FOOD CHAIN MANAGEMENT SYSTEM

Date: 23rd October 2019

Name and Roll no:
Rishab Saini 100
Kenil Shah 105

➢ DESCRIPTION:

First, the project starts with the home page, the page where the user can select his/her occupation as Farmer, Distributor or Retailer. Then the next step is to select whether the user is a new user or registered user. If new is selected, the signup form opens us and the entry is added to the database. If registered user is selected, the login page opens up and the user has to enter two fields: registered unique ID and password to log in. Once user is logged in, he/she has three options: View Data, Update Data and Logout. View Data prints the database, updates the data in the respective tables.

There is a separate application the "ADMIN PANEL" which deals with deleting the data.

➢ TECHNOLOGY STACK:

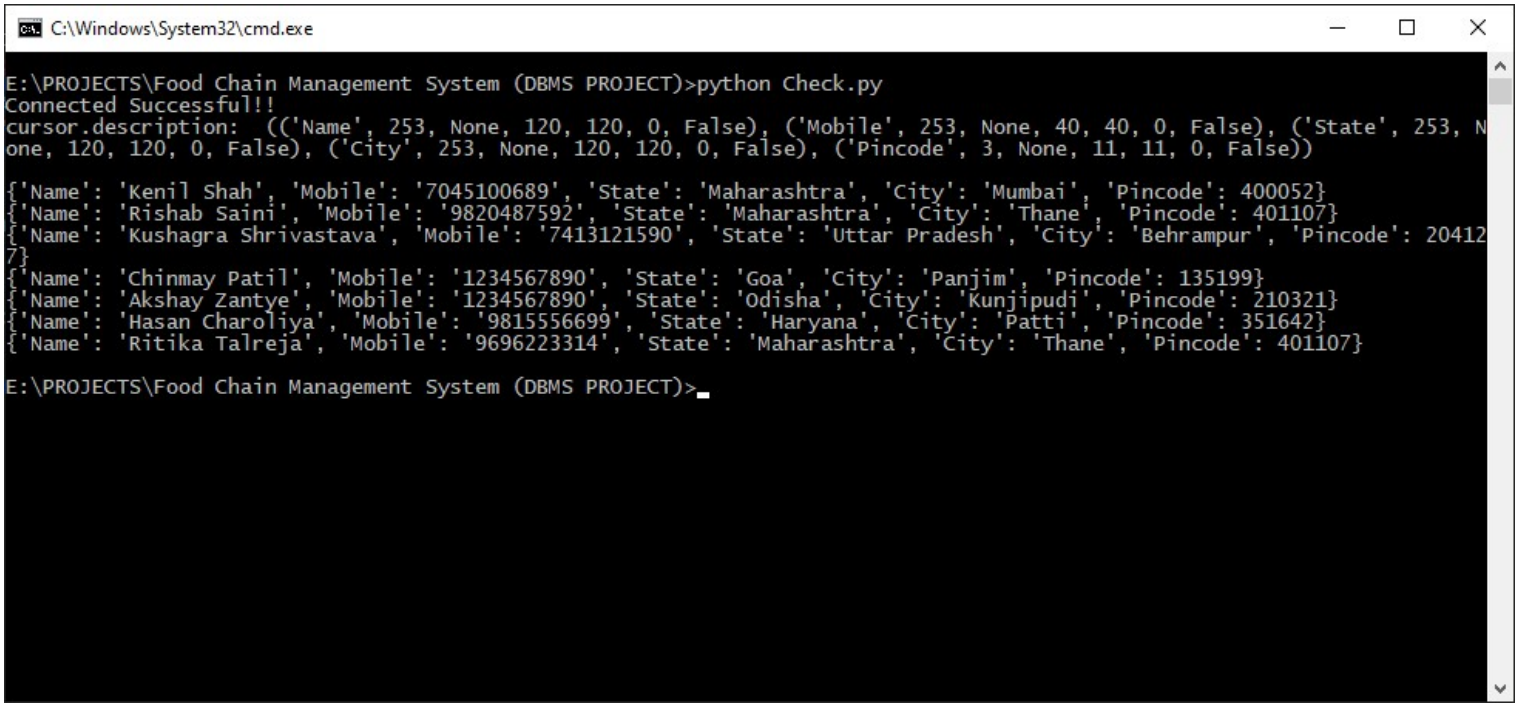
- Python 3.X
- tKinter
- XAMPP

➢ CODE & SCREENSHOTS:

• Database Connectivity Confirmation:

```
import pymysql.cursors
connection = pymysql.connect(host='localhost',user='root',password='WeRock',db='foodchainmanagement',
charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)
```

