':

break

a=0

for x in cost:

if x in items:

qt=int(input("no. of {} :-".format(x)))

a+=(items[x]\*qt)

print(x,items[x])

else:

qt=int(input("no. of {} :-".format(x)))

pc=int(input("Enter price for {}:".format(x)))

print(x,pc)

a+=(qt\*pc)

print(a)

def delete():

print("Following are the different services:-")

print("1. DRY CLEANING")

print("2. IRONING")

print("3. DYEING")

print("4. DARNING")

print("5. WARDROBE SERVICES")

print("6. EXIT")

table=''

t=int(input("Enter the service you want to proceed with(1-6)="))

if t==1:

table='drycleaning'

elif t==2:

table='ironing'

elif t==3:

table='dyeing'

elif t==4:

table='darning'

elif t==5:

table='wardrobeservices'

elif t==6:

print("Proceeding to the main menu")

return

while True:

mycursor.execute("SELECT \* FROM {}".format(table))

myrecords=mycursor.fetchall()

print('sno','name','date','contents','quantity','ecode','dateofreturn','phone',sep=' ')

print("--------------------------------------------------------------------------------------------------------")

for i in myrecords:

print(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7],sep=' ')

found=0

e=int(input("Enter ecode whose service has been completed="))

for i in myrecords:

if i[5]==e:

found=1

if found==0:

print(e,"does not exist in the service")

else:

mycursor.execute("DELETE FROM {} WHERE ECODE={}".format(table,e))

mydb.commit()

mycursor.execute("SELECT \* FROM {}".format(table))

myrecords=mycursor.fetchall()

print('sno','name','date','contents','quantity','ecode','dateofreturn','phone',sep=' ')

print("--------------------------------------------------------------------------------------------------------")

for i in myrecords:

print(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7],sep=' ')

ch=input("Did you choose any other service also(Y/N):-")

if ch.upper()=='Y':

continue

elif ch.upper()=='N':

break

def update\_ph():

table=input("Enter the table in which you want updation to be done=")

mycursor.execute("SELECT \* FROM {}".format(table))

myrecords=mycursor.fetchall()

for i in myrecords:

print(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7])

ec=int(input("Enter ecode whose phone you want to update="))

found=0

for x in myrecords:

if x[5]==ec:

found=1

if found==1:

ph=int(input("Enter new phone="))

mycursor.execute("UPDATE {} SET PHONE={} WHERE ECODE={}".format(table,ph,ec))

mydb.commit()

mycursor.execute("SELECT \* FROM {}".format(table))

myrecords=mycursor.fetchall()

for i in myrecords:

print(i[0],i[1],i[2],i[3],i[4],i[5],i[6],i[7])

mydb.commit()

print("Record given by you is updated....")

#MENU-DRIVEN:-

while True:

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*MAIN MENU\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

print("1. Give things for service")

print("2. Return things given for service")

print("3. Exit")

op=int(input("Enter the option chosen(1-3)="))

if op==1:

insertion()

elif op==2:

delete()

name()

pricing()

elif op==3:

