

PYTHON PROJECT

Delivery Management System (GUI)

Prepared By-
Rohan Mathur (B232)
Naman Gokharu (B214)

Submitted to-
Prof. Varsha Nemade

ABOUT THE PROJECT

The project consists of GUI implementation of “Delivery Management System”, where we can find the records of the important entities involved in this project ,like customer, branch , order delivery and delivery executive details.

It starts with a signup/ login window where one enters login id and password to access the DMS homepage.

After that DMS homepage opens where four buttons are there for- customer details, branch details, order delivery details and delivery executive details. Once we click on a button, a new window opens up asking the user to enter the respective details.

We can access all the details just by entering the respective id(for ex:- branch id for viewing the branch details and so on).

Following the buttons provided in this window:-

- Add button for adding the enteries into the database.
- Search button for searching the details using respective id.
- Update button for updating the information.
- Clear button for clearing the enteries from entry widget.
- Home page button for going back to the homepage.

After the GUI implementation we have also created an application for the above program as well.

DATABASE CONTENTS

Branch (b_id , b_name, b_city, street)

Branch_Contact(b_id , b_phone)

Branch_State(b_city, state)

Product (p_type, p_id, p_no, price, b_id)

Product_Name(p_no, p_name)

contains(del_id , p_id)

e_commerce (e_id, platform_name)

Order_Delivery (del_id, del_date, del_status, b_id, del_ex_id,
c_id, e_id)

Customer(c_id, c_name, city, street)

Customer_Contact(c_id, phone)

Delivers_to(c_id, del_ex_id)

Customer_State(city, state)

Delivery_Executive(del_ex_id, del_ex_name, b_id)

Delivery_Executive_Contact(del_ex_id, del_ex_phone)

GUI IMPLEMENTATION CODE:

```
import mysql.connector
import tkinter
from tkinter import *
from tkinter import messagebox
import datetime
from datetime import *

def onClickCustomer():
    def onClick3(num):
        conn = mysql.connector.connect(host='localhost',
user='root', password='legend_123',

database='mini_project_dms')
        cursor = conn.cursor()
        if (num == 1):
            try:
                b_id = int(e1.get())
                b_phone = int(e3.get())
            except:
                messagebox.showinfo('Alert', 'Branch id and
Phone should be a Number!')
                b_name = e2.get()
                street = e4.get()
                b_city = e5.get()
                state = e6.get()
                tup1 = (b_city, state)
                tup2 = (b_id, b_name, b_city, street)
                tup3 = (b_id, b_phone)
                s1 = "insert into
mini_project_dms.customer_state(city,state) values(%s,%s) "
                s2 = "insert into
mini_project_dms.customer(c_id,c_name,city,street)
values(%s,%s,%s,%s) "
                s3 = "insert into
mini_project_dms.customer_contact(c_id,phone)
values(%s,%s) "
            try:
                cursor.execute(s1, tup1)
                conn.commit()
```

Delivery Management System

```
except:
    conn.rollback()
try:
    cursor.execute(s2, tup2)
    conn.commit()
except:
    conn.rollback()
try:
    cursor.execute(s3, tup3)
    conn.commit()
    messagebox.showinfo('Success', 'Inserted
Successfully!')
    e1.delete(0, "end")
    e2.delete(0, "end")
    e3.delete(0, "end")
    e4.delete(0, "end")
    e5.delete(0, "end")
    e6.delete(0, "end")
except:
    conn.rollback()

elif (num == 2):
    b_id = int(e1.get())
    tup = (b_id)
    s = '''
        select c_id , c_name, phone,
street,city,state
        from mini_project_dms.customer_contact
natural join mini_project_dms.customer natural join
mini_project_dms.customer_state
        where c_id=%d order by phone DESC limit 1;
    '''
    try:
        cursor.execute(s % tup)
        res_tup = cursor.fetchall()
        res_lst = []
        for i in res_tup:
            for j in i:
                res_lst.append(j)
        e1.delete(0, "end")
        e2.delete(0, "end")
        e3.delete(0, "end")
        e4.delete(0, "end")
```

Delivery Management System

```
        e5.delete(0, "end")
        e6.delete(0, "end")
        e1.insert(0, res_lst[0])
        e2.insert(0, res_lst[1])
        e3.insert(0, res_lst[2])
        e4.insert(0, res_lst[3])
        e5.insert(0, res_lst[4])
        e6.insert(0, res_lst[5])
        conn.commit()
    except:
        conn.rollback()
elif (num == 3):
    try:
        b_id = int(e1.get())
        b_phone = int(e3.get())
    except:
        messagebox.showinfo('Alert', 'Branch id
should be a Number!')
    b_name = e2.get()
    street = e4.get()
    b_city = e5.get()
    state = e6.get()
    tup = (b_name, b_phone, street, b_city, state,
b_id)

    print(tup)
    s = '''
        update mini_project_dms.customer_contact bc
natural join mini_project_dms.customer b natural join
mini_project_dms.customer_state bs
        set
b.c_name=%s,bc.phone=%s,b.street=%s,bs.city=%s,bs.state=%s
where c_id=%s;
        '''
    try:
        cursor.execute(s, tup)
        conn.commit()
        messagebox.showinfo('Success', 'Updated
Successfully!')
        e1.delete(0, "end")
        e2.delete(0, "end")
        e3.delete(0, "end")
        e4.delete(0, "end")
        e5.delete(0, "end")
        e6.delete(0, "end")
```

Delivery Management System

```
        except:
            conn.rollback()
    elif (num == 4):
        e1.delete(0, "end")
        e2.delete(0, "end")
        e3.delete(0, "end")
        e4.delete(0, "end")
        e5.delete(0, "end")
        e6.delete(0, "end")
    elif (num == 5):
        root.destroy()
        DMS_Page()
    cursor.close()
    conn.close()

root = Tk()
f = Frame(root, width=620, height=500, bg="black")
root.title("Customer Details")
l1 = LabelFrame(f, text="Customer Details", width=400,
height=420, bg="black", fg="lightgreen",
font=('TIMES NEW ROMAN', 28,
'underline'))
l1.place(x=15, y=25)
labelframe2 = LabelFrame(f, text="", width=180,
height=400, bg="black", fg="lightgreen",
font=('TIMES NEW ROMAN', 28,
'underline'))
labelframe2.place(x=420, y=48)
l2 = Label(f, text="Customer ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l2.place(x=25, y=100)
l3 = Label(f, text="Name:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l3.place(x=25, y=160)
l4 = Label(f, text="Phone:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l4.place(x=25, y=220)
l5 = Label(f, text="Street:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l5.place(x=25, y=280)
l6 = Label(f, text="City:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l6.place(x=25, y=340)
```

Delivery Management System

```
17 = Label(f, text="State:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
17.place(x=25, y=400)
e1 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e1.place(x=150, y=100)
e2 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e2.place(x=150, y=160)
e3 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e3.place(x=150, y=220)
e4 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e4.place(x=150, y=280)
e5 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e5.place(x=150, y=340)
e6 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e6.place(x=150, y=400)
b1 = Button(f, text="Add", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
command=lambda: onClick3(1))
b1.place(x=450, y=100)
b2 = Button(f, text="Search", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
command=lambda: onClick3(2))
b2.place(x=450, y=170)
b3 = Button(f, text="Update", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
command=lambda: onClick3(3))
b3.place(x=450, y=240)
b4 = Button(f, text="Clear", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
command=lambda: onClick3(4))
b4.place(x=450, y=310)
b5 = Button(f, text="Home Page", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
command=lambda: onClick3(5))
b5.place(x=450, y=380)
f.pack()
root.mainloop()
```