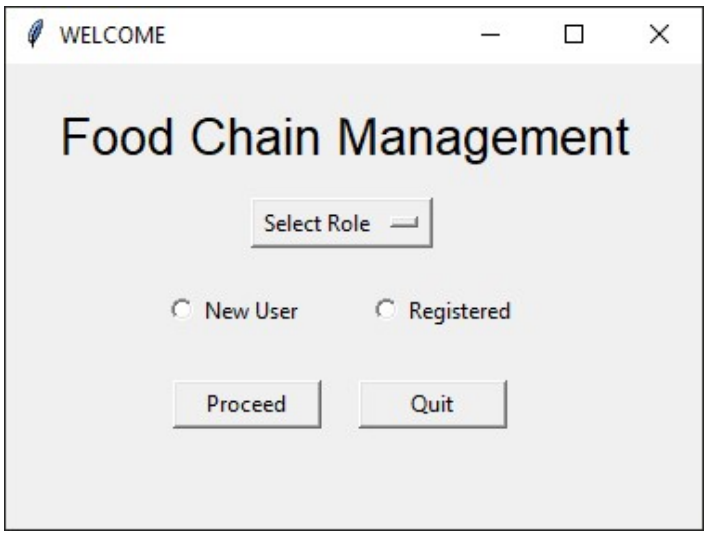
```
print ("Connected Successful!!")

try:
   with connection.cursor() as cursor:
      sql = "SELECT Name,Mobile,State,City,Pincode FROM Farmer"
      cursor.execute(sql)
      print ("cursor.description: ", cursor.description)
      print()
      for row in cursor:
         print(row)

finally:
   connection.close()
```



- **<u>Home Page:</u>**

```
from tkinter import *
import Login as L

def NewWindow():
    if variable.get()=='Farmer':
            if(int(var.get())==1):
                    quit_f()
                    import FarmerSignUp
            elif(int(var.get())==2):
```

```
                    quit_f()
                    L.LoginForm()

    elif variable.get()=='Distributor':
            master.destroy()
            root2=Tk()
            root2.mainloop()

    elif variable.get()=='Retailer':
            master.quit()
            root3=Tk()
            root3.mainloop()

def         quit_f():
    master.destroy()

master=Tk()
master.geometry('375x250')
master.title("WELCOME")

Label(master,                                   text="Food                                   Chain
Management",width=20,font=("bold",20)).place(x=20,y=20)

variable=StringVar(master)
variable.set("Select Role")
option=OptionMenu(master, variable, "Farmer", "Distributor", "Retailer")
option.place(x=130,y=70)

var=IntVar()
Radiobutton(master, text="New User",padx = 5, variable=var, value=1).place(x=80,y=120)
Radiobutton(master,          text="Registered",padx          =          20,          variable=var,
value=2).place(x=175,y=120)

Button(master,text='Proceed',width=10,command=NewWindow).place(x=90,y=170)
Button(master,text='Quit',width=10,command=quit_f).place(x=190,y=170)

master.mainloop()
```



- **Sign Up Page:**

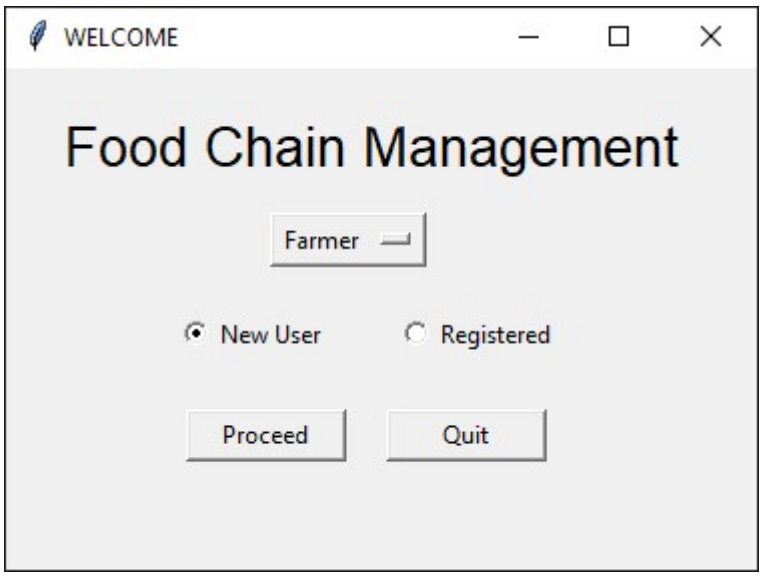
```
from tkinter import *
import pymysql.cursors
import random
import Login as L
import PostLogin as P
import Update as U

root = Tk()
root.geometry('500x750')
root.title("Registration form for FARMER")

#GENERATING RANDOM UNID
def create_unid():
    return(random.randint(1001, 1999))

#VARIABLE INITIALIZATIONS
unid= create_unid()
name= StringVar()
mobile= StringVar()
crop_name= StringVar()
soil_type= StringVar()
weight_per_yeild= IntVar()
cost_per_quintal= IntVar()
aadhar= StringVar()
city= StringVar()
nstate= StringVar()
pincode= IntVar()
```