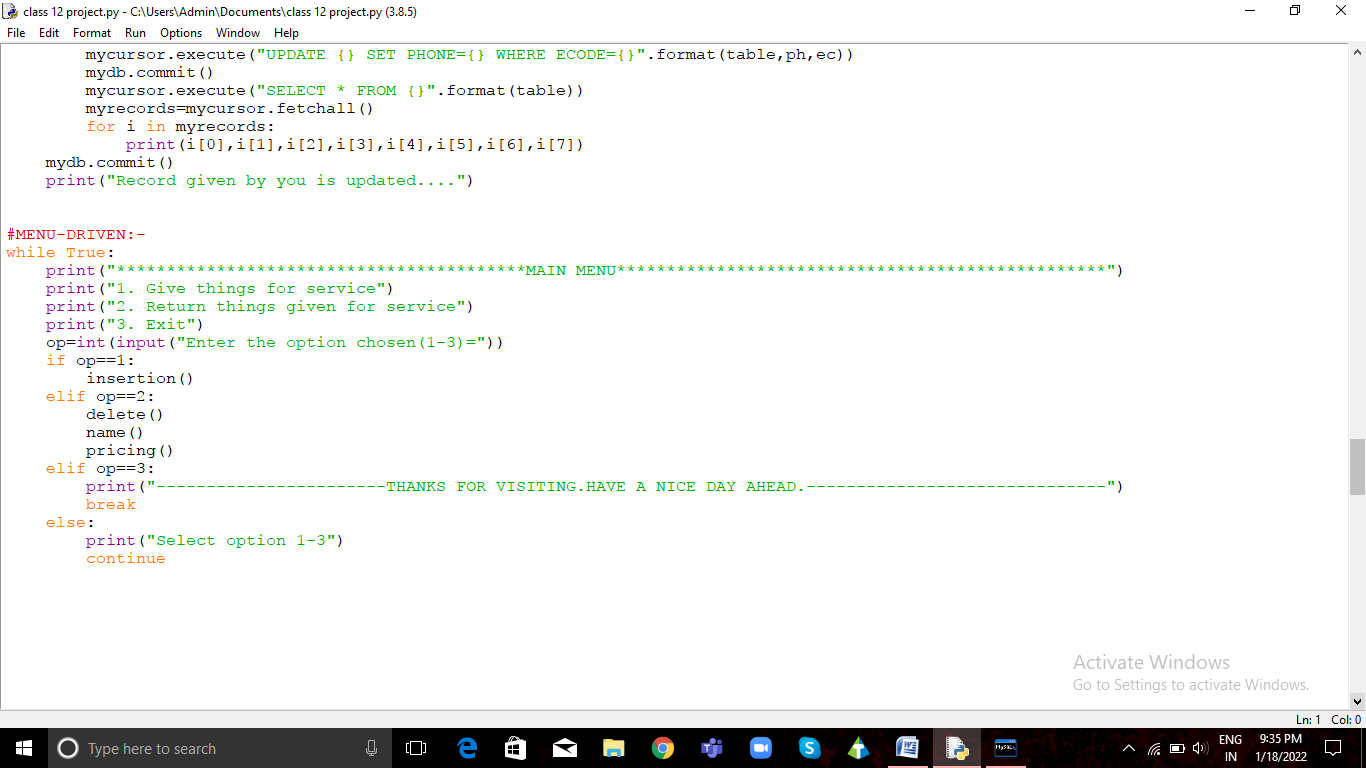
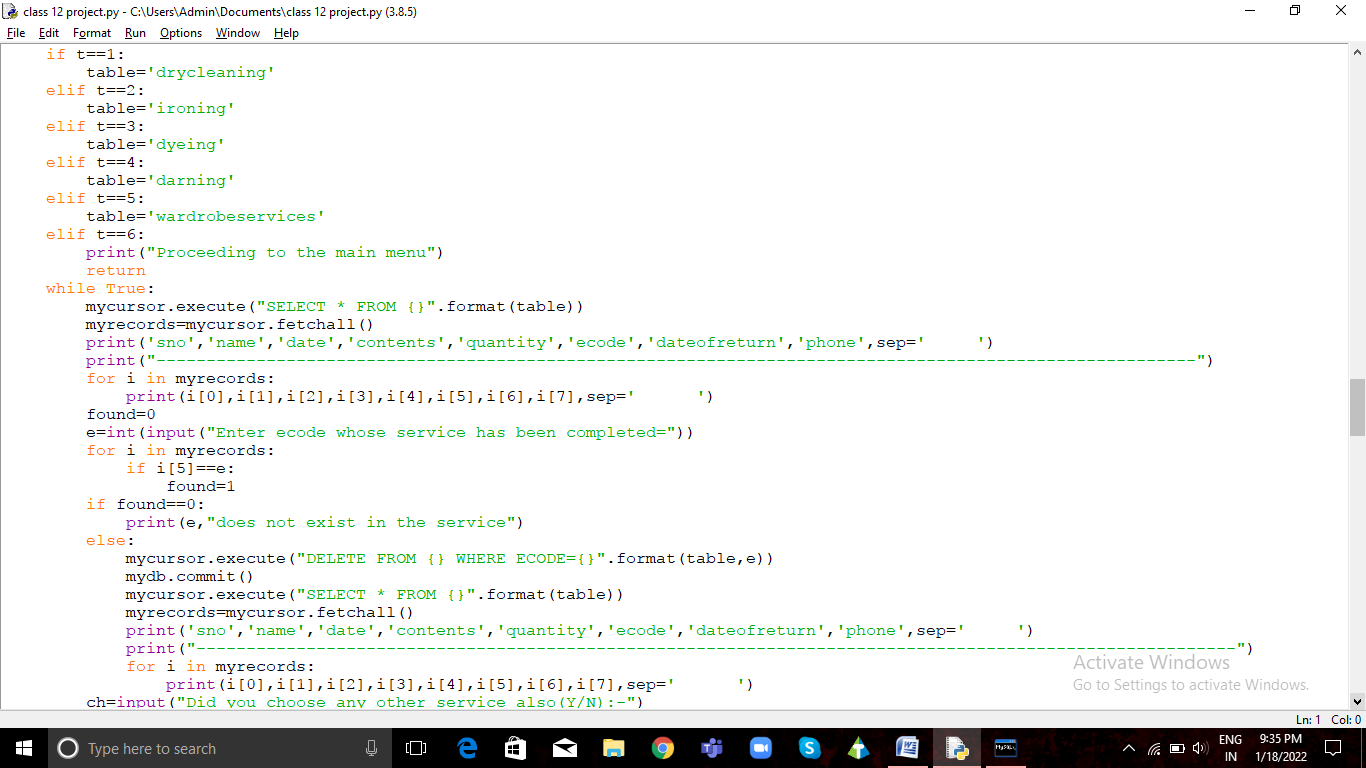
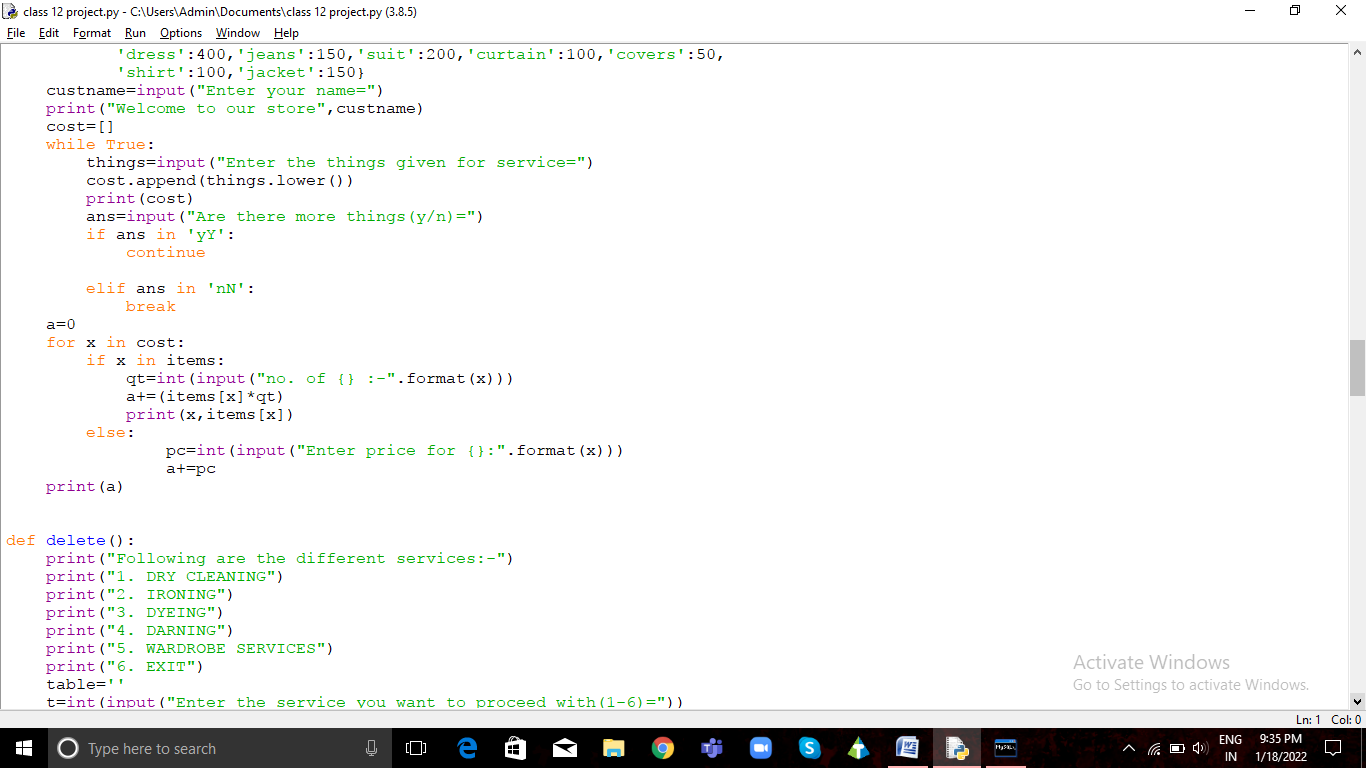
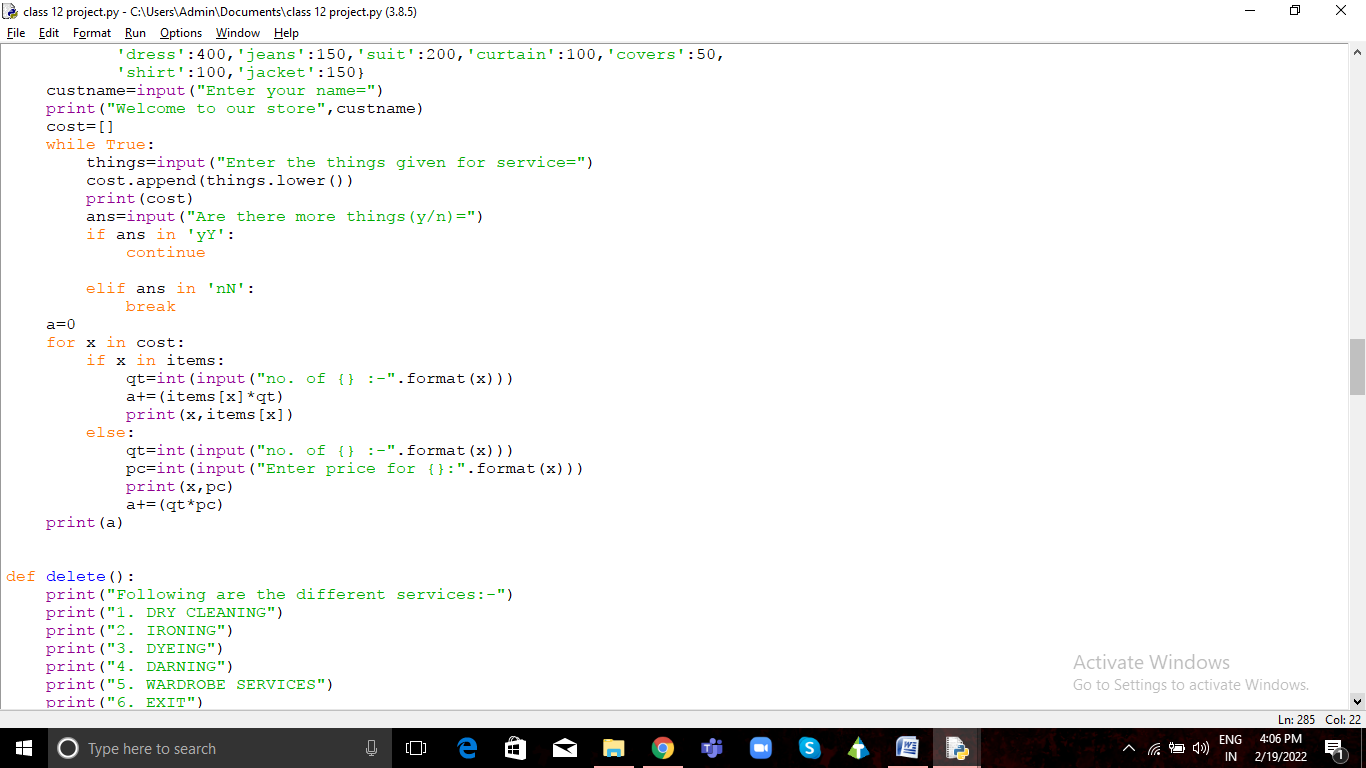
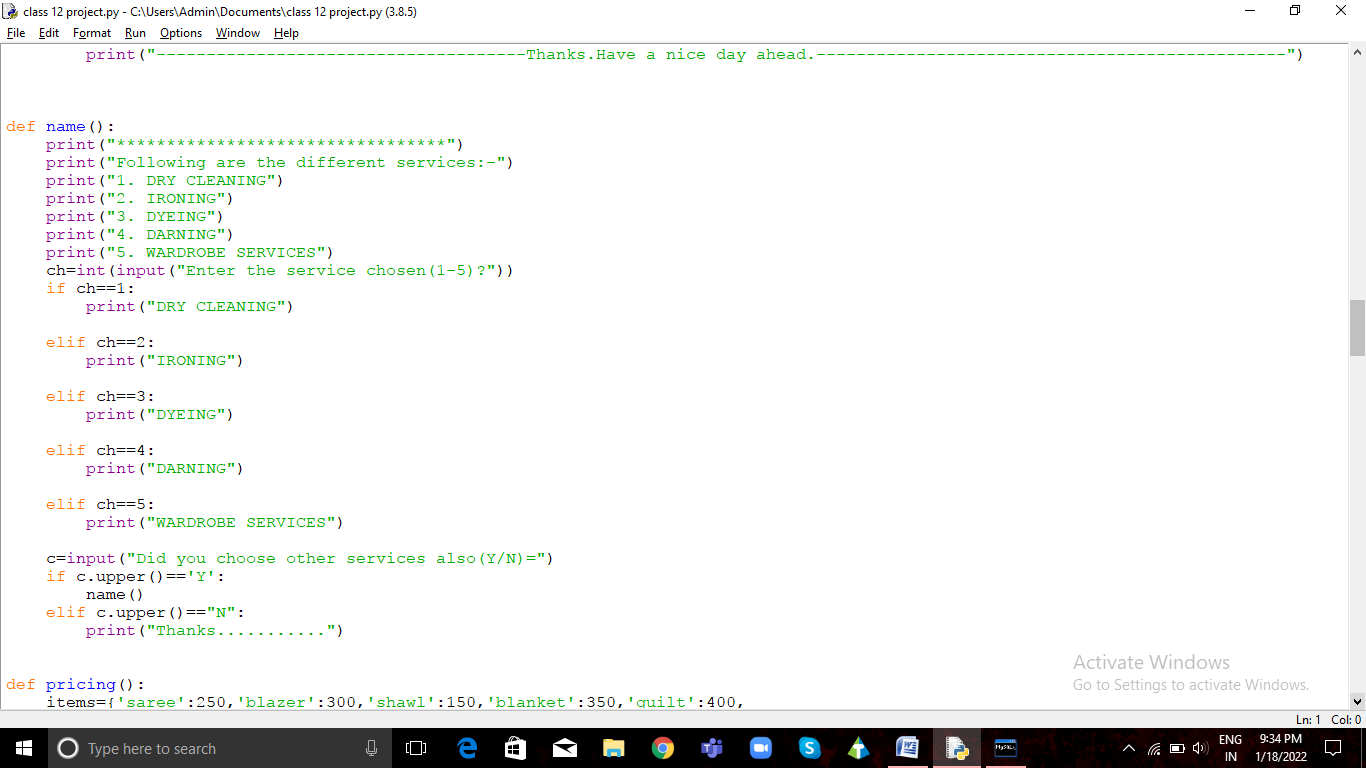
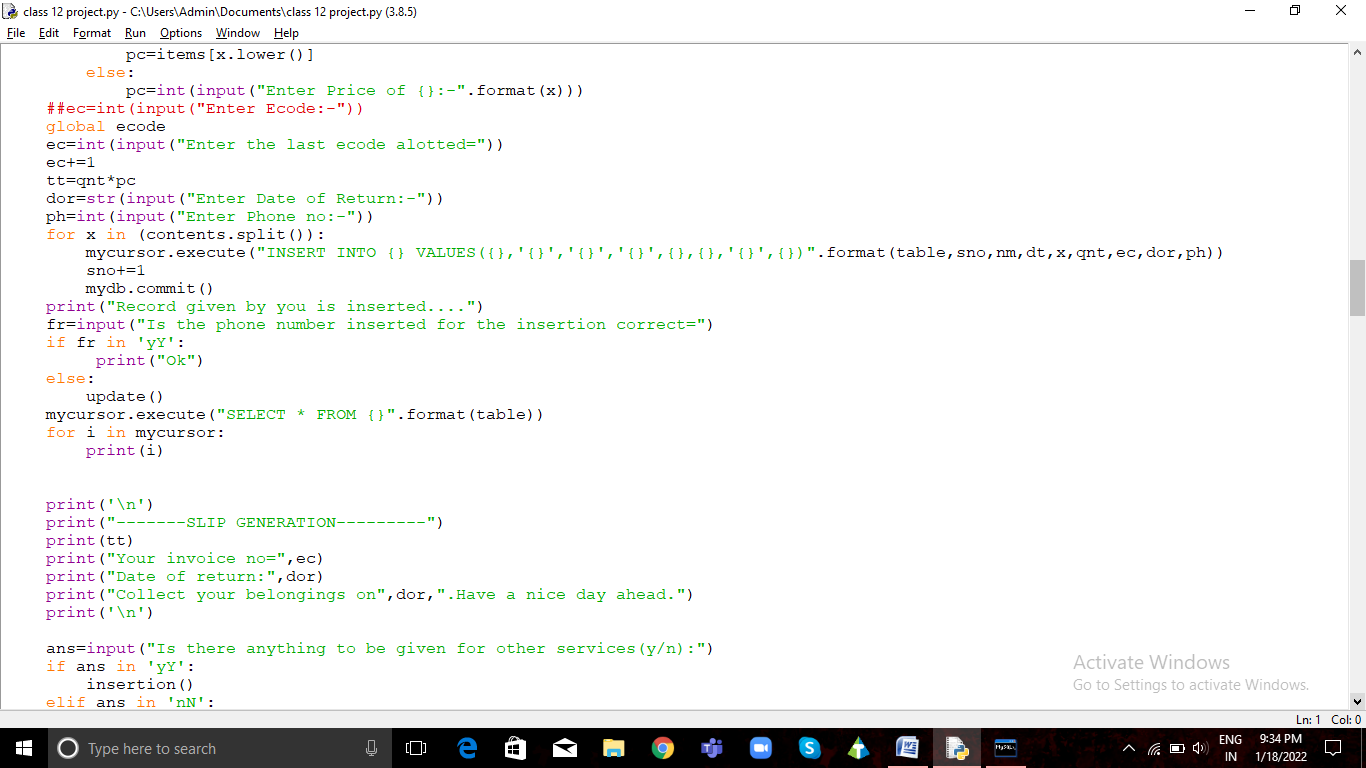
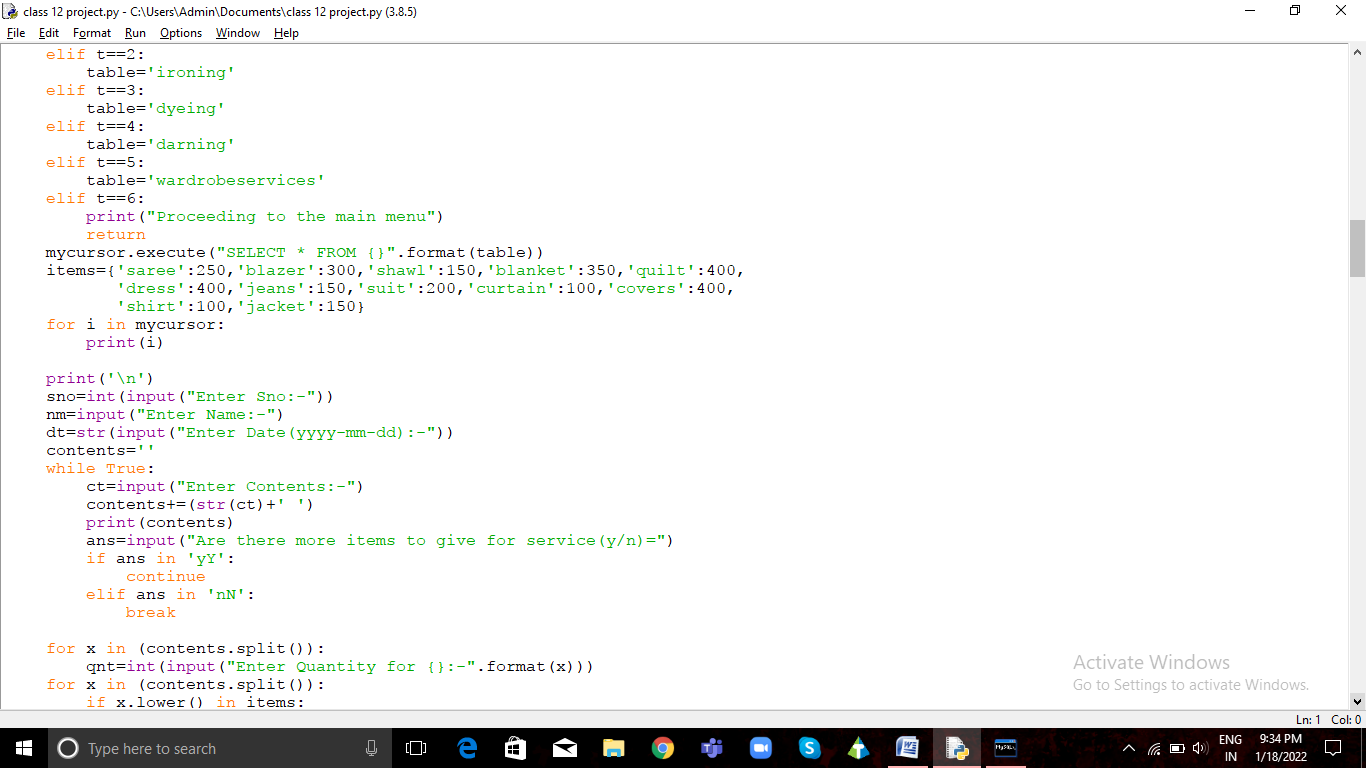
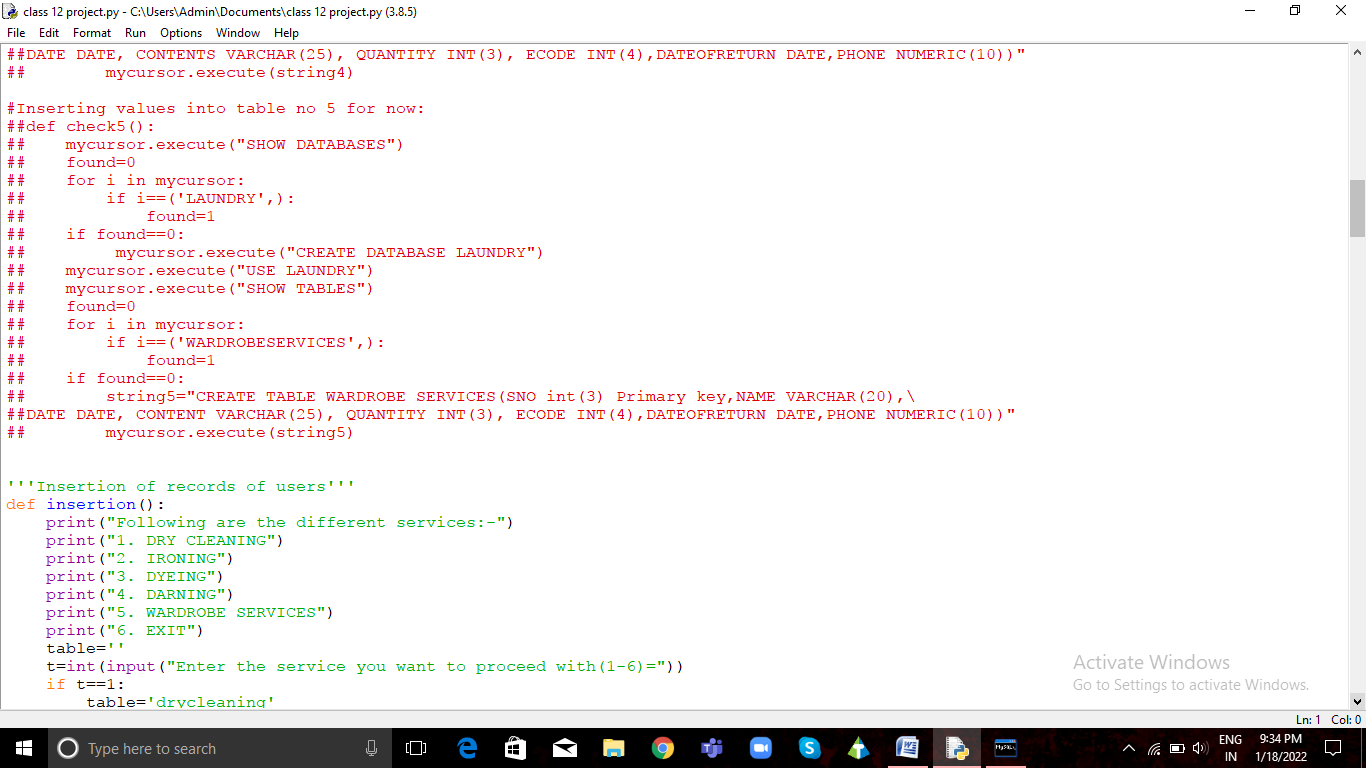
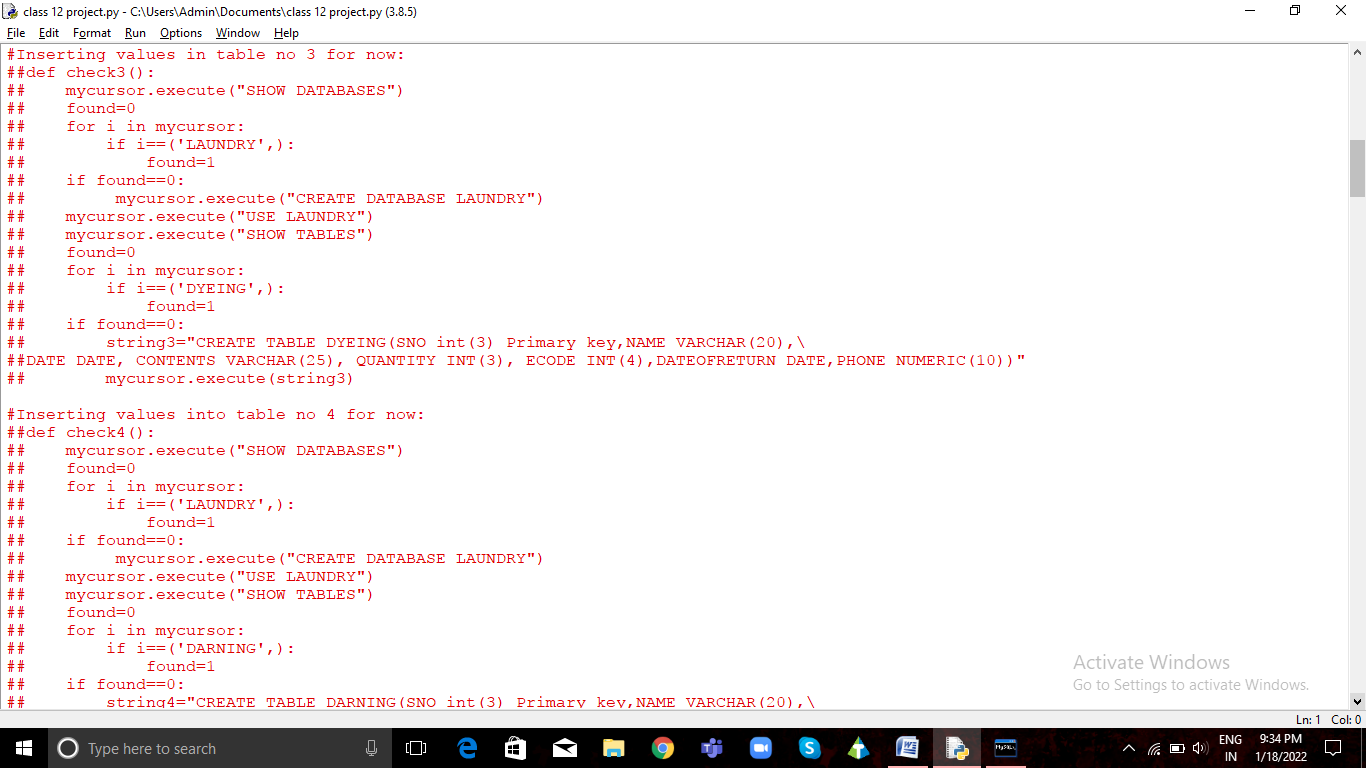
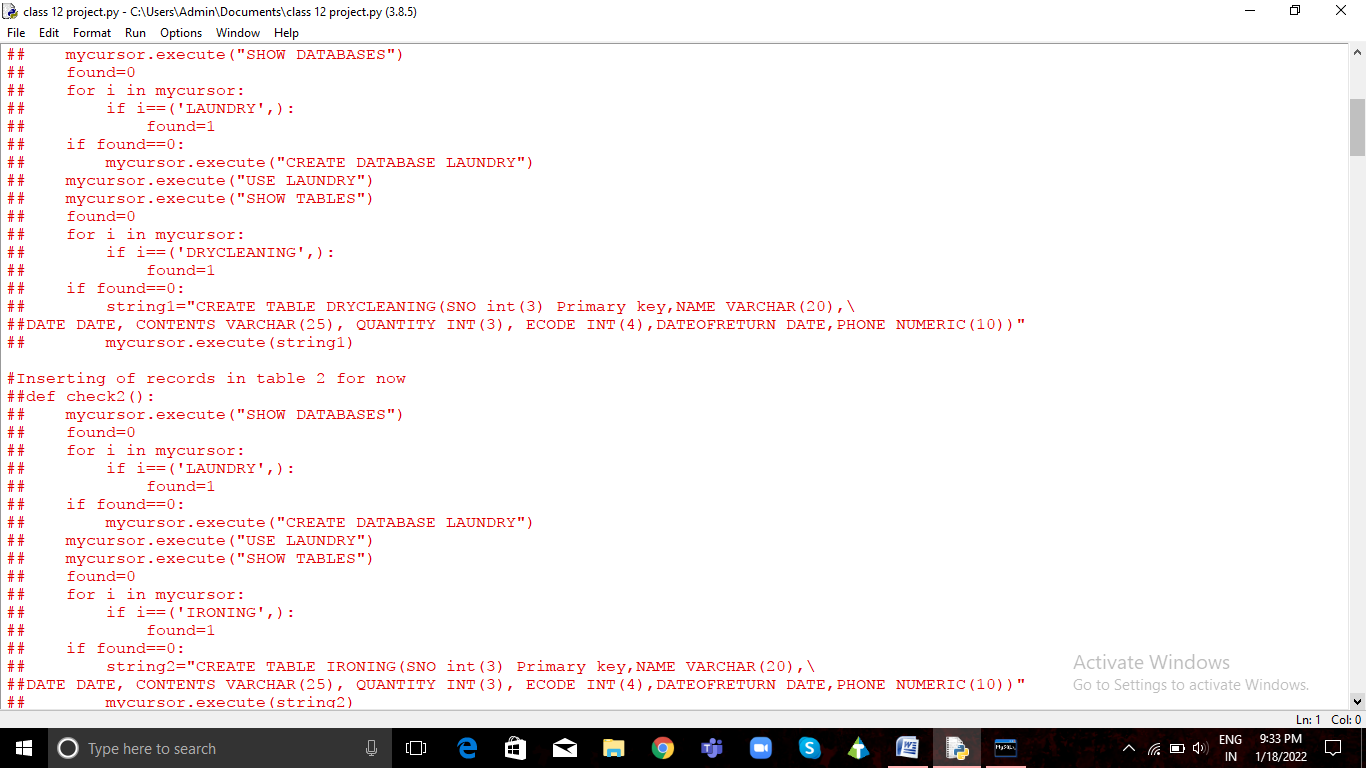
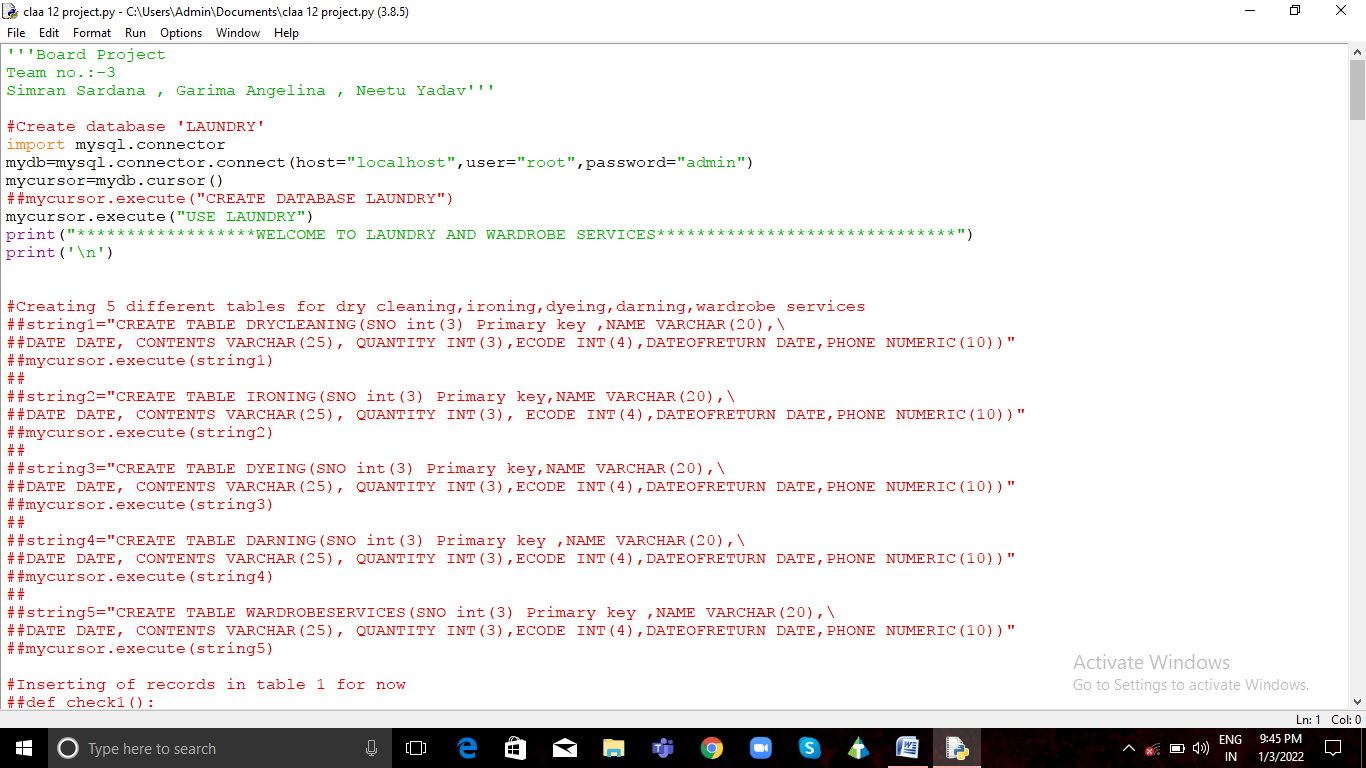
print("-----------------------THANKS FOR VISITING.HAVE A NICE DAY AHEAD.------------------------------")