Delivery Management System

```
def onClickBranch():
    def onClick3(num):
        conn = mysql.connector.connect(host='localhost',
user='root', password='legend_123',

database='mini_project_dms')
        cursor = conn.cursor()
        if (num == 1):
            try:
                b_id = int(e1.get())
                b_phone = int(e3.get())
            except:
                messagebox.showinfo('Alert', 'Branch id and
Phone should be a Number!')
                b_name = e2.get()
                street = e4.get()
                b_city = e5.get()
                state = e6.get()
                tup1 = (b_city, state)
                tup2 = (b_id, b_name, b_city, street)
                tup3 = (b_id, b_phone)
                s1 = "insert into
mini_project_dms.Branch_State(b_city,state) values(%s,%s)"
                s2 = "insert into
mini_project_dms.Branch(b_id,b_name,b_city,street)
values(%s,%s,%s,%s)"
                s3 = "insert into
mini_project_dms.Branch_Contact(b_id,b_phone)
values(%s,%s)"
            try:
                cursor.execute(s1, tup1)
                conn.commit()
            except:
                conn.rollback()
            try:
                cursor.execute(s2, tup2)
                conn.commit()
            except:
                conn.rollback()
            try:
                cursor.execute(s3, tup3)
                conn.commit()
                messagebox.showinfo('Success', 'Inserted
```

Delivery Management System

```
Successfully!')
        e1.delete(0, "end")
        e2.delete(0, "end")
        e3.delete(0, "end")
        e4.delete(0, "end")
        e5.delete(0, "end")
        e6.delete(0, "end")
    except:
        conn.rollback()

    elif (num == 2):
        b_id = int(e1.get())
        tup = (b_id)
        s = '''
            select b_id , b_name, b_phone,
street,b_city,state
            from mini_project_dms.Branch_Contact
natural join mini_project_dms.Branch natural join
mini_project_dms.Branch_State
            where b_id=%d order by b_phone DESC limit
1;
        '''
        try:
            cursor.execute(s % tup)
            res_tup = cursor.fetchall()
            res_lst = []
            for i in res_tup:
                for j in i:
                    res_lst.append(j)
            e1.delete(0, "end")
            e2.delete(0, "end")
            e3.delete(0, "end")
            e4.delete(0, "end")
            e5.delete(0, "end")
            e6.delete(0, "end")
            e1.insert(0, res_lst[0])
            e2.insert(0, res_lst[1])
            e3.insert(0, res_lst[2])
            e4.insert(0, res_lst[3])
            e5.insert(0, res_lst[4])
            e6.insert(0, res_lst[5])
            conn.commit()
        except:
```

Delivery Management System

```
        conn.rollback()
    elif (num == 3):
        try:
            b_id = int(e1.get())
            b_phone = int(e3.get())
        except:
            messagebox.showinfo('Alert', 'Branch id
should be a Number!')
            b_name = e2.get()
            street = e4.get()
            b_city = e5.get()
            state = e6.get()
            tup = (b_name, b_phone, street, b_city, state,
b_id)

            print(tup)
            s = '''
                update mini_project_dms.Branch_Contact bc
natural join mini_project_dms.Branch b natural join
mini_project_dms.Branch_State bs
                set
b.b_name=%s,bc.b_phone=%s,b.street=%s,bs.b_city=%s,bs.state
=%s where b_id=%s;
                '''
            try:
                cursor.execute(s, tup)
                conn.commit()
                messagebox.showinfo('Success', 'Updated
Successfully!')
                e1.delete(0, "end")
                e2.delete(0, "end")
                e3.delete(0, "end")
                e4.delete(0, "end")
                e5.delete(0, "end")
                e6.delete(0, "end")

            except:
                conn.rollback()
    elif (num == 4):
        e1.delete(0, "end")
        e2.delete(0, "end")
        e3.delete(0, "end")
        e4.delete(0, "end")
        e5.delete(0, "end")
        e6.delete(0, "end")
```

Delivery Management System

```
        elif (num == 5):
            root.destroy()
            DMS_Page()
            cursor.close()
            conn.close()

root = Tk()
f = Frame(root, width=620, height=500, bg="black")
root.title("Branch Details")
l1 = LabelFrame(f, text="Branch Details", width=400,
height=420, bg="black", fg="lightgreen",
font=('TIMES NEW ROMAN', 28,
'underline'))
l1.place(x=15, y=25)
labelframe2 = LabelFrame(f, text="", width=180,
height=400, bg="black", fg="lightgreen",
font=('TIMES NEW ROMAN', 28,
'underline'))
labelframe2.place(x=420, y=48)
l2 = Label(f, text="Branch ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l2.place(x=25, y=100)
l3 = Label(f, text="Name:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l3.place(x=25, y=160)
l4 = Label(f, text="Phone:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l4.place(x=25, y=220)
l5 = Label(f, text="Street:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l5.place(x=25, y=280)
l6 = Label(f, text="City:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l6.place(x=25, y=340)
l7 = Label(f, text="State:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l7.place(x=25, y=400)
e1 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e1.place(x=150, y=100)
e2 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e2.place(x=150, y=160)
e3 = Entry(f, width=20, fg="white", bg="black",
```

Delivery Management System

```
font=('TIMES NEW ROMAN', 13))
    e3.place(x=150, y=220)
    e4 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e4.place(x=150, y=280)
    e5 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e5.place(x=150, y=340)
    e6 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e6.place(x=150, y=400)
    b1 = Button(f, text="Add", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(1))
    b1.place(x=450, y=100)
    b2 = Button(f, text="Search", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(2))
    b2.place(x=450, y=170)
    b3 = Button(f, text="Update", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(3))
    b3.place(x=450, y=240)
    b4 = Button(f, text="Clear", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(4))
    b4.place(x=450, y=310)
    b5 = Button(f, text="Home Page", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(5))
    b5.place(x=450, y=380)
    f.pack()
    root.mainloop()

def onClickDelivery():
    def onClick3(num):
        conn = mysql.connector.connect(host='localhost',
user='root', password='legend_123',

database='mini_project_dms')
        cursor = conn.cursor()
        if (num == 1):
            try:
```

Delivery Management System

```
        del_id = int(e1.get())
        b_id = int(e4.get())
        del_ex_id = int(e5.get())
        c_id = int(e6.get())
        e_id = int(e7.get())
    except:
        messagebox.showinfo('Alert', 'Branch id and
Phone should be a Number!')
        del_date = e2.get()
        del_status = e3.get()
        tup1 = (del_id, del_date, del_status, b_id,
del_ex_id, c_id, e_id)

        s1 = "insert into
mini_project_dms.Order_Delivery (del_id, del_date,
del_status,b_id, del_ex_id, c_id, e_id)
values(%s,%s,%s,%s,%s,%s,%s) "
        try:
            cursor.execute(s1, tup1)
            conn.commit()
            messagebox.showinfo('Success', 'Inserted
Successfully!')
            e1.delete(0, "end")
            e2.delete(0, "end")
            e3.delete(0, "end")
            e4.delete(0, "end")
            e5.delete(0, "end")
            e6.delete(0, "end")
            e7.delete(0, "end")
        except:
            conn.rollback()
    elif (num == 2):
        del_id = int(e1.get())
        tup = (del_id)
        s = '''
        select del_id,del_date, del_status,b_id,
del_ex_id, c_id, e_id
        from mini_project_dms.Order_Delivery
        where del_id=%d;
        '''
        try:
            cursor.execute(s % tup)
            res_tup = cursor.fetchall()
            res_lst = []
```

Delivery Management System

```
        for i in res_tup:
            for j in i:
                res_lst.append(j)
            e1.delete(0, "end")
            e2.delete(0, "end")
            e3.delete(0, "end")
            e4.delete(0, "end")
            e5.delete(0, "end")
            e6.delete(0, "end")
            e7.delete(0, "end")
            e1.insert(0, res_lst[0])
            e2.insert(0, res_lst[1])
            e3.insert(0, res_lst[2])
            e4.insert(0, res_lst[3])
            e5.insert(0, res_lst[4])
            e6.insert(0, res_lst[5])
            e7.insert(0, res_lst[6])
            conn.commit()
    except:
        conn.rollback()

elif (num == 3):
    e1.delete(0, "end")
    e2.delete(0, "end")
    e3.delete(0, "end")
    e4.delete(0, "end")
    e5.delete(0, "end")
    e6.delete(0, "end")
    e7.delete(0, "end")
elif (num == 4):
    root.destroy()
    DMS_Page()
    cursor.close()
    conn.close()

root = Tk()
f = Frame(root, width=620, height=570, bg="black")
root.title("Customer Details")
l1 = LabelFrame(f, text="Customer Details", width=400,
height=520, bg="black", fg="lightgreen",
font=('TIMES NEW ROMAN', 28,
'underline'))
l1.place(x=15, y=25)
labelframe2 = LabelFrame(f, text="", width=180,
```