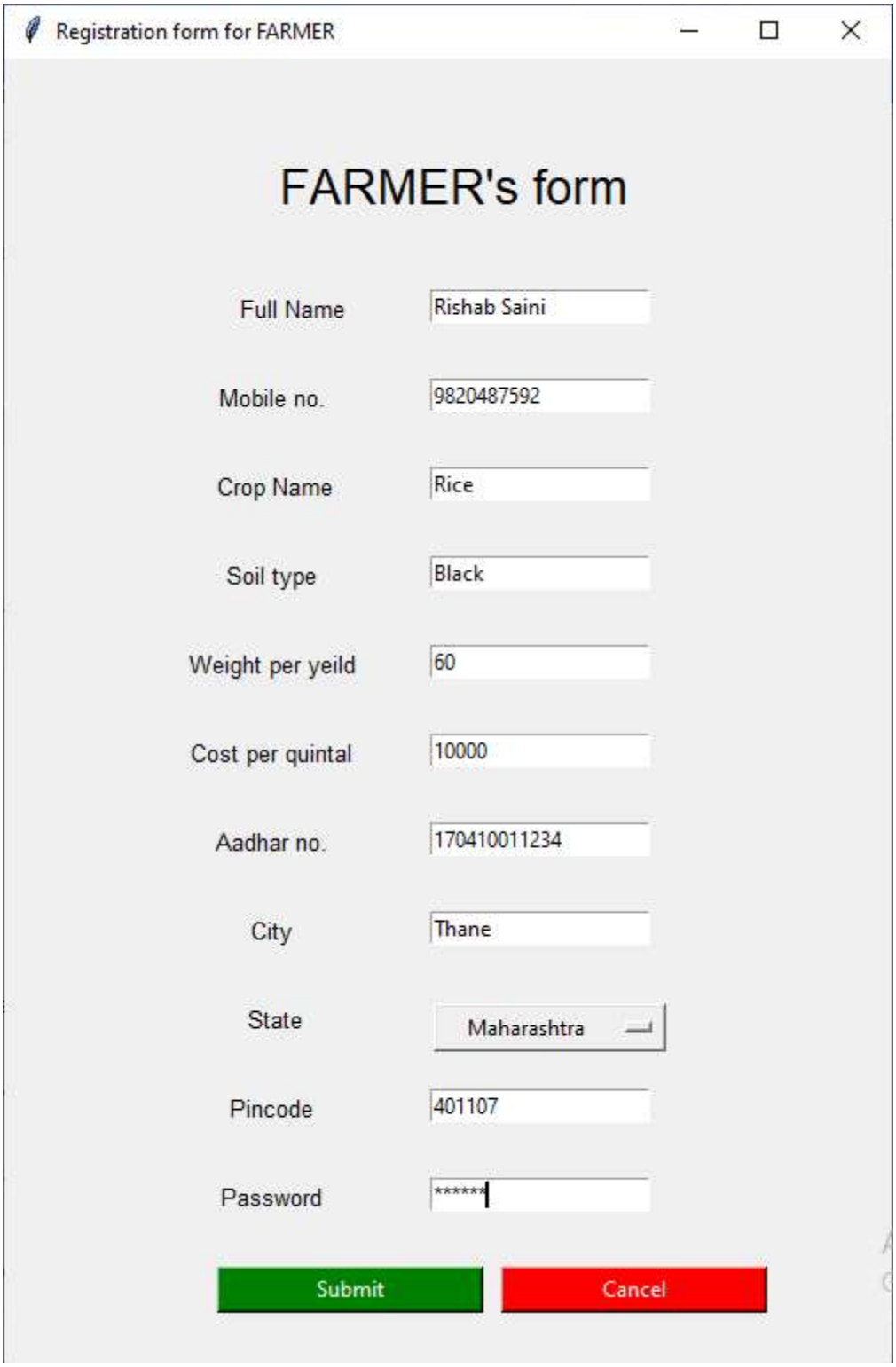
```
password= StringVar()
c=StringVar()

def Back():
   PSU.destroy()
   root.deiconify()

#DATABASE CONNECTIVITY
def database_conn():

    conn=
pymysql.connect(host='localhost',user='root',password='WeRock',db='foodchainmanagem
ent',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)

    print ("Connected successfully!")

    if(conn):

            unid=int(create_unid())
            name=str(entry_1.get())
            mobile=str(entry_2.get())
            crop_name=str(entry_3.get())
            soil_type=str(entry_4.get())
            weight_per_yeild=int(entry_5.get())
            cost_per_quintal=int(entry_6.get())
            aadhar=str(entry_7.get())
            city=str(entry_8.get())
            state=str(c.get())
            pincode=int(entry_10.get())
            password=str(entry_11.get())

    print(unid,name,mobile,crop_name,soil_type,weight_per_yeild,cost_per_quintal,aadha
r,city,nstate,pincode,password)

            try:
                    with conn.cursor() as cursor:
                            insert1 = "INSERT INTO Farmer
(Unid,Name,Mobile,Crop_name,Soil_type,Weight_per_yeild,Cost_per_quintal,Aadhar,City,
State,Pincode,Password)  VALUES(%d,'%s','%s','%s','%s',%d,%d,'%s','%s','%s',%d,'%s')"  %
(unid,name,mobile,crop_name,soil_type,weight_per_yeild,cost_per_quintal,aadhar,city,st
ate,pincode,password)
                            cursor.execute(insert1)
                            conn.commit()
                            print("Values Inserted Successfully!")
```

```
            finally:
                conn.close()

    else:
            print("Connection Unsuccesfull.")

    global PSU
    PSU = Toplevel()
    PSU.title("Successfully logged in!")
    width = 400
    height = 200
    screen_width = root.winfo_screenwidth()
    screen_height = root.winfo_screenheight()
    x = (screen_width/2) - (width/2)
    y = (screen_height/2) - (height/2)
    root.resizable(0, 0)
    PSU.geometry("%dx%d+%d+%d" % (width, height, x, y))

    lbl_home = Label(PSU, text="Welcome! %s" % str(unid), font=('times new roman',
20)).pack()

    btn_1= Button(PSU, text='Update', width=20,
command=U.UpdateValues).place(x=100,y=100)