break

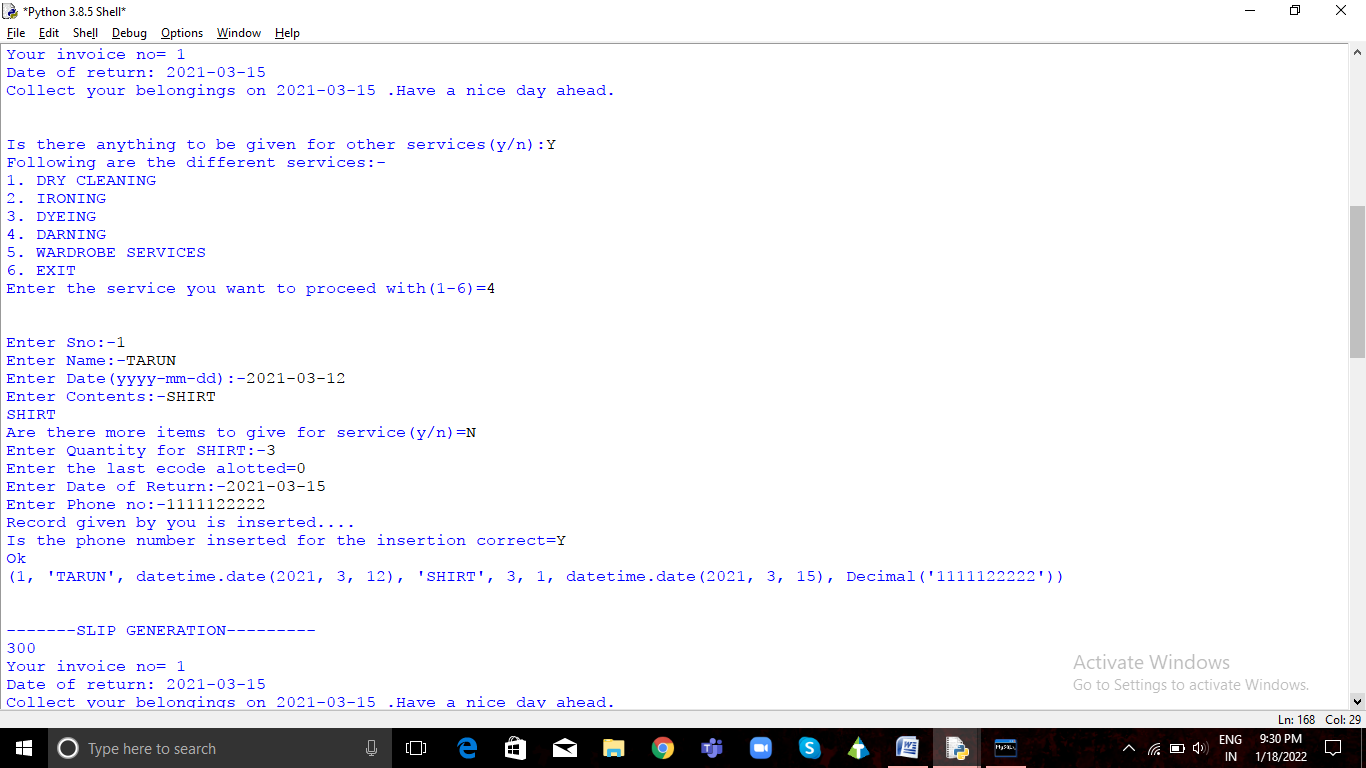
else:

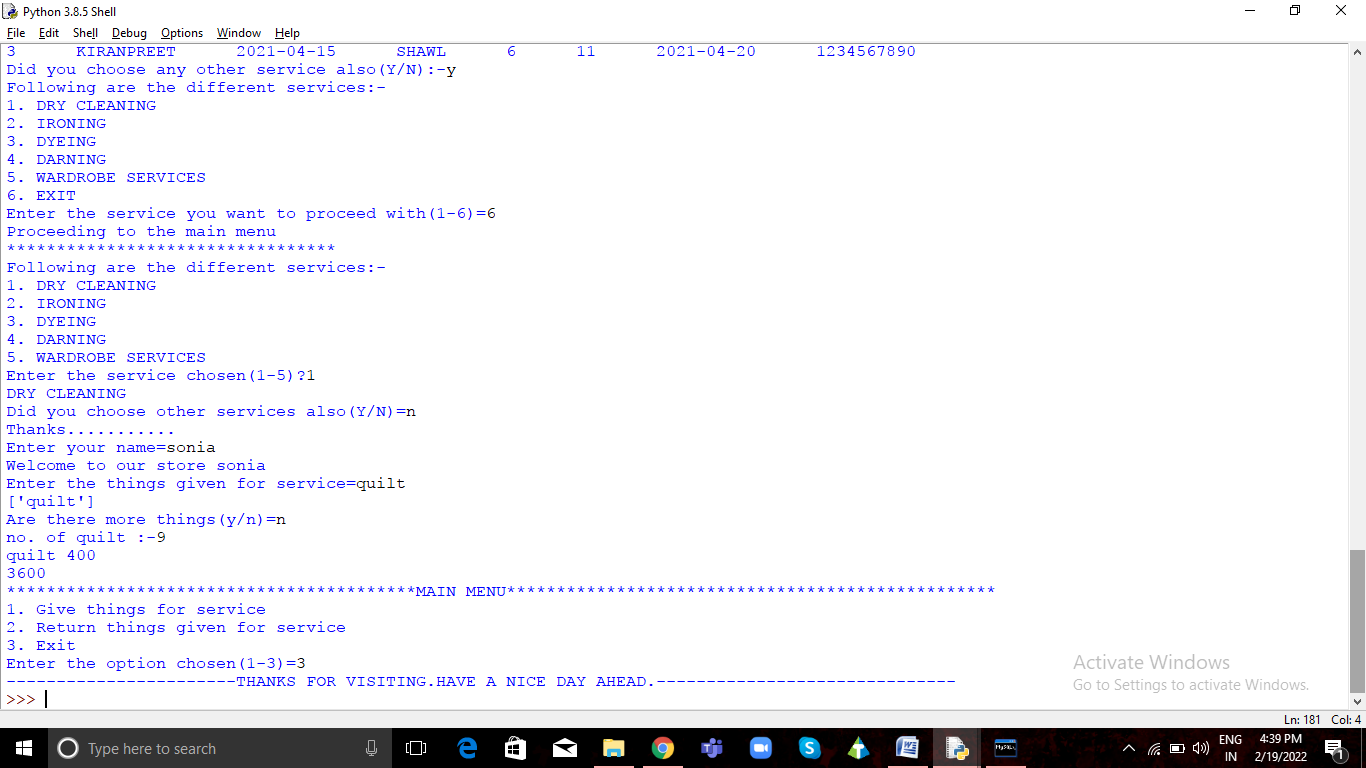
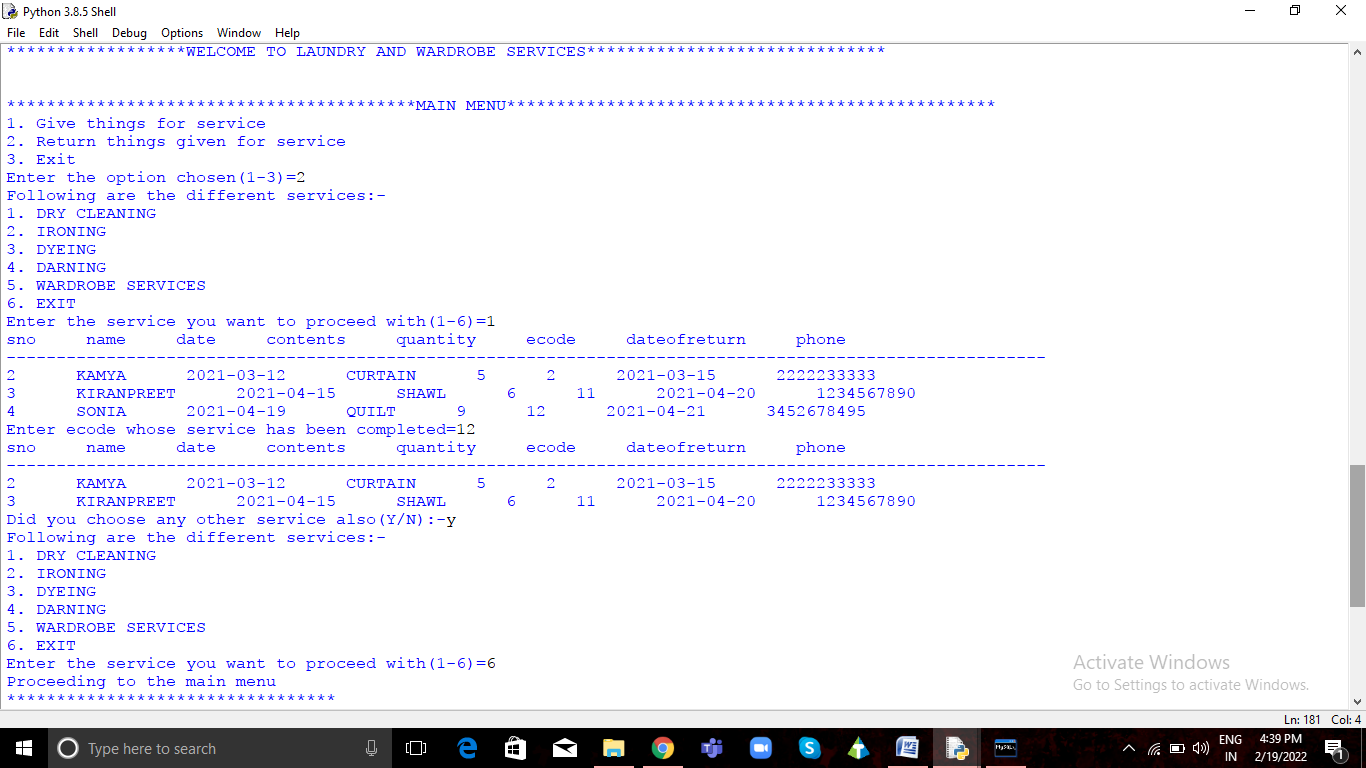
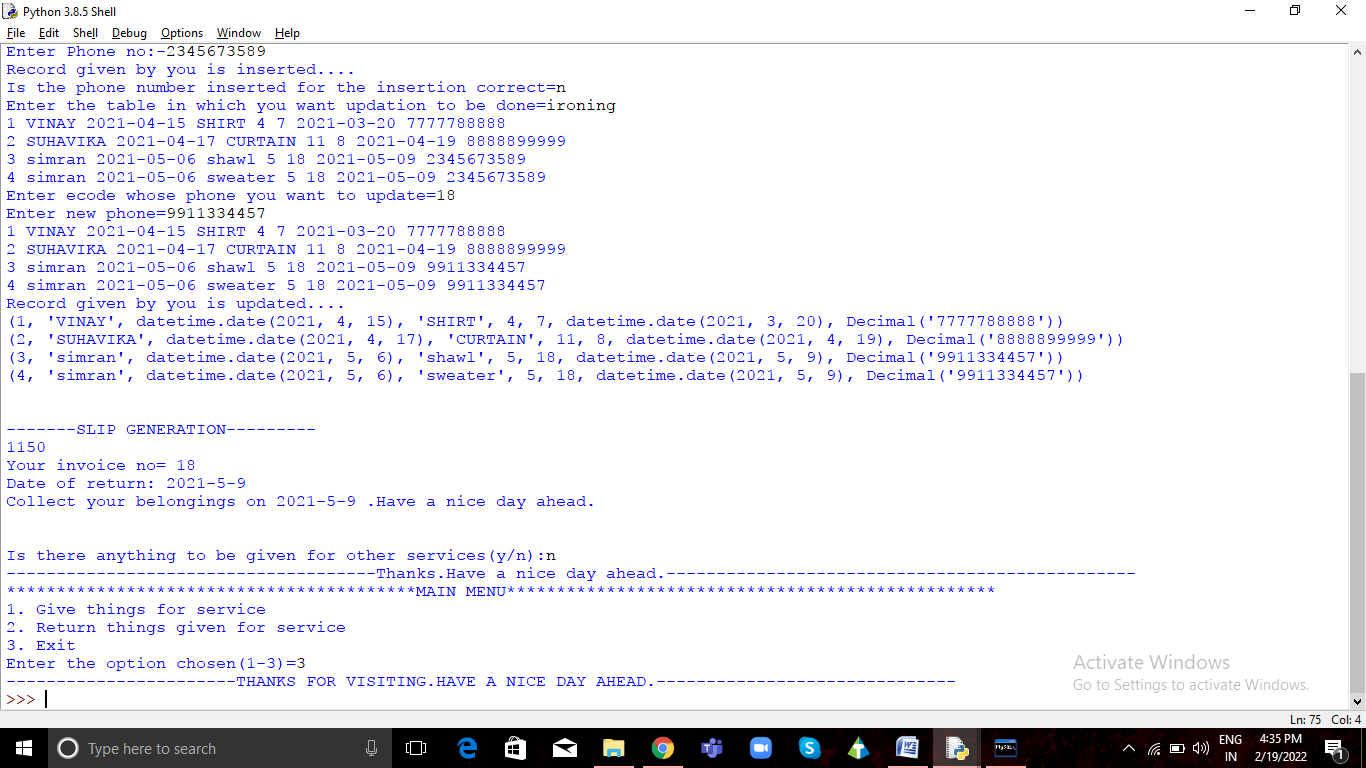
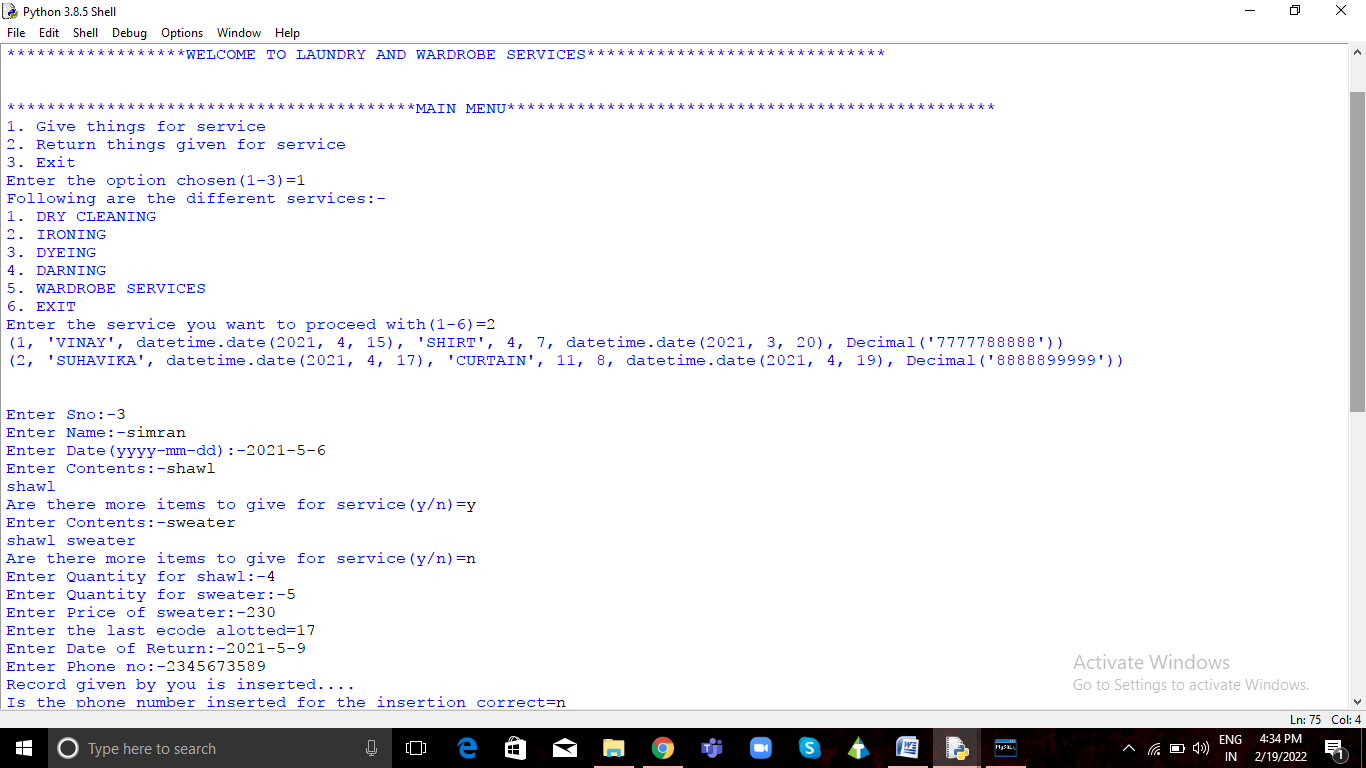
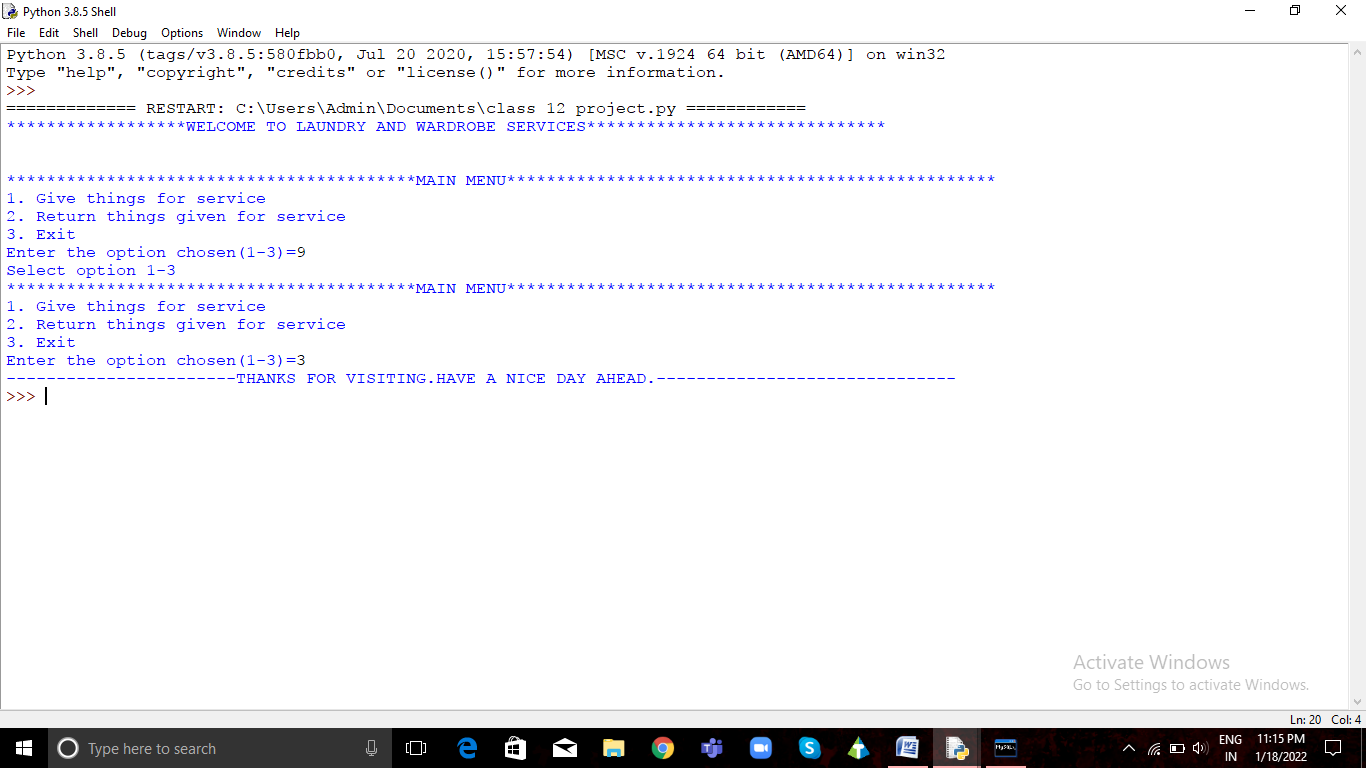
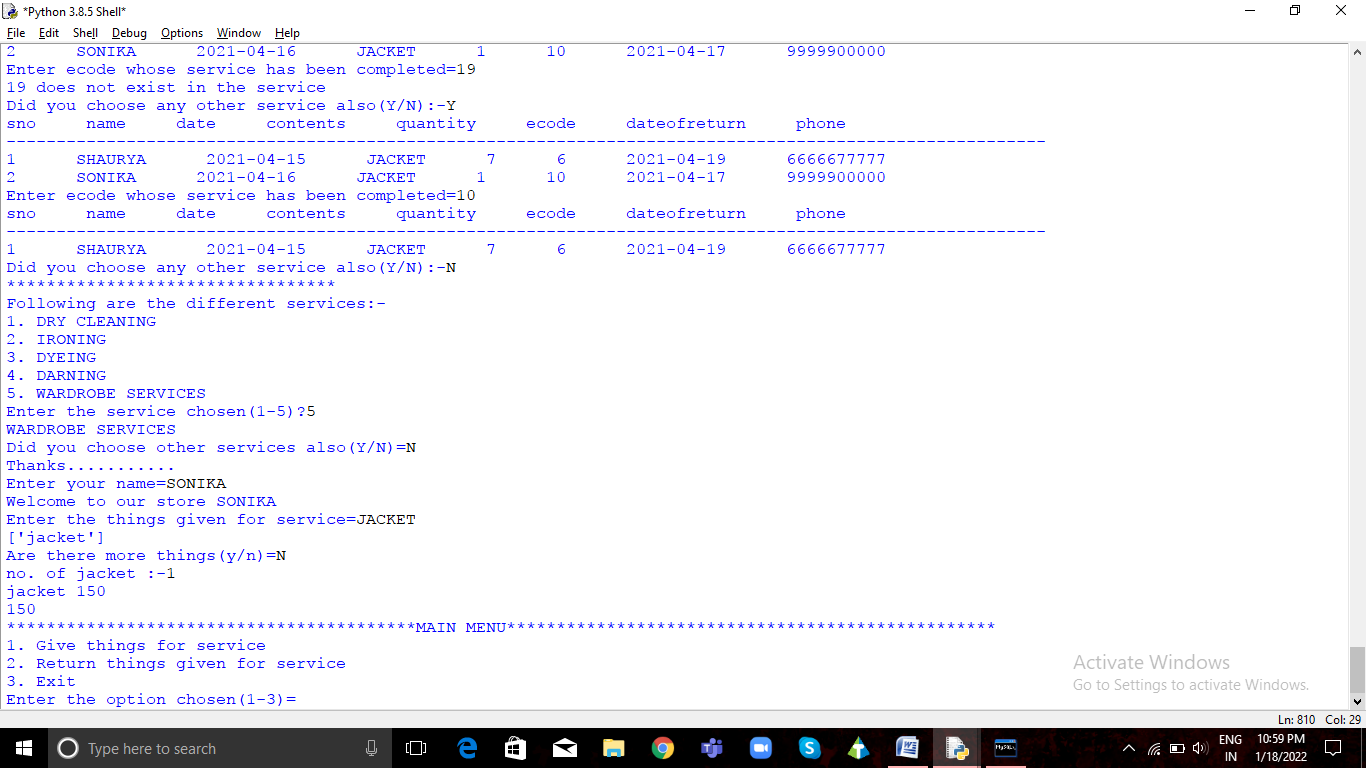
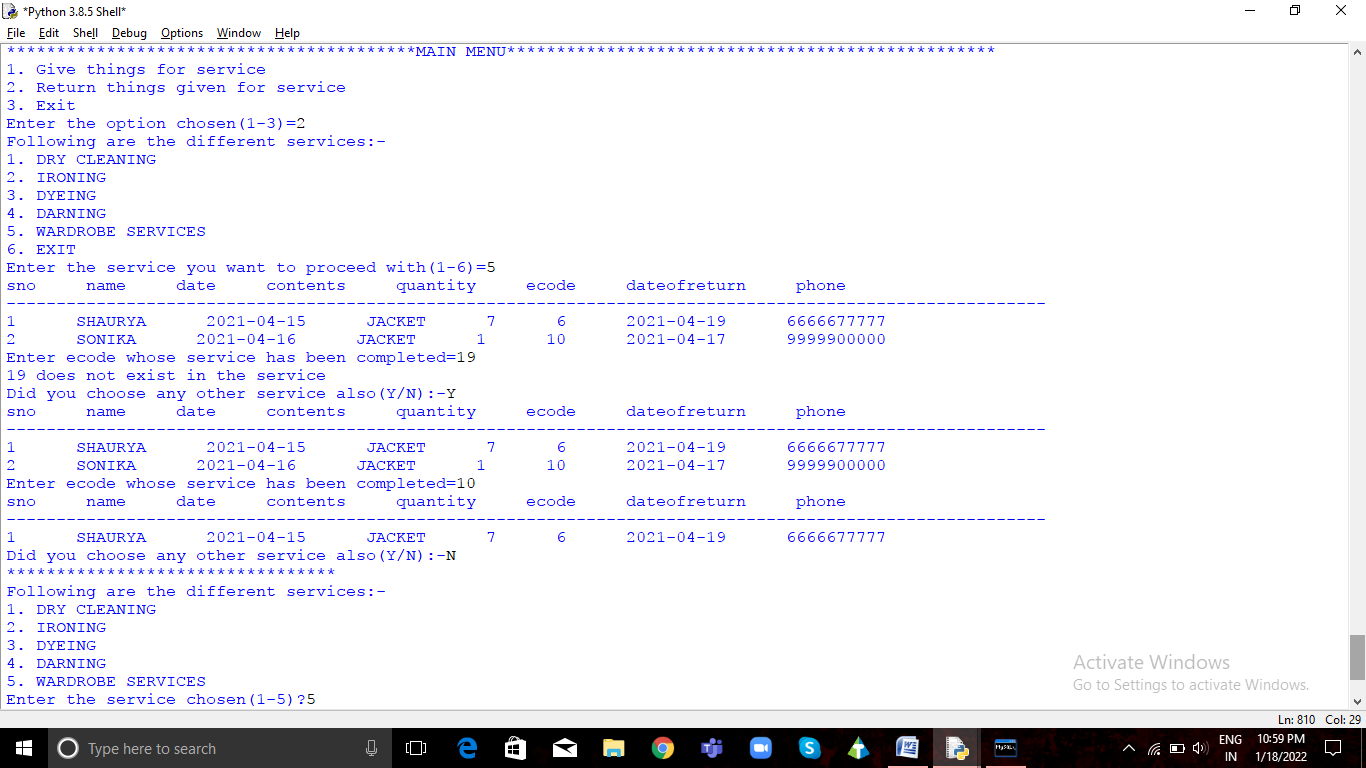