Delivery Management System

```
height=500, bg="black", fg="lightgreen",
                                font=('TIMES NEW ROMAN', 28,
'underline'))
    labelframe2.place(x=420, y=48)
    l2 = Label(f, text="Delivery ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
    l2.place(x=25, y=100)
    l3 = Label(f, text="Delivery Date:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
    l3.place(x=25, y=160)
    l4 = Label(f, text="Delivery Status:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
    l4.place(x=25, y=220)
    l5 = Label(f, text="Branch ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
    l5.place(x=25, y=280)
    l6 = Label(f, text="Executive ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
    l6.place(x=25, y=340)
    l7 = Label(f, text="Customer ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
    l7.place(x=25, y=400)
    l8 = Label(f, text="E-Commerce ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
    l8.place(x=25, y=460)
    e1 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e1.place(x=150, y=100)
    e2 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e2.place(x=150, y=160)
    e3 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e3.place(x=150, y=220)
    e4 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e4.place(x=150, y=280)
    e5 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e5.place(x=150, y=340)
    e6 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
    e6.place(x=150, y=400)
    e7 = Entry(f, width=20, fg="white", bg="black",
```


Delivery Management System

```
font=('TIMES NEW ROMAN', 13))
    e7.place(x=150, y=460)
    b1 = Button(f, text="Add", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(1))
    b1.place(x=450, y=120)
    b2 = Button(f, text="Search", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(2))
    b2.place(x=450, y=200)
    b3 = Button(f, text="Clear", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(3))
    b3.place(x=450, y=280)
    b4 = Button(f, text="Home Page", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
                command=lambda: onClick3(4))
    b4.place(x=450, y=360)
    f.pack()
    root.mainloop()

def onClickDel_Executive():
    def onClick3(num):
        conn = mysql.connector.connect(host='localhost',
user='root', password='legend_123',

database='mini_project_dms')
        cursor = conn.cursor()
        if (num == 1):
            try:
                del_ex_id = int(e1.get())
                del_ex_phone = int(e3.get())
            except:
                messagebox.showinfo('Alert', 'Branch id and
Phone should be a Number!')
            del_ex_name = e2.get()
            b_id = e4.get()
            tup2 = (del_ex_id, del_ex_name, b_id)
            tup3 = (del_ex_id, del_ex_phone)
            s2 = "insert into
mini_project_dms.delivery_executive(del_ex_id,del_ex_name,b
_id) values(%s,%s,%s)"
            s3 = "insert into
```

Delivery Management System

```
mini_project_dms.delivery_executive_contact(del_ex_id,del_e
x_phone) values(%s,%s)"
        try:
            cursor.execute(s2, tup2)
            conn.commit()
        except:
            conn.rollback()
        try:
            cursor.execute(s3, tup3)
            conn.commit()
            messagebox.showinfo('Success', 'Inserted
Successfully!')
            e1.delete(0, "end")
            e2.delete(0, "end")
            e3.delete(0, "end")
            e4.delete(0, "end")
        except:
            conn.rollback()

    elif (num == 2):
        del_ex_id = int(e1.get())
        tup = (del_ex_id)
        s = '''
            select
del_ex_id,del_ex_name,del_ex_phone,b_id
            from
mini_project_dms.delivery_executive_contact natural join
mini_project_dms.delivery_executive
            where del_ex_id=%d order by del_ex_phone
DESC limit 1;
        '''
        try:
            cursor.execute(s % tup)
            res_tup = cursor.fetchall()
            res_lst = []
            for i in res_tup:
                for j in i:
                    res_lst.append(j)
            e1.delete(0, "end")
            e2.delete(0, "end")
            e3.delete(0, "end")
            e4.delete(0, "end")
            e1.insert(0, res_lst[0])
            e2.insert(0, res_lst[1])
```

Delivery Management System

```
        e3.insert(0, res_lst[2])
        e4.insert(0, res_lst[3])
        conn.commit()
    except:
        conn.rollback()
elif (num == 3):
    try:
        del_ex_id = int(e1.get())
        del_ex_phone = int(e3.get())
    except:
        messagebox.showinfo('Alert', 'Branch id
should be a Number!')
        del_ex_name = e2.get()
        b_id = e4.get()
        tup = (del_ex_name, del_ex_phone, b_id,
del_ex_id)
        print(tup)
        s = '''
            update
mini_project_dms.delivery_executive_contact bc natural join
mini_project_dms.delivery_executive b
            set
b.del_ex_name=%s,bc.del_ex_phone=%s,b.b_id=%s where
del_ex_id=%s;
            '''
    try:
        cursor.execute(s, tup)
        conn.commit()
        messagebox.showinfo('Success', 'Updated
Successfully!')
        e1.delete(0, "end")
        e2.delete(0, "end")
        e3.delete(0, "end")
        e4.delete(0, "end")

    except:
        conn.rollback()
elif (num == 4):
    e1.delete(0, "end")
    e2.delete(0, "end")
    e3.delete(0, "end")
    e4.delete(0, "end")
elif (num == 5):
    root.destroy()
```

Delivery Management System

```
DMS_Page()
cursor.close()
conn.close()

root = Tk()
f = Frame(root, width=620, height=390, bg="black")
root.title("Executive Details")
l1 = LabelFrame(f, text="Executive Details", width=400,
height=350, bg="black", fg="lightgreen",
font=('TIMES NEW ROMAN', 28,
'underline'))
l1.place(x=15, y=25)
labelframe2 = LabelFrame(f, text="", width=180,
height=330, bg="black", fg="lightgreen",
font=('TIMES NEW ROMAN', 28,
'underline'))
labelframe2.place(x=420, y=48)
l2 = Label(f, text="Executive ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l2.place(x=25, y=100)
l3 = Label(f, text="Executive Name:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l3.place(x=25, y=160)
l4 = Label(f, text="Executive Phone:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l4.place(x=25, y=220)
l5 = Label(f, text="Branch ID:", fg="lightgreen",
bg="black", font=('TIMES NEW ROMAN', 13))
l5.place(x=25, y=280)
e1 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e1.place(x=150, y=100)
e2 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e2.place(x=150, y=160)
e3 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e3.place(x=150, y=220)
e4 = Entry(f, width=20, fg="white", bg="black",
font=('TIMES NEW ROMAN', 13))
e4.place(x=150, y=280)
b1 = Button(f, text="Add", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
command=lambda: onClick3(1))
```

Delivery Management System

```
b1.place(x=450, y=55)
b2 = Button(f, text="Search", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
            command=lambda: onClick3(2))
b2.place(x=450, y=120)
b3 = Button(f, text="Update", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
            command=lambda: onClick3(3))
b3.place(x=450, y=190)
b4 = Button(f, text="Clear", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
            command=lambda: onClick3(4))
b4.place(x=450, y=260)
b5 = Button(f, text="Home Page", width=8, height=1,
fg="white", bg="green", font=('TIMES NEW ROMAN', 16),
            command=lambda: onClick3(5))
b5.place(x=450, y=330)
f.pack()
root.mainloop()

class DMS_Page:
    def __init__(self):
        self.mainWindow = Tk()
        self.f2 = Frame(self.mainWindow, width=600,
height=530, bg="black")
        self.mainWindow.title("Delivery Management System")
        self.f2.pack()
        self.l1 = Label(self.f2, text="Delivery Management
System", fg="lightgreen", bg="black",
                        font=('TIMES NEW ROMAN', 28,
'underline'))
        self.l1.place(x=94, y=25)
        self.b1 = Button(self.f2, text="Customer Details",
width=15, height=1, fg="lightgreen", bg="black",
                        font=('TIMES NEW ROMAN', 25),
command=lambda: self.onClose(1))
        self.b1.place(x=140, y=120)
        self.b2 = Button(self.f2, text="Branch Details",
width=15, height=1, fg="lightgreen", bg="black",
                        font=('TIMES NEW ROMAN', 25),
command=lambda: self.onClose(2))
        self.b2.place(x=140, y=220)
        self.b3 = Button(self.f2, text="Delivery Details",
```