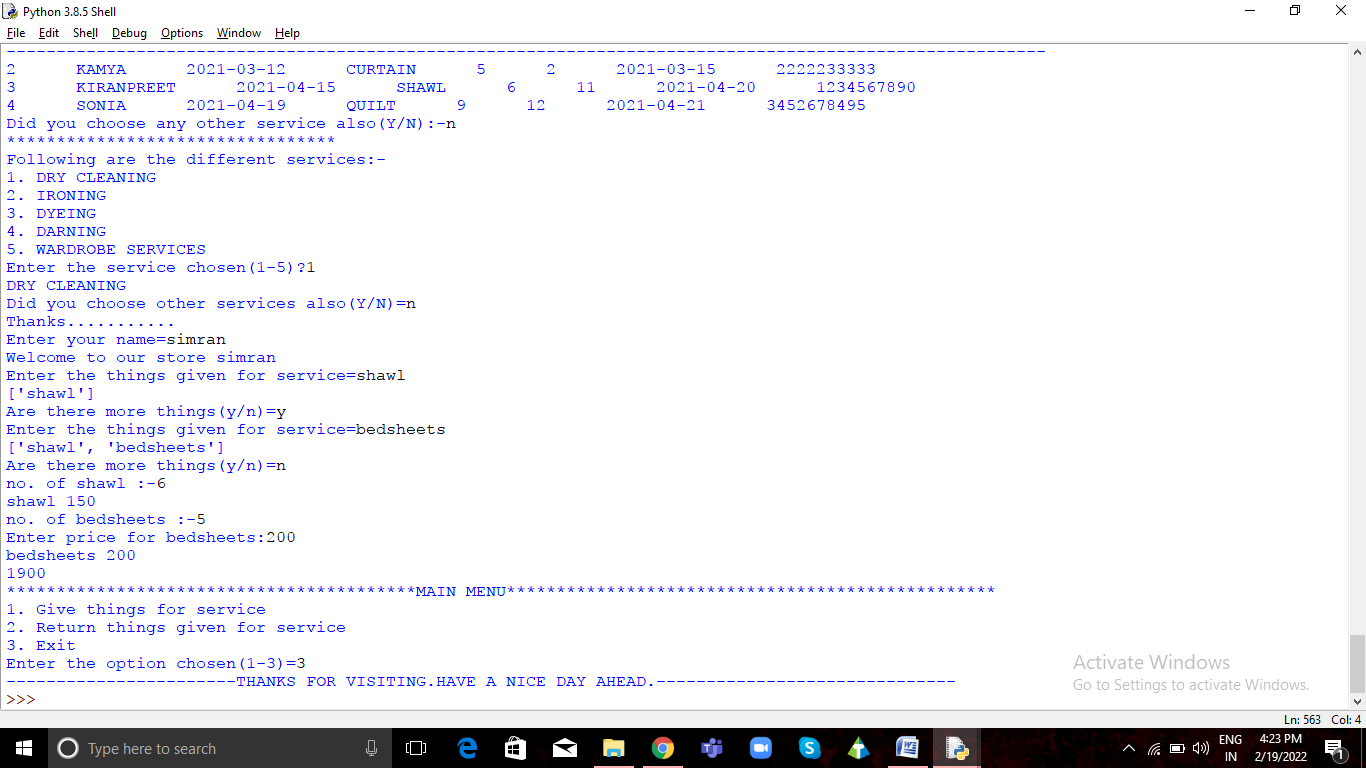
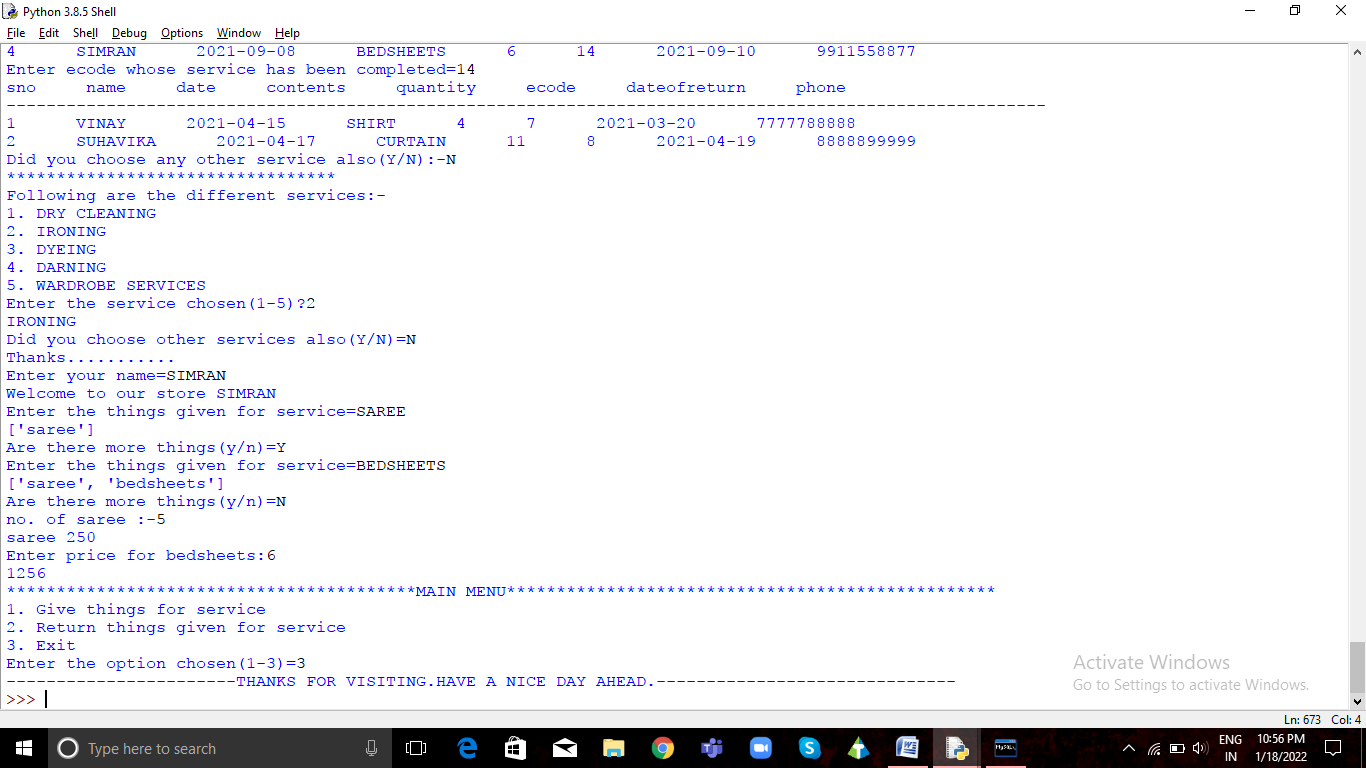
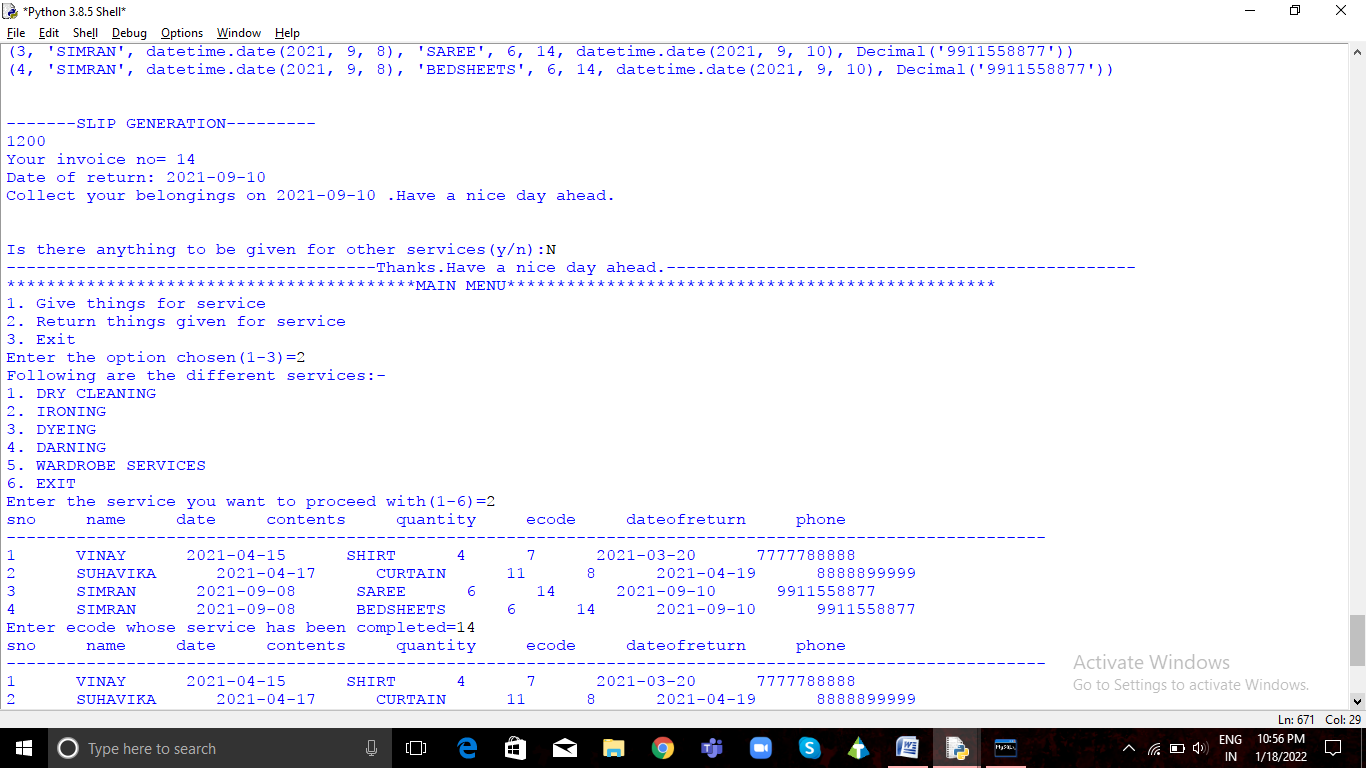
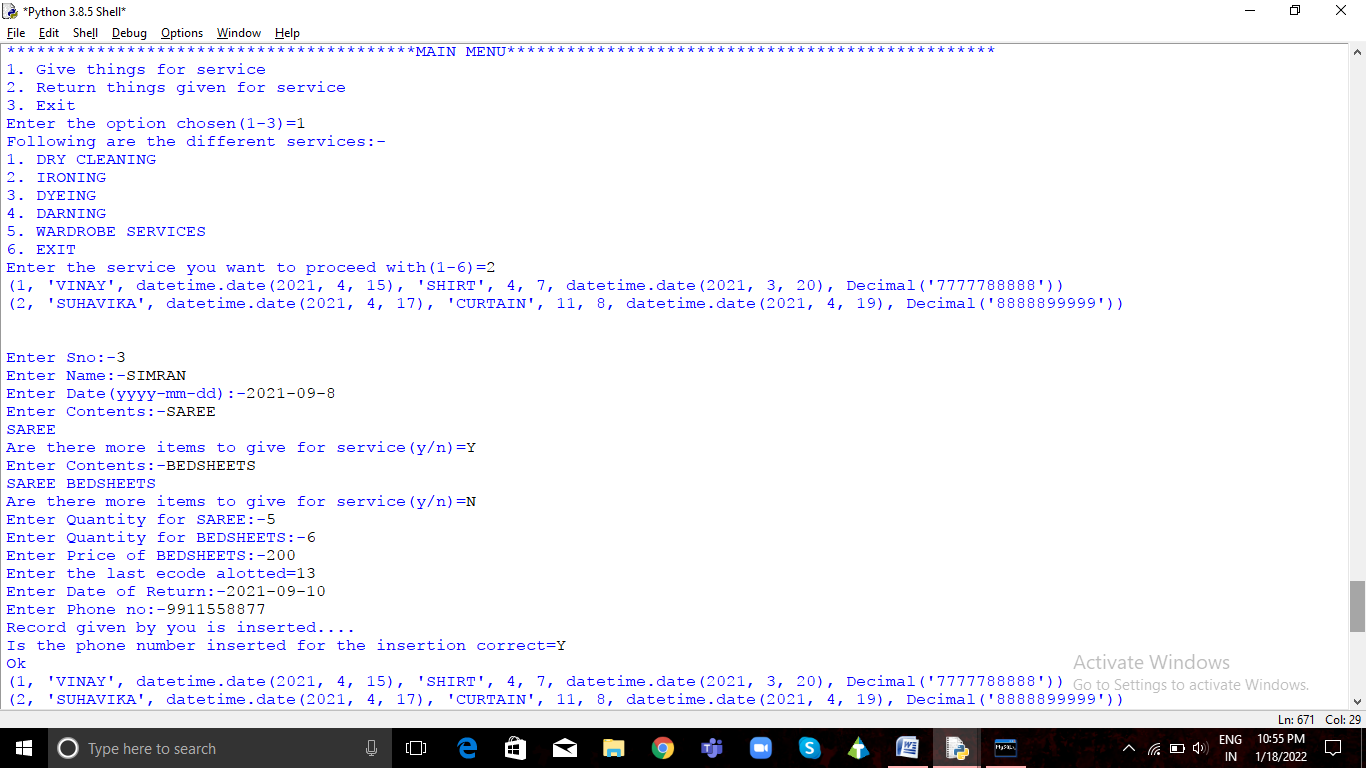
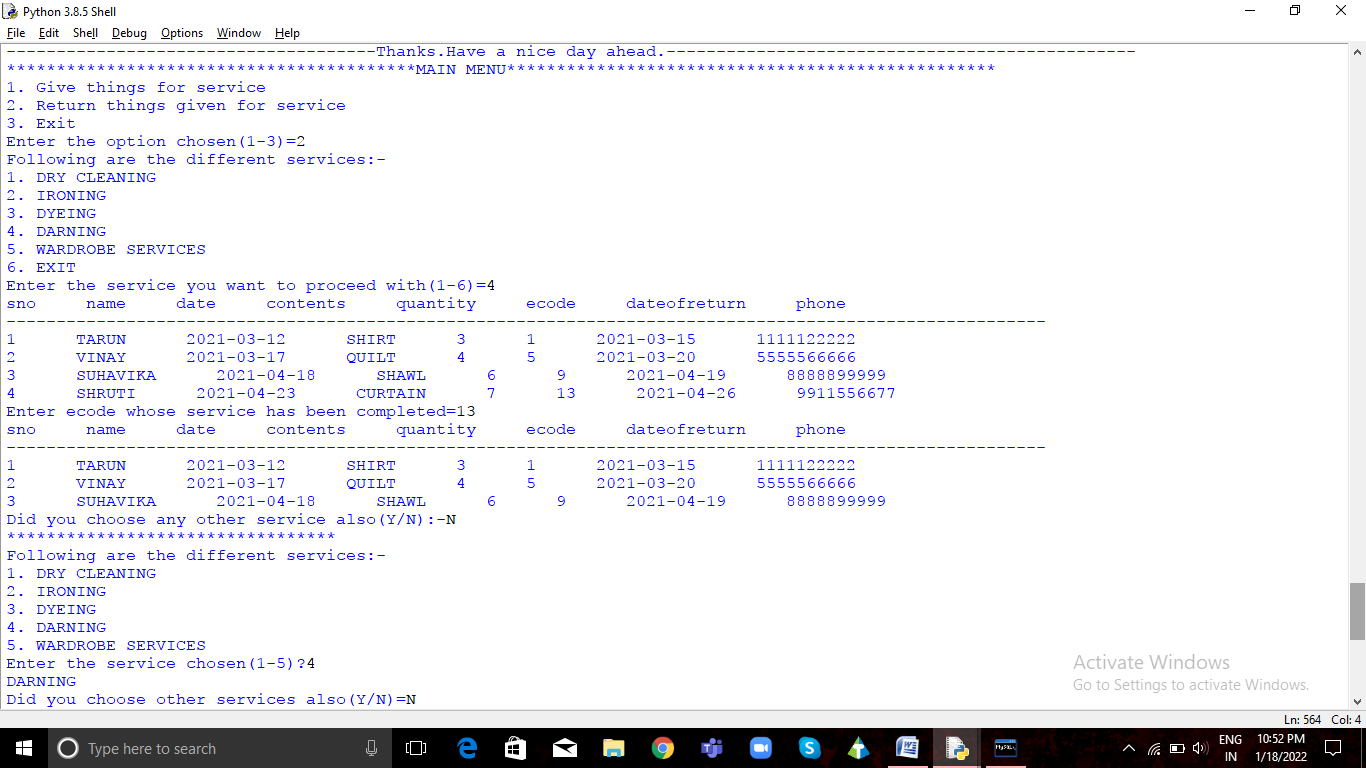
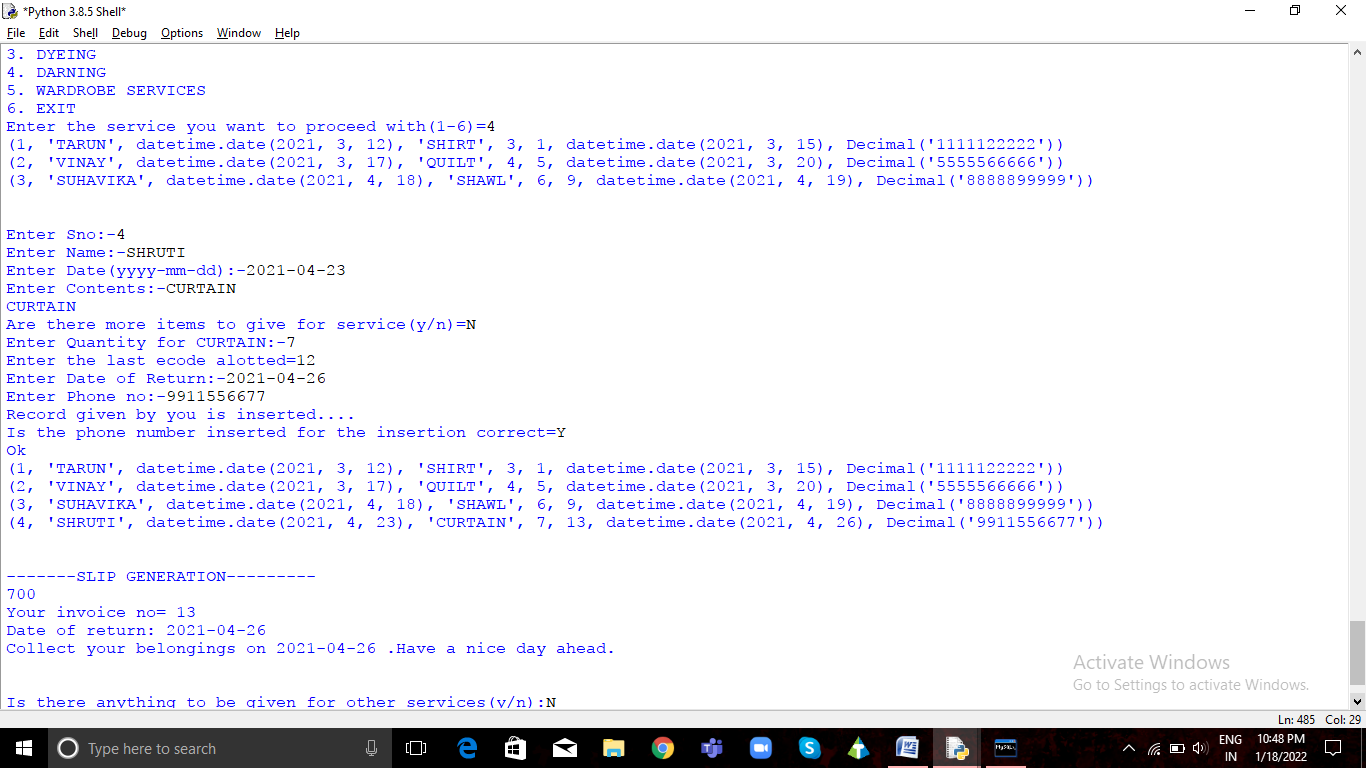
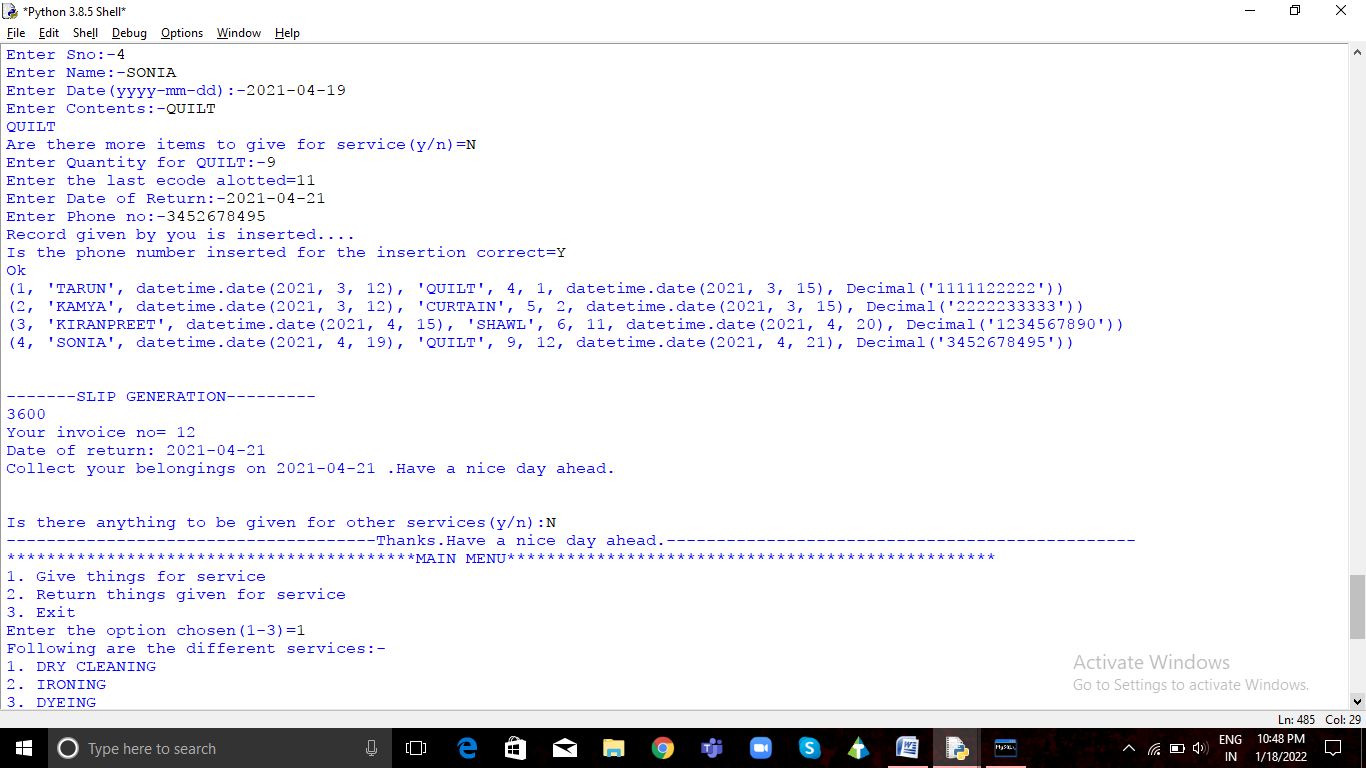
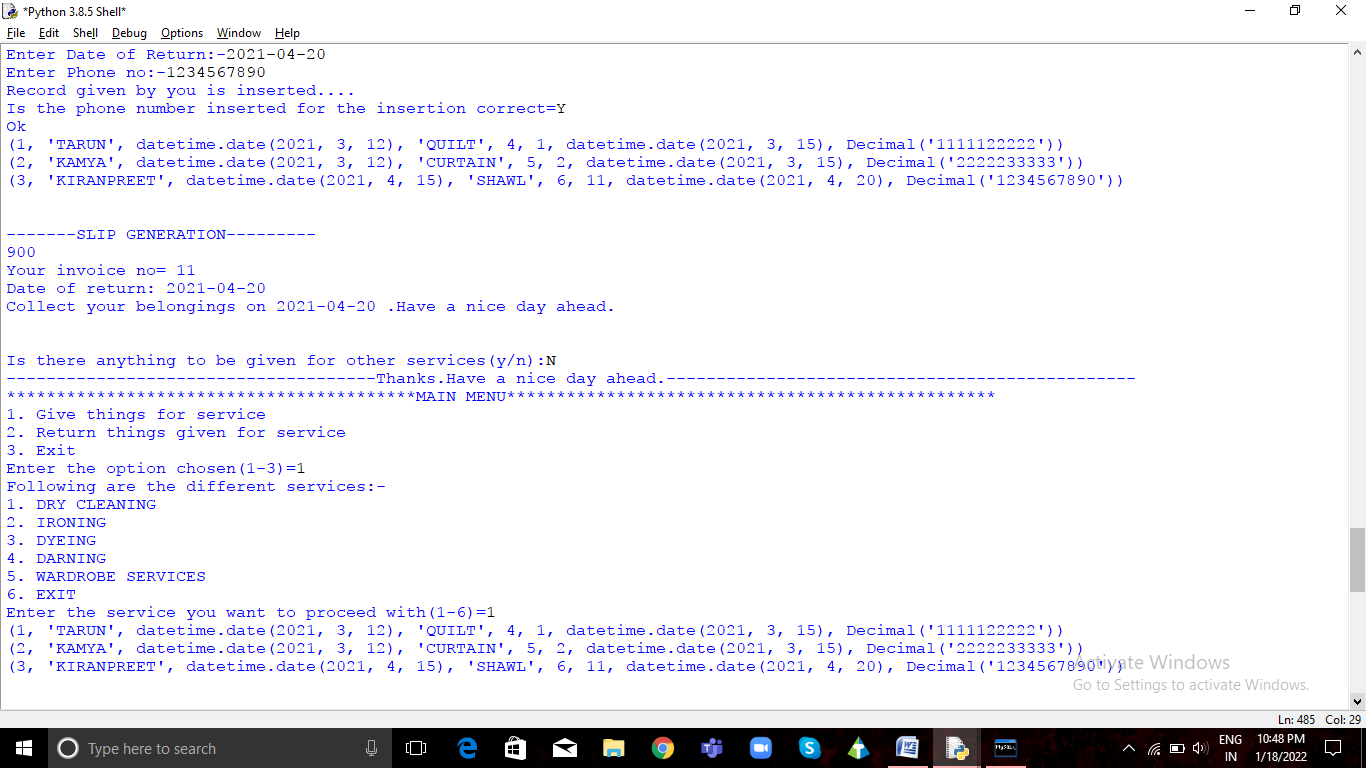
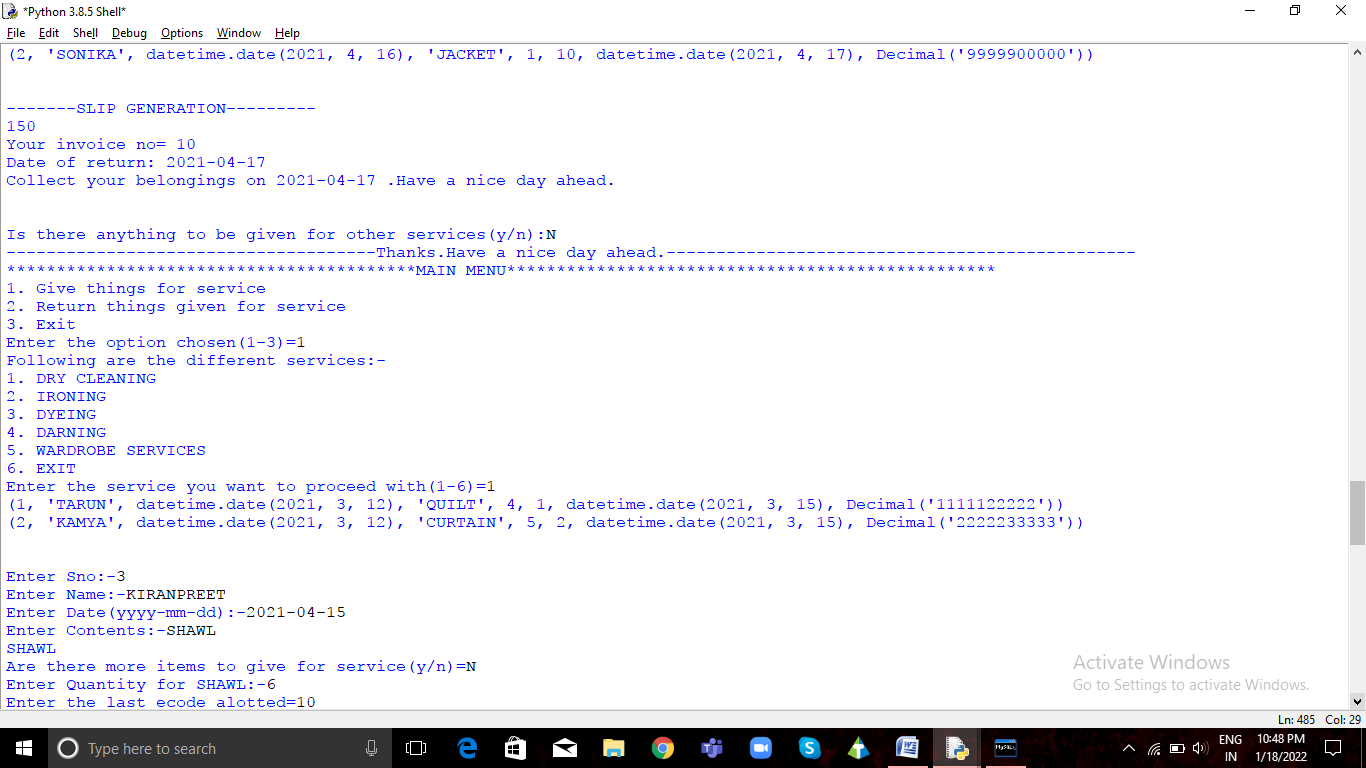
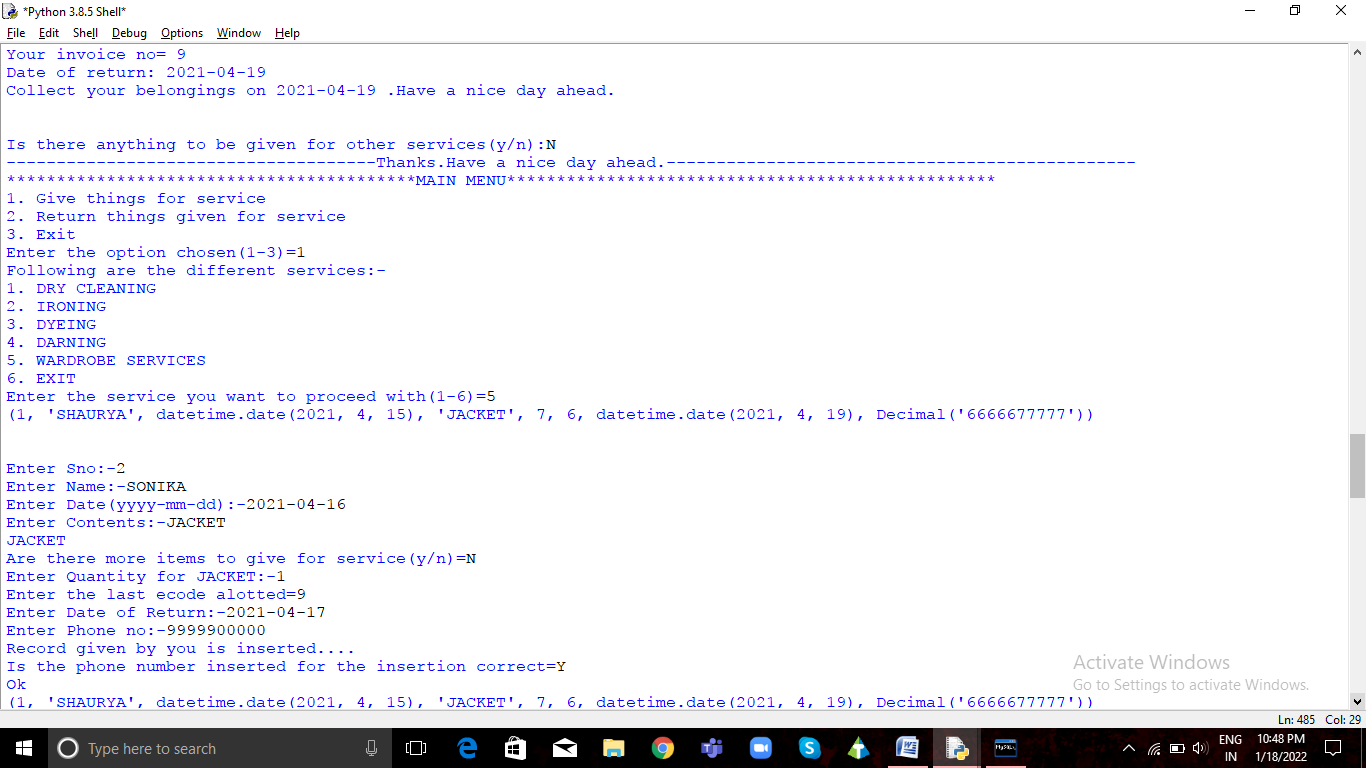
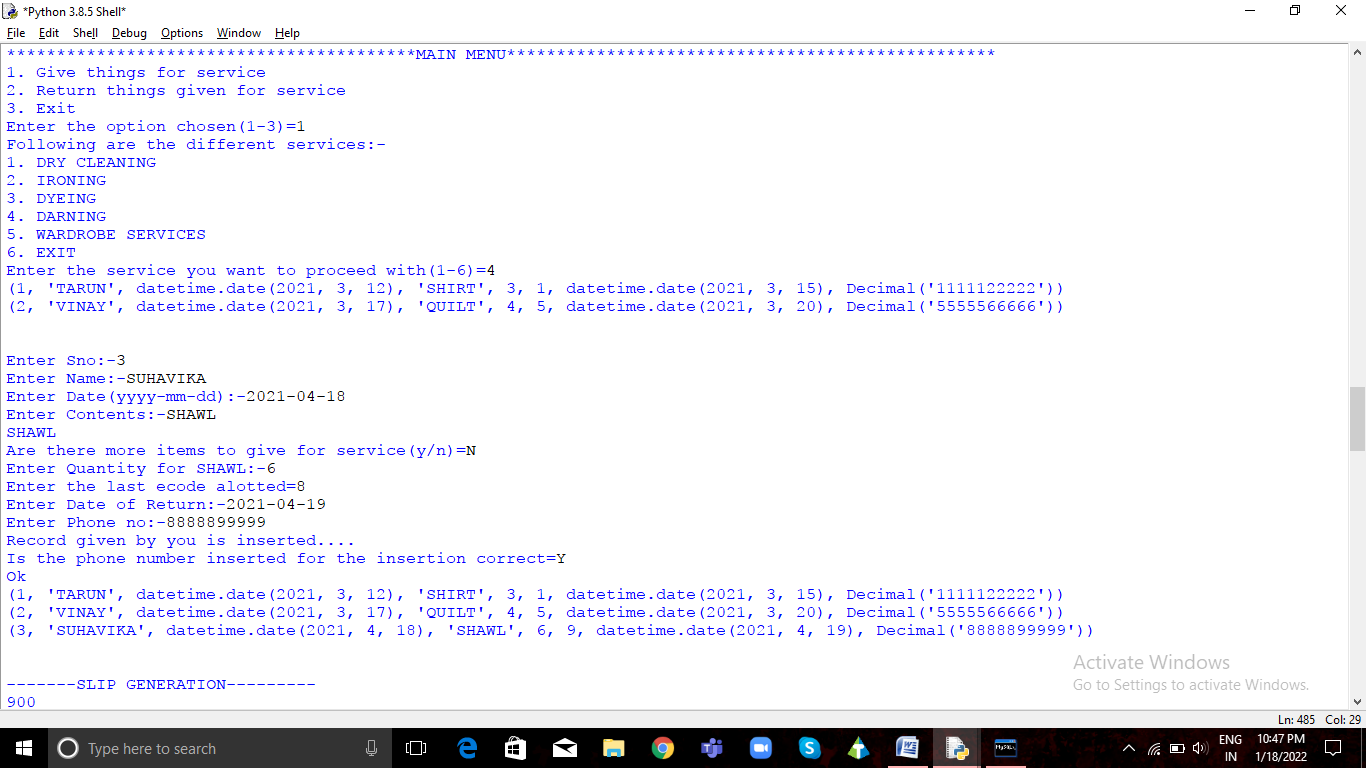
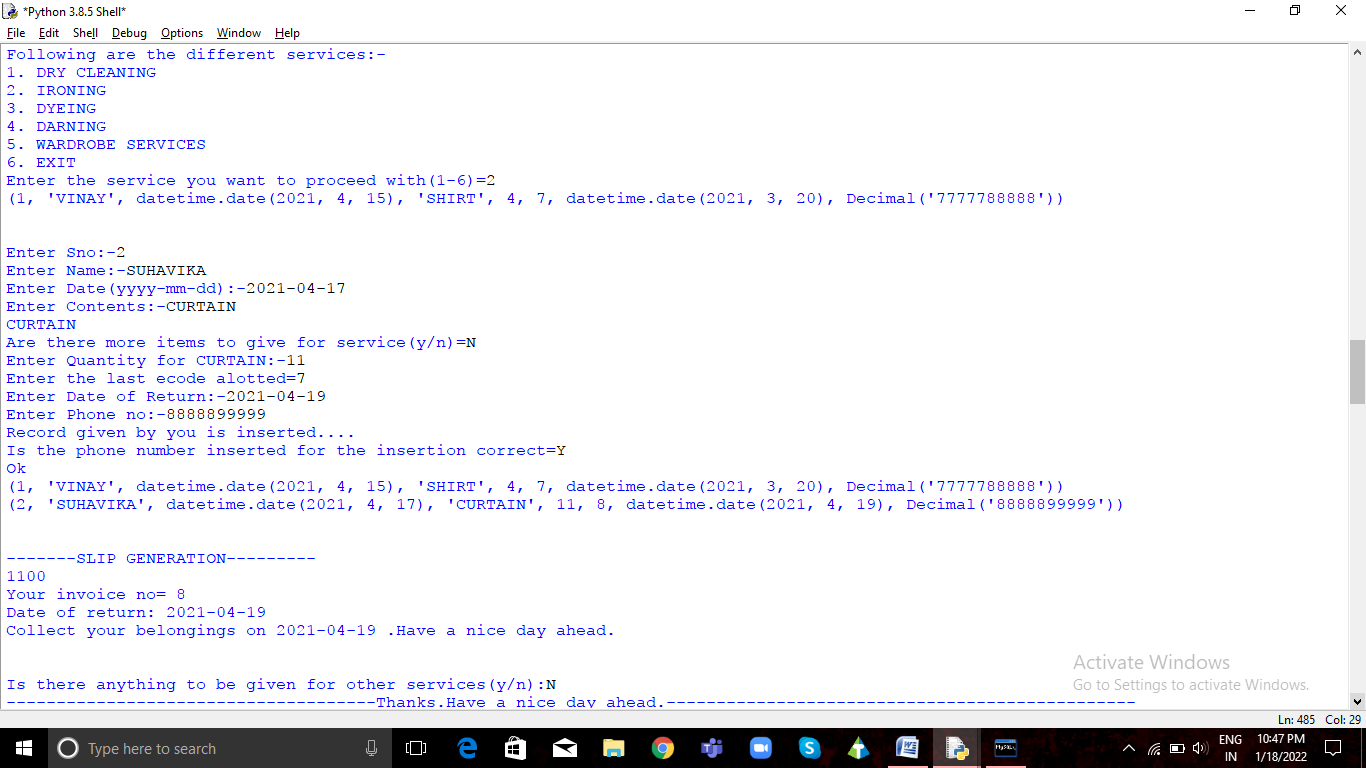
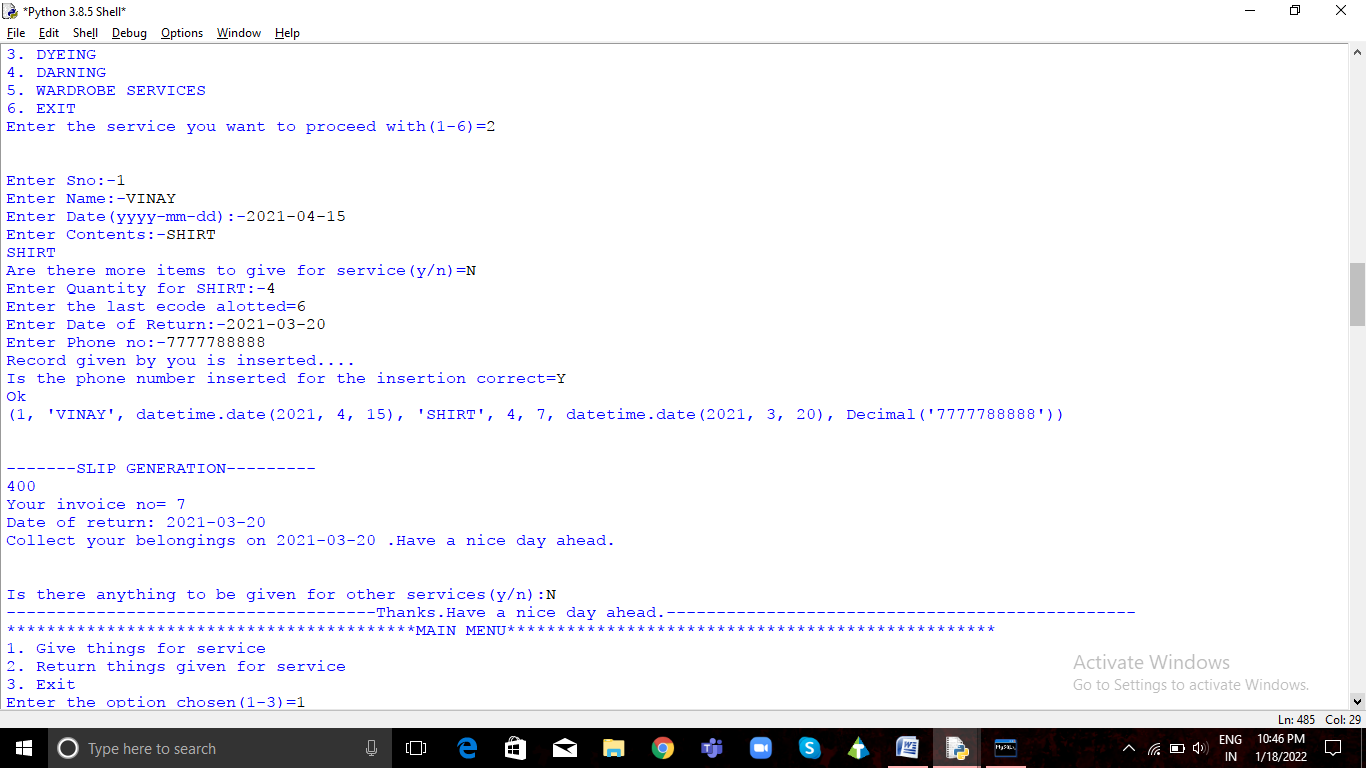