Delivery Management System

```
width=15, height=1, fg="lightgreen", bg="black",
                                font=('TIMES NEW ROMAN', 25),
command=lambda: self.onClose(3))
    self.b3.place(x=140, y=320)
    self.b4 = Button(self.f2, text="Executive Details",
width=15, height=1, fg="lightgreen", bg="black",
                                font=('TIMES NEW ROMAN', 25),
command=lambda: self.onClose(4))
    self.b4.place(x=140, y=420)
    self.mainWindow.mainloop()

def onClose(self, num):
    self.mainWindow.destroy()
    self.num = num
    if (self.num == 1):
        onClickCustomer()
    elif (self.num == 2):
        onClickBranch()
    elif (self.num == 3):
        onClickDelivery()
    elif (self.num == 4):
        onClickDel_Executive()

def onClick(num):
    conn = mysql.connector.connect(host='localhost',
user='root', password='legend_123',
database='mini_project_dms')
    if conn.is_connected():
        print('Connected to MySQL database')
        cursor = conn.cursor()
        branch_id = int(e1.get()) # b_id is login id
        password = e2.get()
        tuple1 = (branch_id, password)
        if (num == 1):
            s = "insert into
mini_project_dms.signup_page(b_id,password) values(%s,%s)"
            try:
                cursor.execute(s, tuple1)
                conn.commit()
                messagebox.showinfo('SignUp Successful', 'You
have succesfully signed up!')
            except:
                conn.rollback()
```

Delivery Management System

```
        messagebox.showinfo('Alert', 'Login id with
password already exist!')
    elif (num == 2):
        tuple1 = (branch_id)
        s = "select password from
mini_project_dms.signup_page where b_id=%d"
        try:
            cursor.execute(s % tuple1)
            password_fetched = cursor.fetchone()
            if password_fetched is not None:
                for i in password_fetched:
                    if (i == password):
                        conn.commit()
                        messagebox.showinfo('Login
Successful', 'You have succesfully logged in!')
                        root.destroy()
                        DMS_Page()

            else:
                messagebox.showinfo('Alert', 'Wrong
Password!')

        elif password_fetched is None:
            messagebox.showinfo('New User', 'Please
sign up first!')
        except:
            conn.rollback()
            cursor.close()
            conn.close()

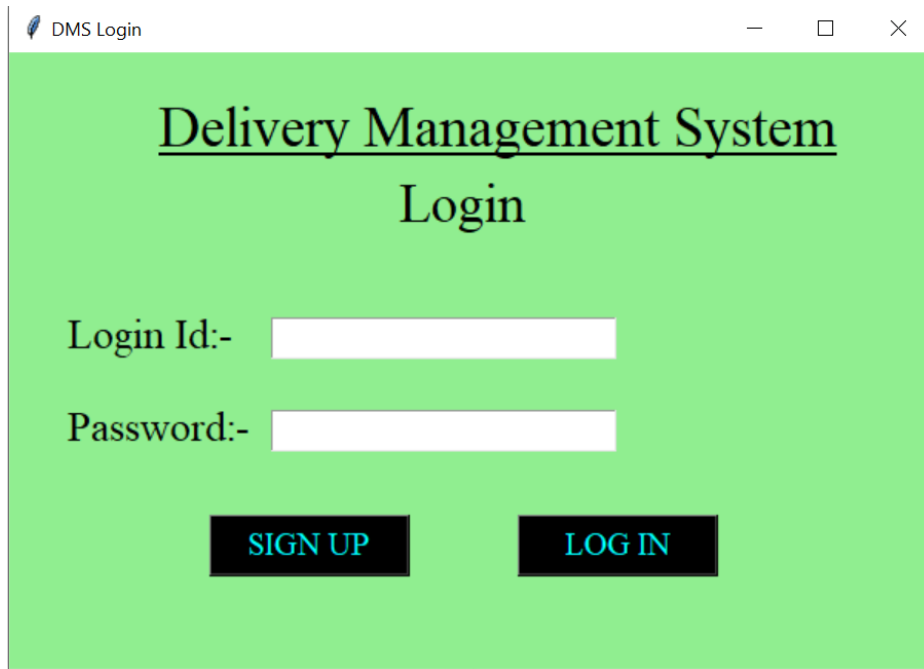
root = Tk()
f = Frame(root, width=600, height=400, bg="lightgreen")
root.title("DMS Login")
l1 = Label(f, text="Delivery Management System",
fg="black", bg="lightgreen", font=('TIMES NEW ROMAN', 28,
'underline'))
l1.place(x=94, y=25)
l2 = Label(f, text="Login", fg="black", bg="lightgreen",
font=('TIMES NEW ROMAN', 26))
l2.place(x=250, y=75)
l3 = Label(f, text="Login Id:-", fg="black",
bg="lightgreen", font=('TIMES NEW ROMAN', 20))
l3.place(x=35, y=165)
l4 = Label(f, text="Password:-", fg="black",
```

Delivery Management System

```
bg="lightgreen", font=('TIMES NEW ROMAN', 20))
l4.place(x=35, y=225)
e1 = Entry(f, width=20, bg="white", fg="black",
font=('TIMES NEW ROMAN', 16))
e1.place(x=170, y=172)
e2 = Entry(f, width=20, bg="white", fg="black",
font=('TIMES NEW ROMAN', 16), show='*')
e2.place(x=170, y=232)
b1 = Button(f, text="SIGN UP", width=10, height=1,
fg="cyan", bg="black", font=('TIMES NEW ROMAN', 16),
command=lambda: onClick(1))
b1.place(x=130, y=300)
b2 = Button(f, text="LOG IN", width=10, height=1,
fg="cyan", bg="black", font=('TIMES NEW ROMAN', 16),
command=lambda: onClick(2))
b2.place(x=330, y=300)
f.pack()
root.mainloop()
```

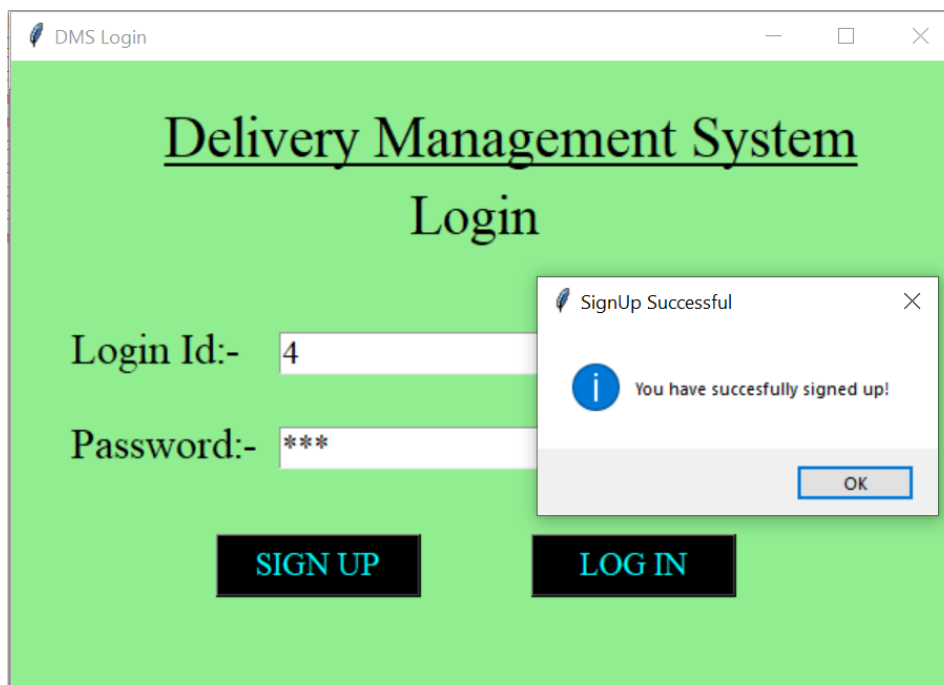

GUI FRAMES

1) Login page to DMS:



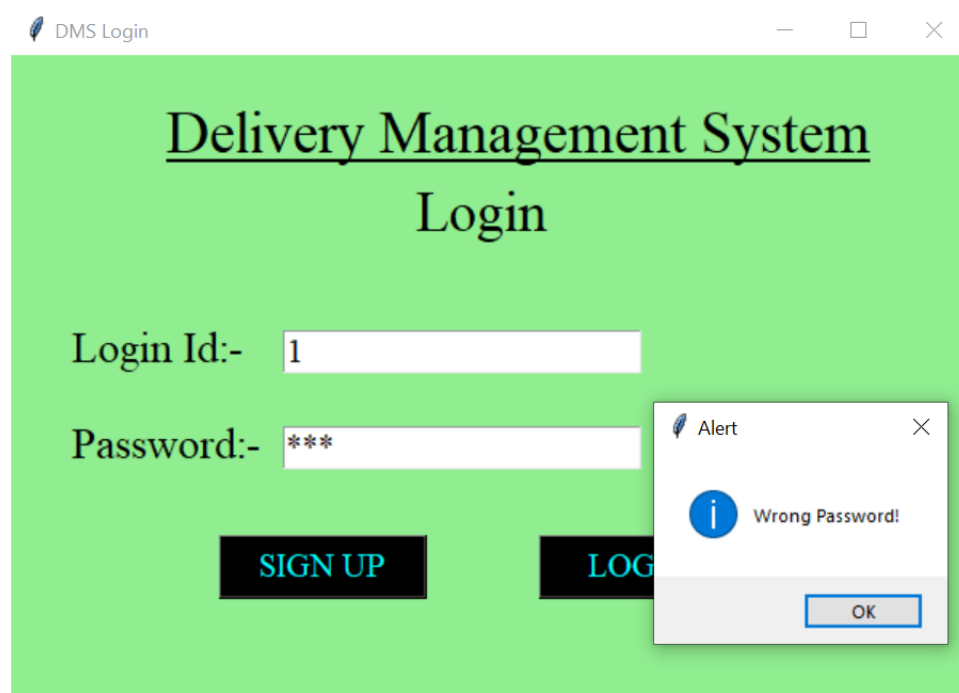
A screenshot of a web application window titled "DMS Login". The background is light green. At the top center, the text "Delivery Management System" is underlined in black, with "Login" centered below it. There are two input fields: "Login Id:-" followed by a white text box, and "Password:-" followed by a white text box. Below these fields are two black buttons with white text: "SIGN UP" on the left and "LOG IN" on the right.

2) Sign up page:

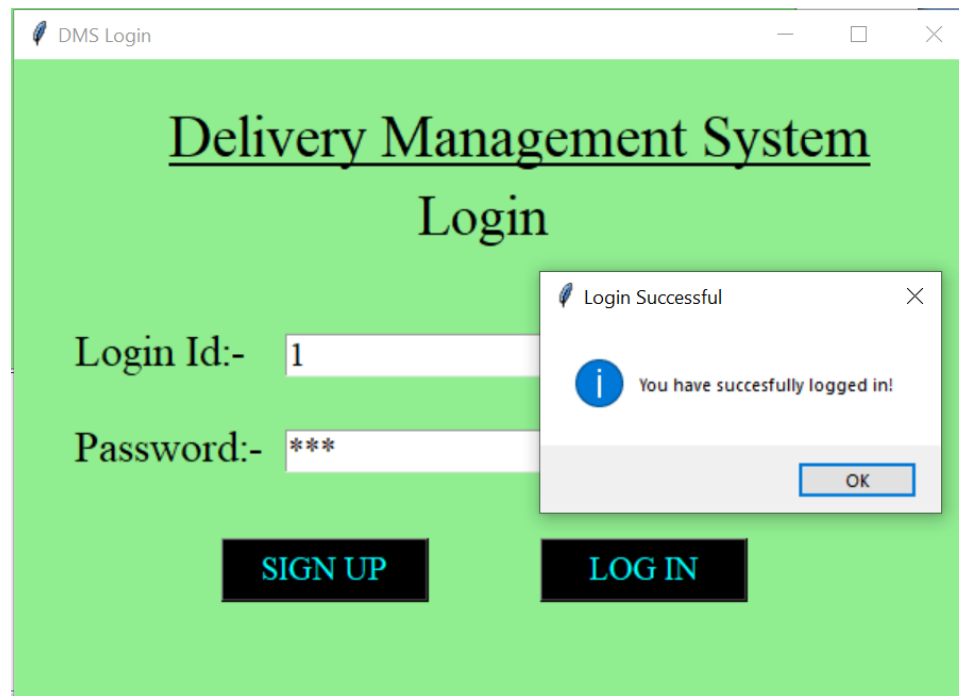


A screenshot of the same "DMS Login" window. The "Login Id:-" field now contains the number "4" and the "Password:-" field contains four asterisks "****". A small white dialog box with a blue information icon and the text "You have succesfully signed up!" is overlaid on the right side of the form. The dialog box has an "OK" button at the bottom right. The "SIGN UP" and "LOG IN" buttons remain at the bottom.

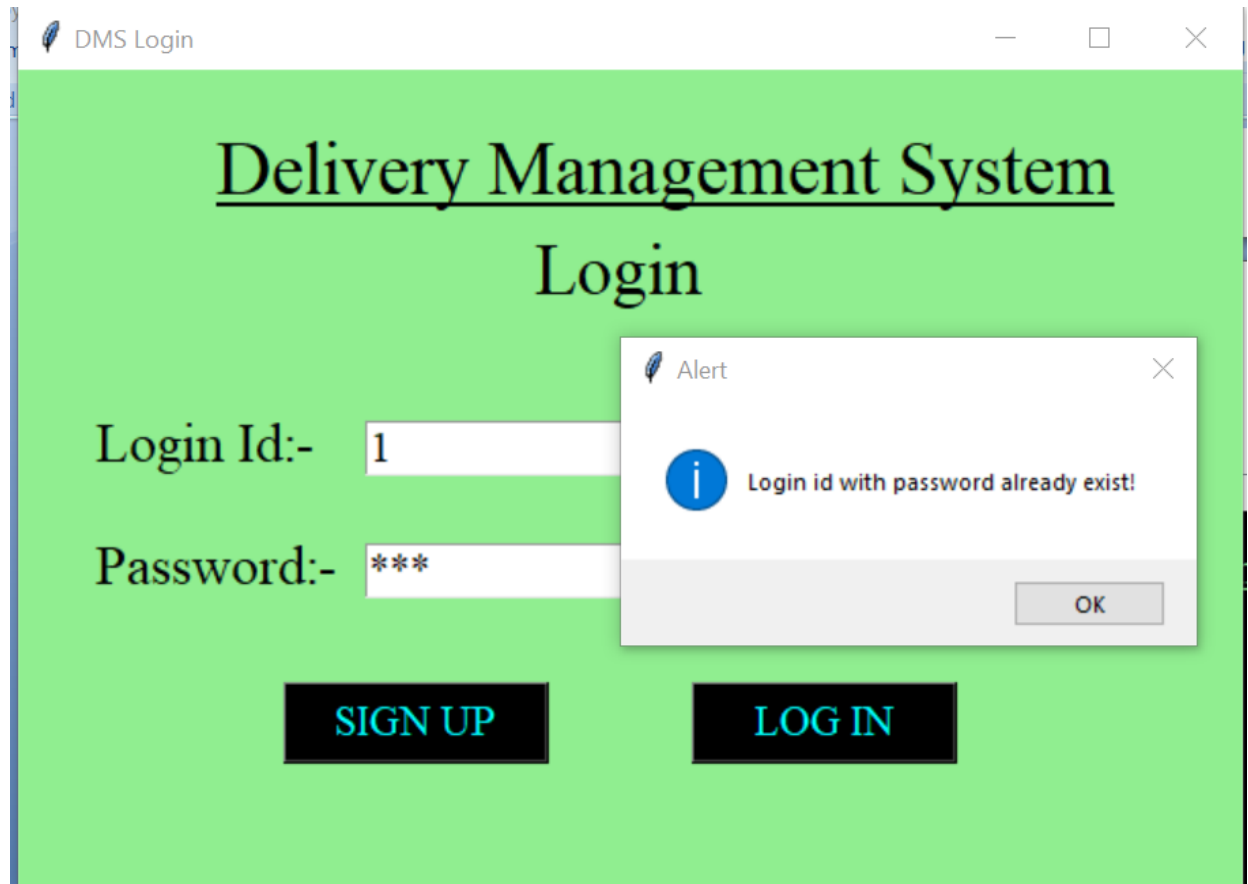
3) Logging in with wrong password:



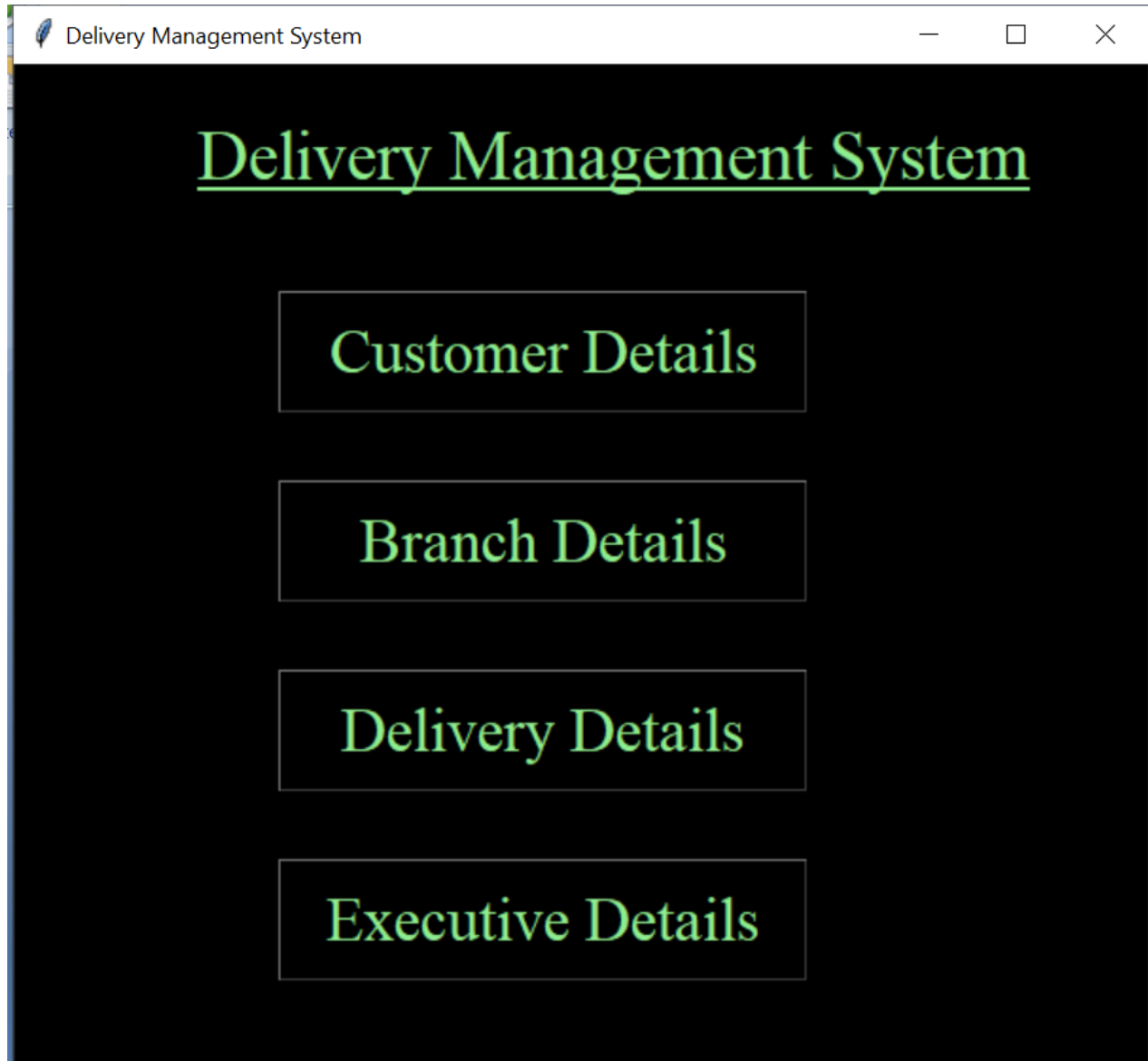
4) Logged in successfully:



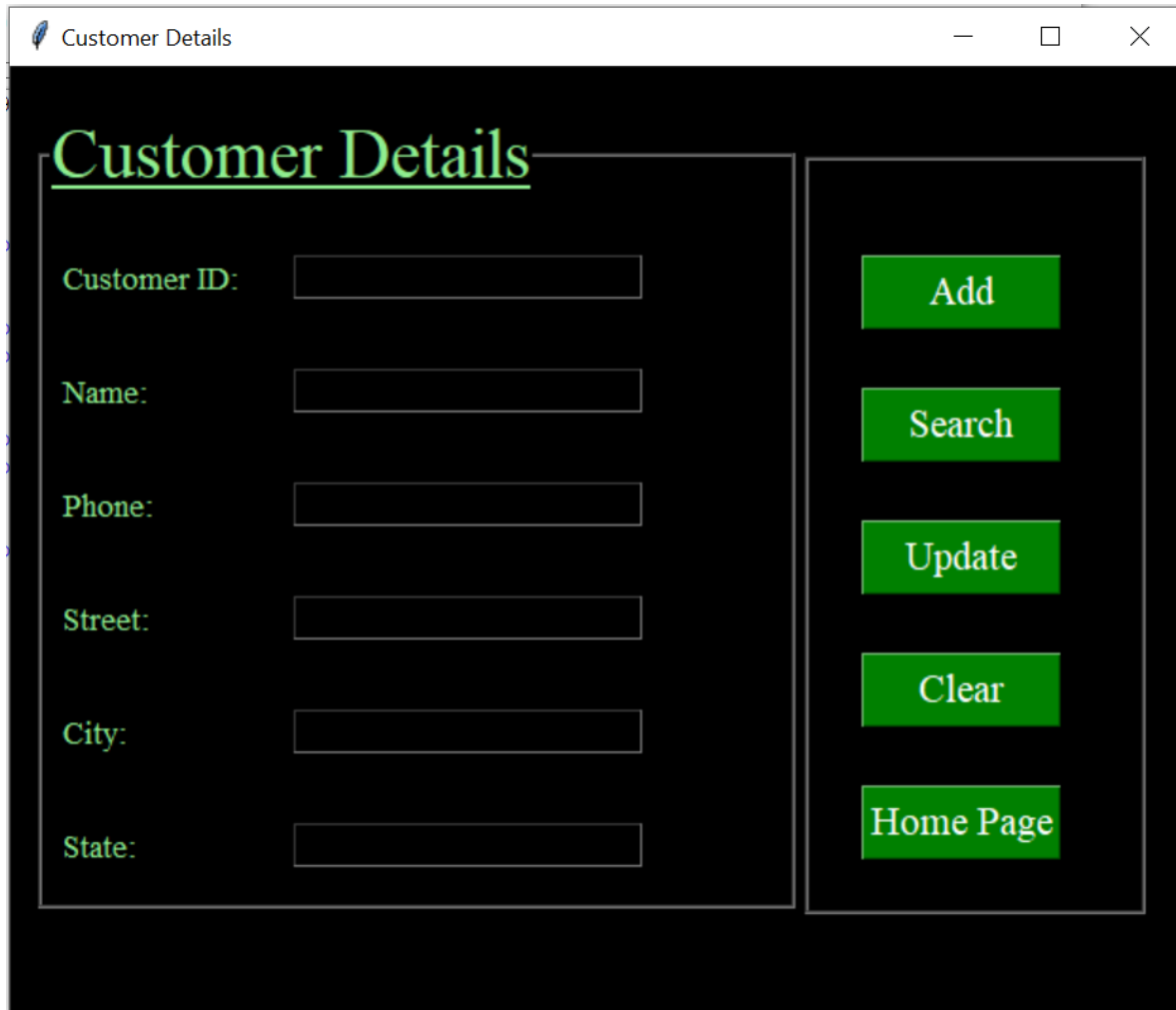
5) Signing up with same login id:



6) DMS Homepage:



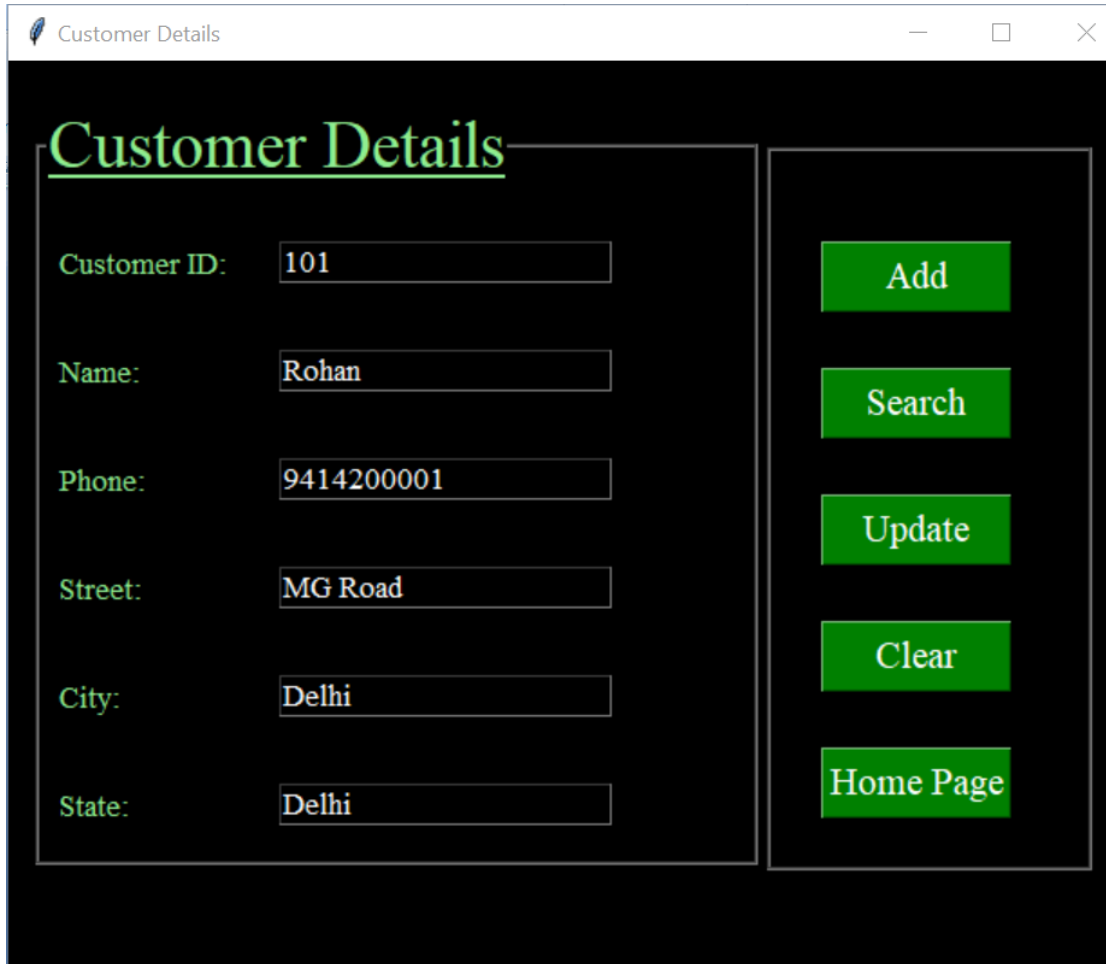
7) Customer Details:



The screenshot shows a web application window titled "Customer Details". The interface has a dark background. On the left, there is a form with the title "Customer Details" in a large, stylized font. Below the title, there are six input fields labeled "Customer ID:", "Name:", "Phone:", "Street:", "City:", and "State:". To the right of the form, there is a vertical column of five green buttons: "Add", "Search", "Update", "Clear", and "Home Page".

Add	To add new customer entries to the database.
Search	To search customer details using customer id.
Update	To update the customer data using customer id
Clear	To clear the values from the entry widget
Home Page	To return to the DMS homepage

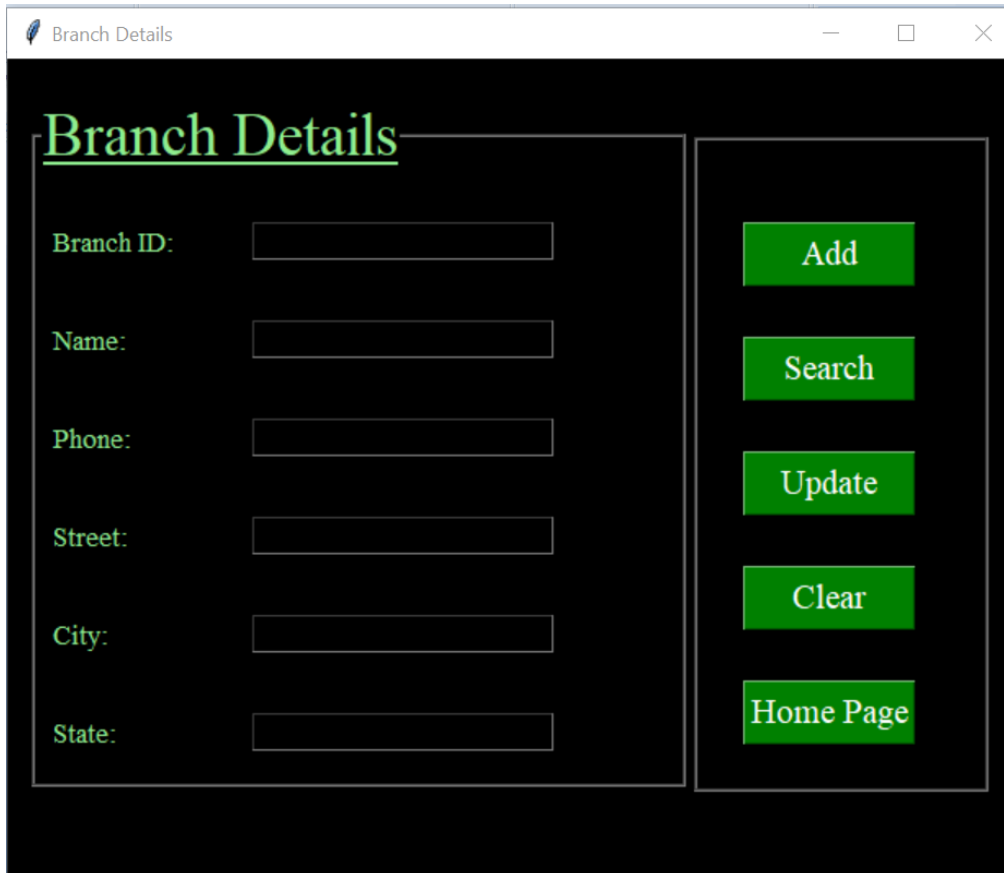
Ex- Searching the details of a customer with customer id = 101

A screenshot of a web application window titled "Customer Details". The window has a dark blue header bar with the title and standard window controls (minimize, maximize, close). The main content area is white and contains a form with several input fields and a sidebar of action buttons. The form fields are labeled "Customer ID:", "Name:", "Phone:", "Street:", "City:", and "State:". The input values are "101", "Rohan", "9414200001", "MG Road", "Delhi", and "Delhi" respectively. The sidebar on the right contains five blue buttons: "Add", "Search", "Update", "Clear", and "Home Page".