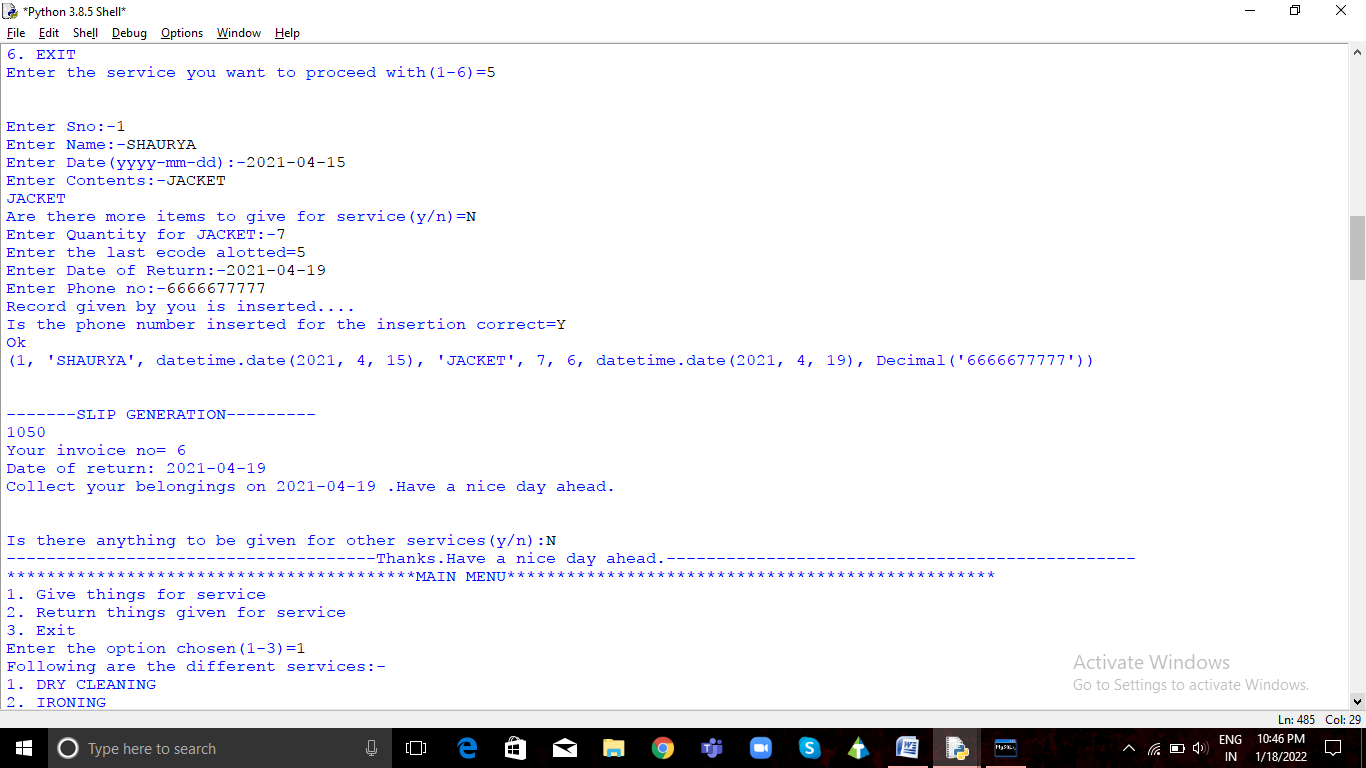
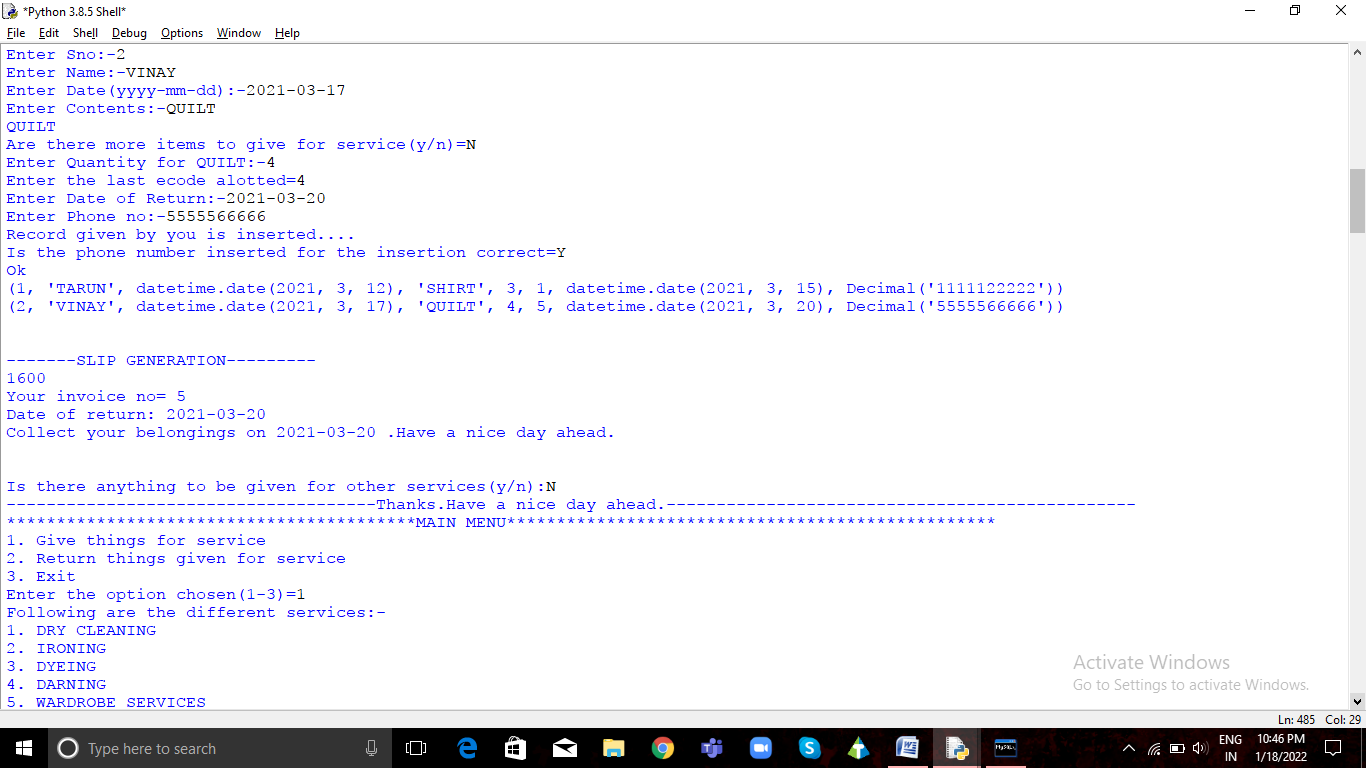
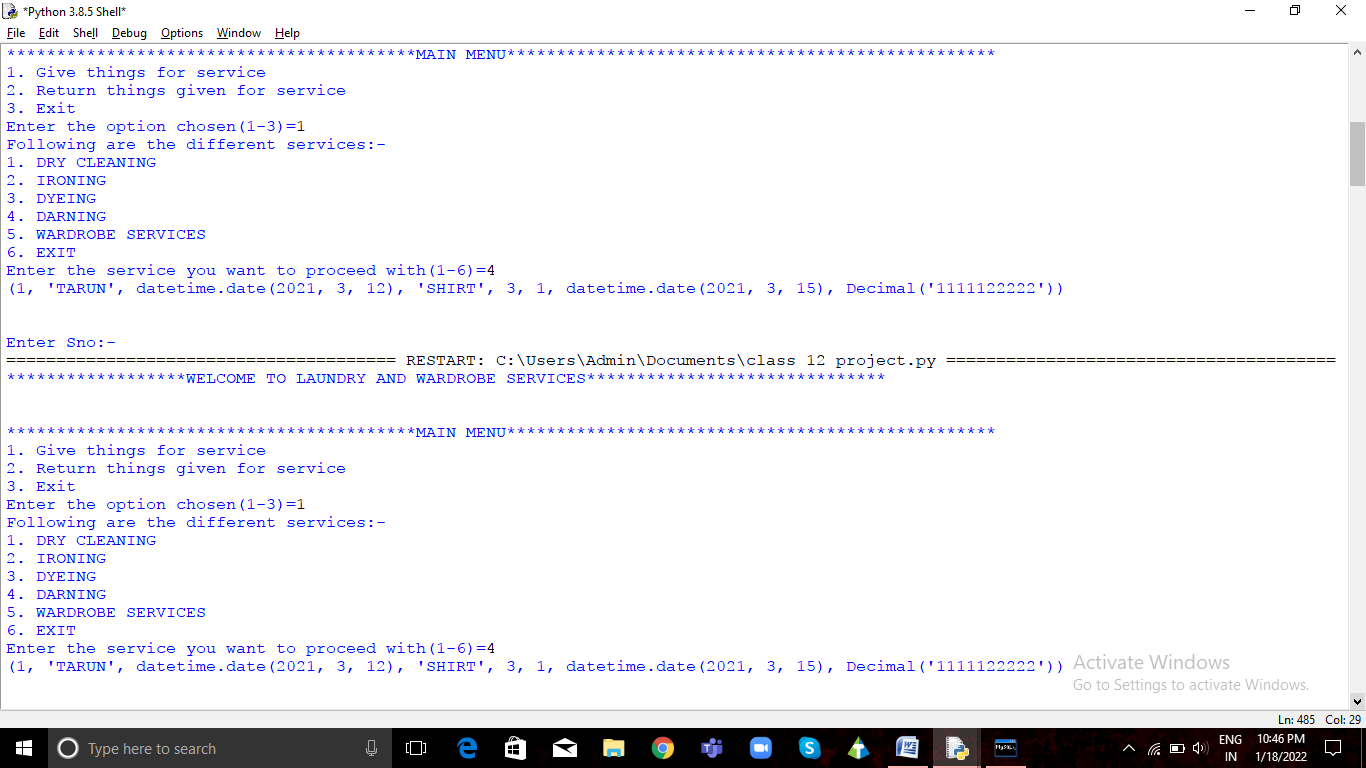
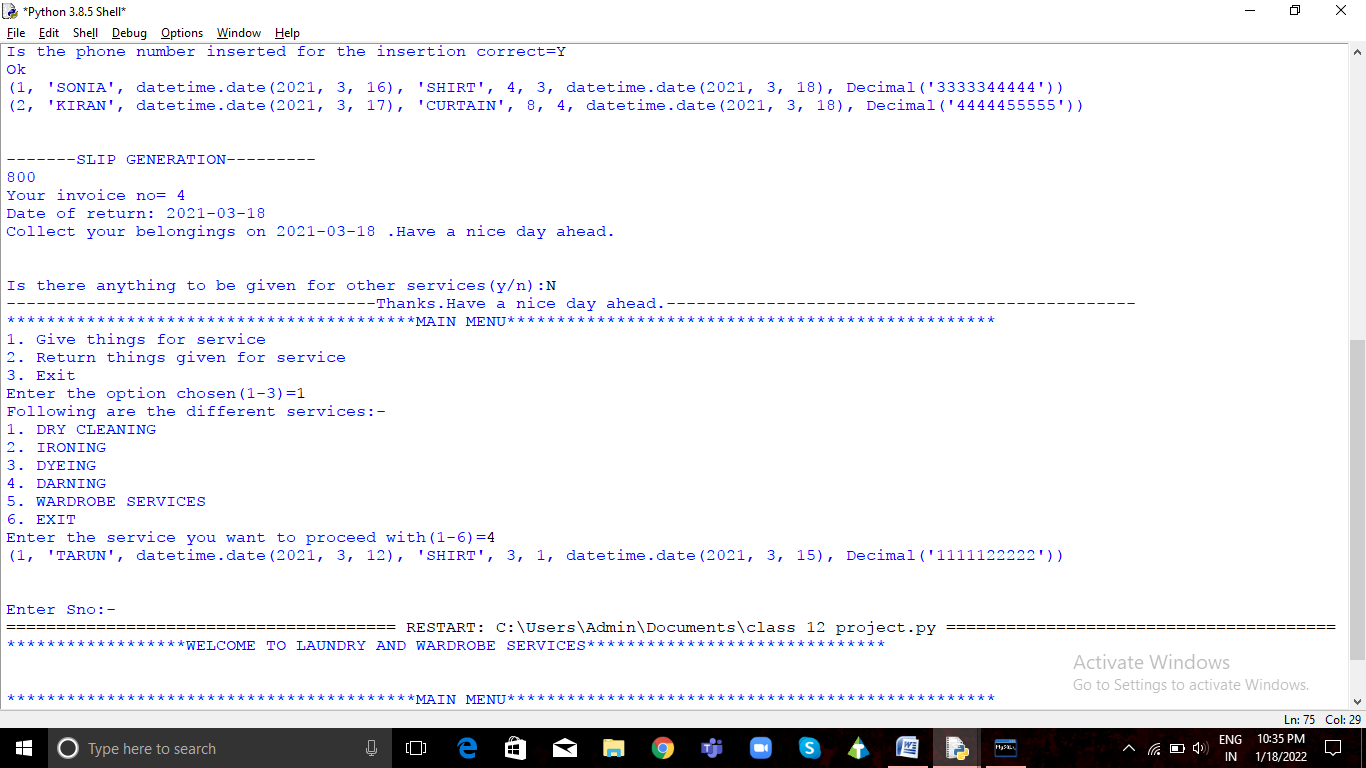
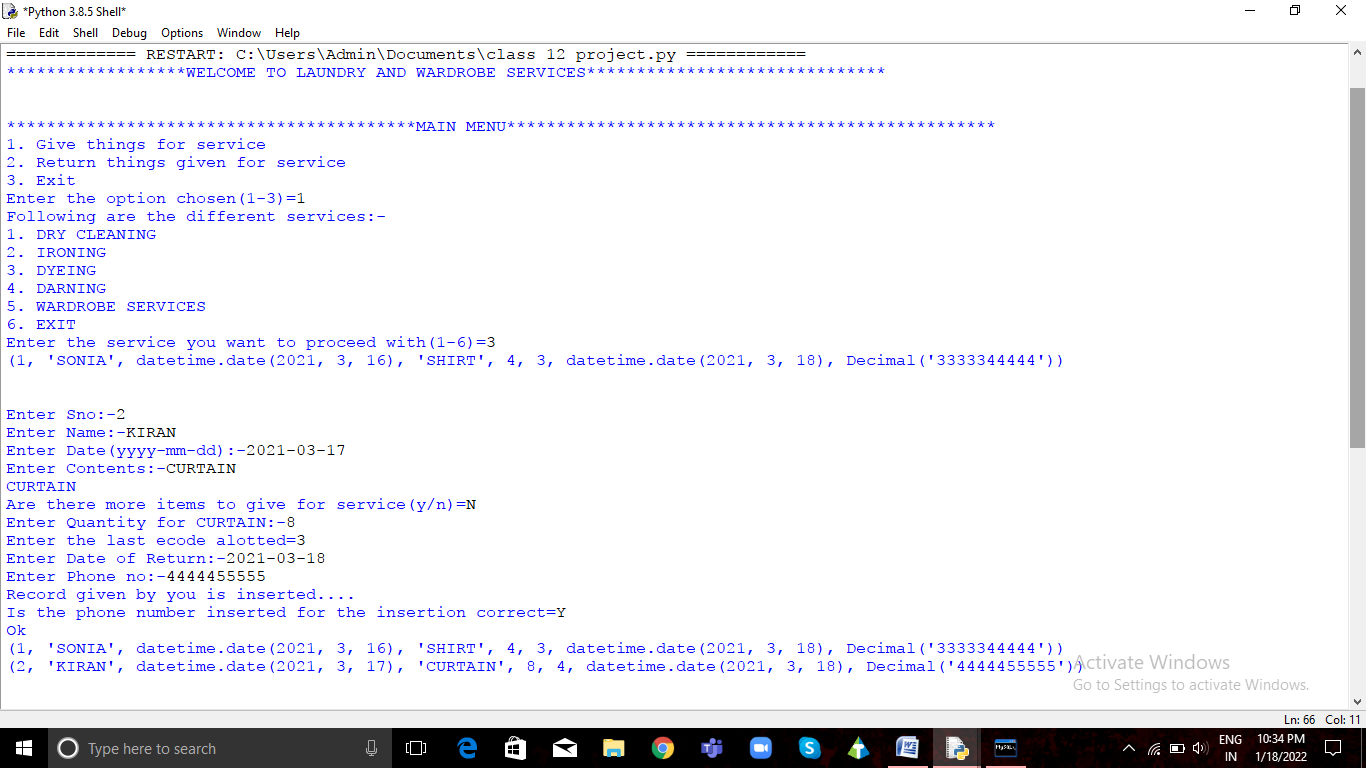
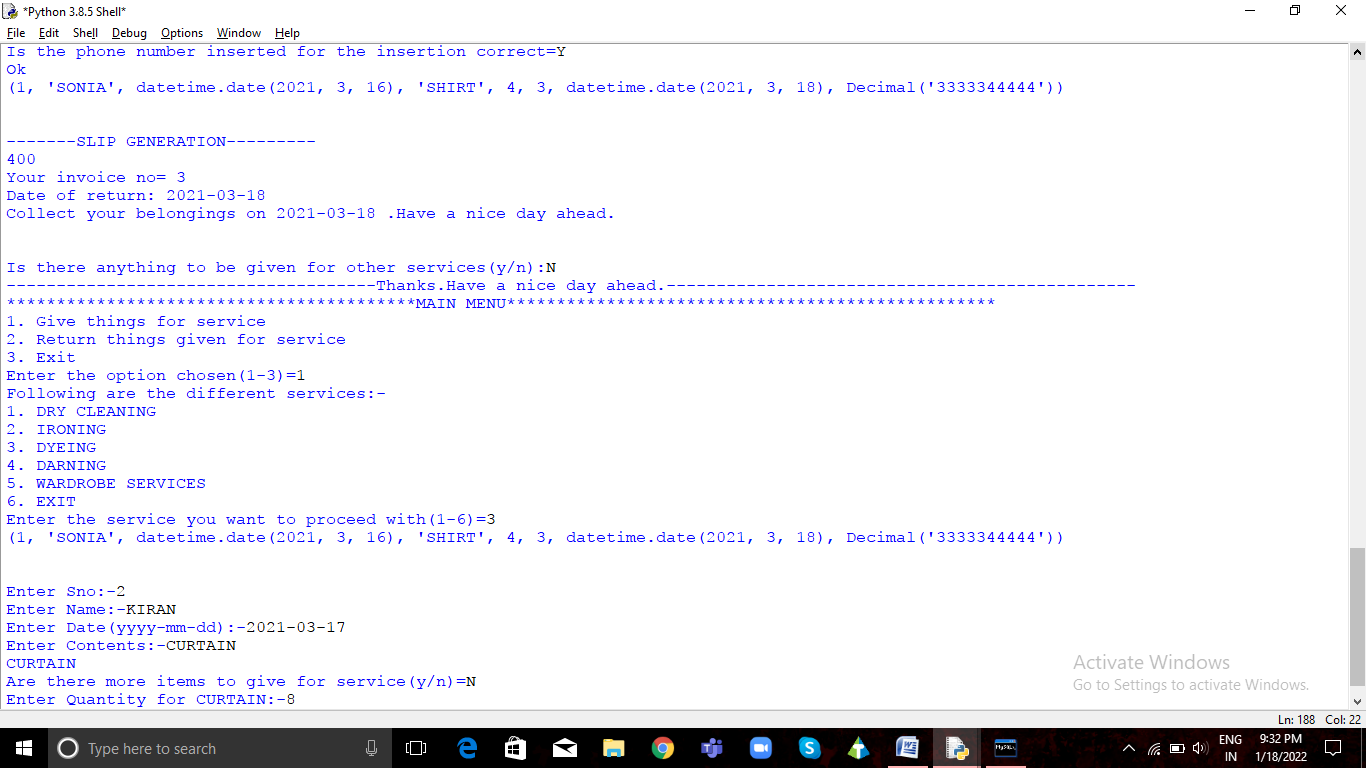