Customer Details	
Customer ID:	101
Name:	Rohan
Phone:	9414200001
Street:	MG Road
City:	Delhi
State:	Delhi

[Add](#)
[Search](#)
[Update](#)
[Clear](#)
[Home Page](#)

8) Branch Details:



The screenshot shows a web browser window titled "Branch Details". The interface has a dark blue background. On the left, there is a form titled "Branch Details" with the following fields: "Branch ID:", "Name:", "Phone:", "Street:", "City:", and "State:". Each field has a corresponding text input box. On the right side of the form, there is a vertical stack of five green buttons: "Add", "Search", "Update", "Clear", and "Home Page".

Add	To add new branch entries to the database.
Search	To search branch details using branch id.
Update	To update the branch data using branch id
Clear	To clear the values from the entry widget
Home Page	To return to the DMS homepage

Ex- Branch details of branch with branch id= 2

Branch Details

Branch ID:

2

Name:

ABC

Phone:

9414100003

Street:

MG Road

City:

Delhi

State:

Delhi

Add


Search

Update

Clear

Home Page

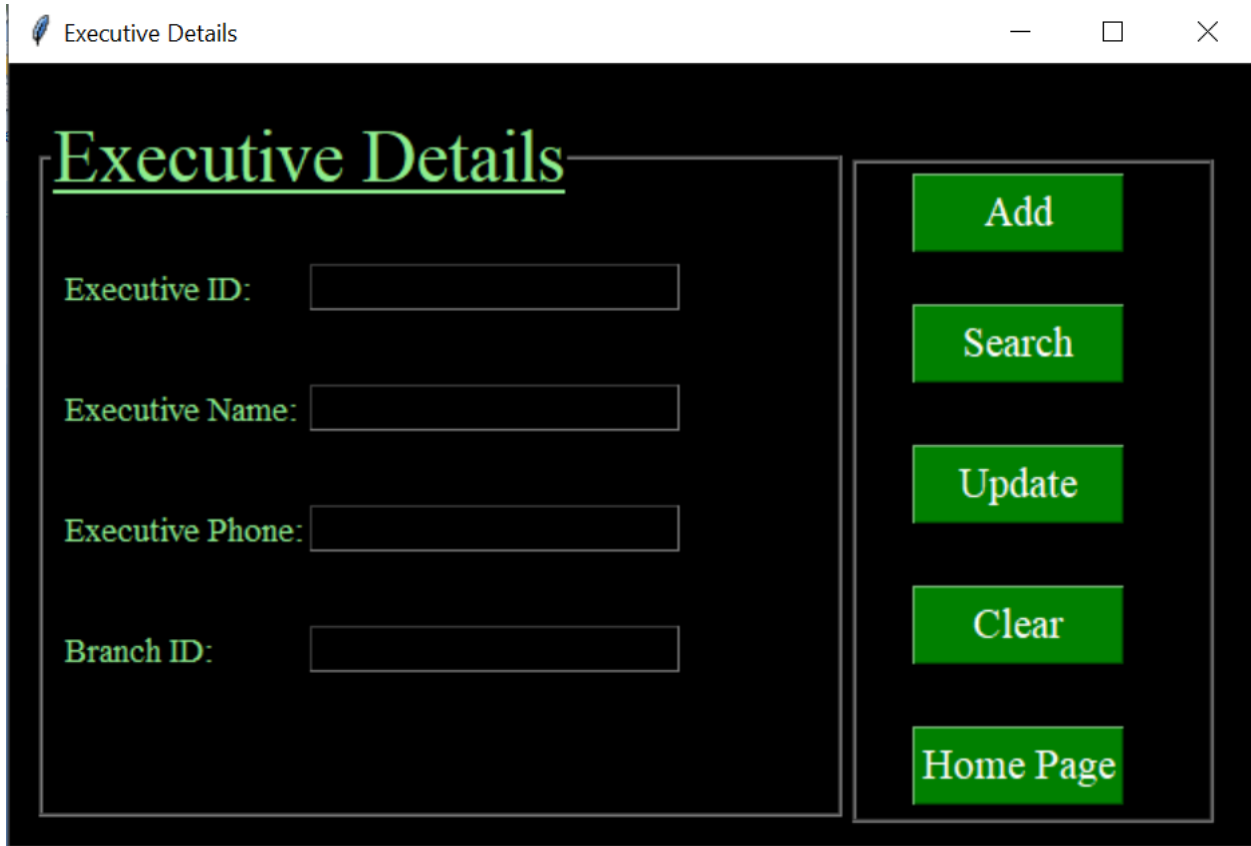
Ex- Delivery details of an order with delivery d = 501

 Customer Details—□×

Customer Details

Delivery ID:	<input type="text" value="501"/>	<div>Add</div> <div>Search</div> <div>Clear</div> <div>Home Page</div>
Delivery Date:	<input type="text" value="2017-06-15"/>	
Delivery Status:	<input type="text" value="Delivered"/>	
Branch ID:	<input type="text" value="1"/>	
Executive ID:	<input type="text" value="201"/>	
Customer ID:	<input type="text" value="104"/>	
E-Commerce ID:	<input type="text" value="11"/>	

10) Delivery Executive details:



Executive Details

Executive ID:

Executive Name:

Executive Phone:

Branch ID:

Add

Search

Update

Clear

Home Page

Add	To add new delivery executive entries to the database.
Search	To search customer details using delivery executive id.
Update	To update the delivery executive data
Clear	To clear the values from the entry widget
Home Page	To return to the DMS homepage

Ex- Delivery Executive details where del_ex_id = 202

Executive Details

Executive Details

Executive ID: 202

Executive Name: Kunj

Executive Phone: 9414300001

Branch ID: 1

Add

Search

Update

Clear

Home Page

Ex- New customer added

Customer Details

Customer ID:

107

Name:

Arihant

Phone:

9414200008

Street:

Ramlal Jat

City:

Chennai

State:

Tamil Nadu

Add

Search

Home Page

Success

i

Inserted Successfully!

OK


Ex- Street UPDATED for customer id 101.

The screenshot shows a web application window titled "Customer Details". The window has a dark blue header bar with the title and standard window controls (minimize, maximize, close). The main content area has a light blue background. On the left, there is a form with the title "Customer Details" in a large, bold, blue font. The form contains six input fields, each with a label and a value:

- Customer ID: 101
- Name: Rohan
- Phone: 9414200001
- Street: Cannaut Place (highlighted in red)
- City: Delhi
- State: Delhi

On the right side of the form, there are three blue buttons: "Add", "Search", and "Update". A "Success" dialog box is open in the foreground, displaying a green information icon, the text "Updated Successfully!", and an "OK" button.

APPLICATION CREATED (exe file)



lib	4/12/2021 2:23 AM	File folder	
api-ms-win-crt-convert-l1-1-0.dll	7/10/2015 2:02 AM	Application extens...	23 KB
api-ms-win-crt-heap-l1-1-0.dll	7/10/2015 2:02 AM	Application extens...	20 KB
api-ms-win-crt-locale-l1-1-0.dll	7/10/2015 2:02 AM	Application extens...	19 KB
api-ms-win-crt-math-l1-1-0.dll	7/10/2015 2:02 AM	Application extens...	29 KB
api-ms-win-crt-runtime-l1-1-0.dll	7/10/2015 2:02 AM	Application extens...	23 KB
api-ms-win-crt-stdio-l1-1-0.dll	7/10/2015 2:02 AM	Application extens...	25 KB
api-ms-win-crt-string-l1-1-0.dll	7/10/2015 2:02 AM	Application extens...	25 KB
api-ms-win-crt-utility-l1-1-0.dll	7/10/2015 2:02 AM	Application extens...	19 KB
<u>Delivery Management System</u>	4/12/2021 2:19 AM	Application	15 KB
libzmq.cp38-win32.pyd	9/25/2020 8:50 PM	PYD File	433 KB
msvcp140.dll	9/25/2020 8:50 PM	Application extens...	430 KB
python3.dll	7/20/2020 4:03 PM	Application extens...	58 KB
python38.dll	7/20/2020 4:03 PM	Application extens...	3,957 KB
vcruntime140.dll	6/23/2020 7:39 PM	Application extens...	79 KB