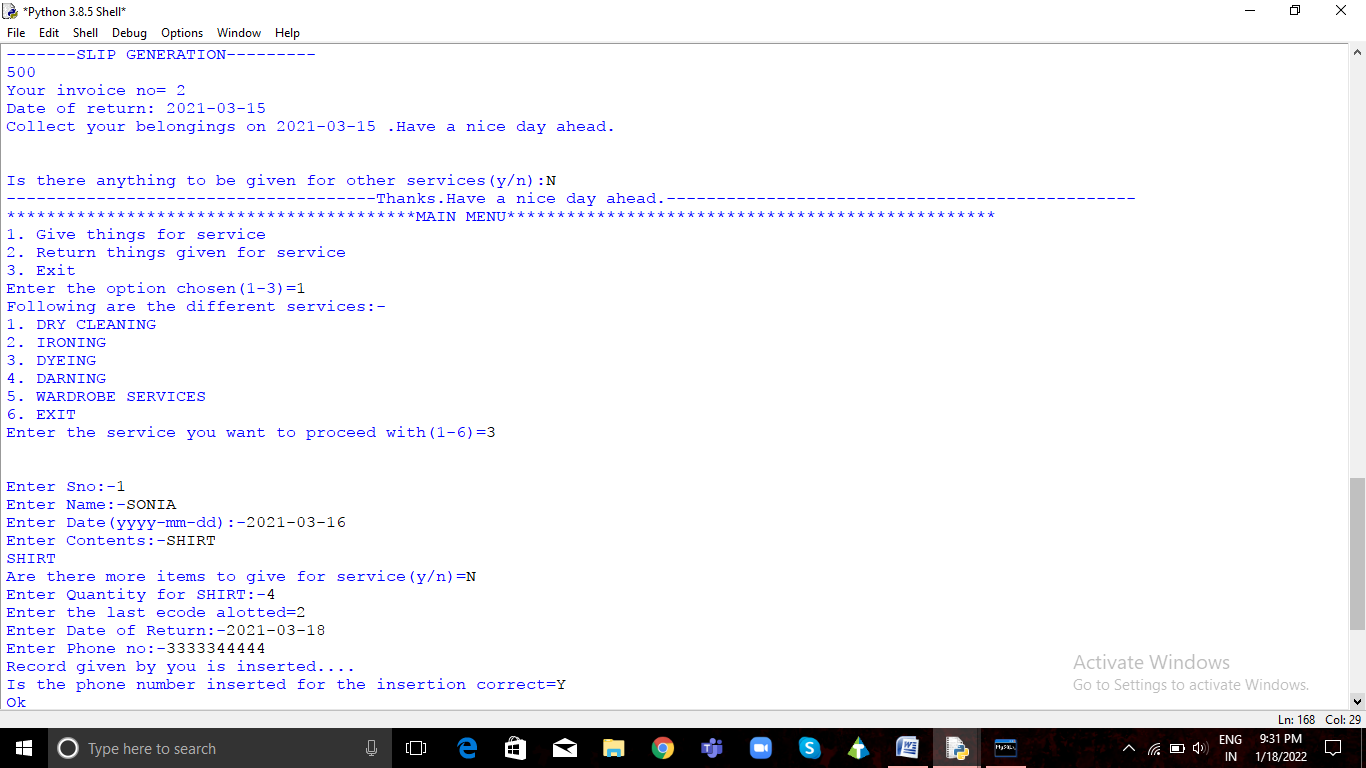
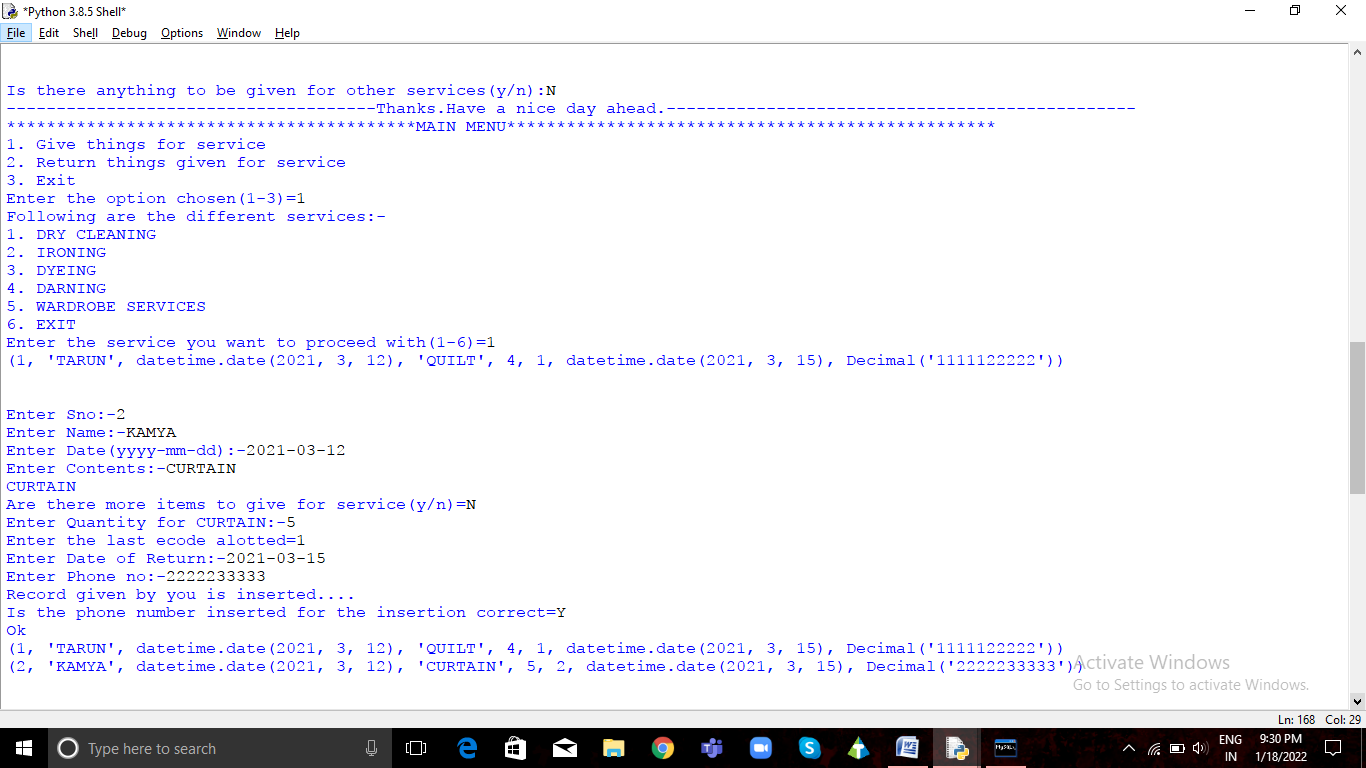
print("Select option 1-3")

continue



# OUTPUT SCREENSHOTS





# 

# BIBLIOGRAPHY

* NCERT COMPUTER SCIENCE CLASS XII
* PREETI ARORA CLASS XI
* PREETI ARORA CLASS XII
* Programs done in class