    btn_cont = Button(PSU, text='View All Data', width=20,
command=P.FarmerView).place(x=20, y= 150)
    btn_back = Button(PSU, text='Logout', width=20, command=Back).place(x=200, y=150)

def quit_func():
    root.withdraw()
    import HomePage


#GUI CODE
label_0 = Label(root, text="FARMER's form",width=20,font=("bold", 20))
label_0.place(x=90,y=53)


label_1 = Label(root, text="Full Name",width=20,font=("bold", 10))
label_1.place(x=80,y=130)

entry_1 = Entry(root,textvar=name)
entry_1.place(x=240,y=130)
```

```
label_2 = Label(root, text="Mobile no.",width=20,font=("bold", 10))
label_2.place(x=68,y=180)

entry_2 = Entry(root,textvar=mobile)
entry_2.place(x=240,y=180)

label_3 = Label(root, text="Crop Name",width=20,font=("bold", 10))
label_3.place(x=70,y=230)

entry_3 = Entry(root,textvar=crop_name)
entry_3.place(x=240,y=230)

label_4 = Label(root, text="Soil type",width=20,font=("bold", 10))
label_4.place(x=68,y=280)

entry_4 = Entry(root,textvar=soil_type)
entry_4.place(x=240,y=280)

label_5 = Label(root, text="Weight per yeild",width=20,font=("bold", 10))
label_5.place(x=68,y=330)

entry_5 = Entry(root,textvar=weight_per_yeild)
entry_5.place(x=240,y=330)

label_6 = Label(root, text="Cost per quintal",width=20,font=("bold", 10))
label_6.place(x=68,y=380)

entry_6 = Entry(root,textvar=cost_per_quintal)
entry_6.place(x=240,y=380)

label_7 = Label(root, text="Aadhar no.",width=20,font=("bold", 10))
label_7.place(x=68,y=430)

entry_7 = Entry(root,textvar=aadhar)
entry_7.place(x=240,y=430)

label_8 = Label(root, text="City",width=20,font=("bold", 10))
label_8.place(x=68,y=480)

entry_8 = Entry(root,textvar=city)
entry_8.place(x=240,y=480)

label_9 = Label(root, text="State",width=20,font=("bold", 10))
```

```
label_9.place(x=70,y=530)

list1 = ['Andhra Pradesh','Arunachal Pradesh','Assam','Bihar','Chhattisgarh','Goa', 'Gujrat', 'Haryana', 'Himachal Pradesh', 'Jammu & Kashmir', 'Jharkhand', 'Karnataka', 'Kerala', 'Madhya Pradesh', 'Maharashtra', 'Manipur', 'Meghalaya', 'Mizoram', 'Nagaland', 'Odisha', 'Punjab', 'Rajasthan', 'Sikkim', 'Tamil Nadu', 'Tamil Nadu', 'Tripura', 'Uttar Pradesh', 'Uttarakhand', 'West Bengal' ];

droplist=OptionMenu(root,c, *list1)
droplist.config(width=15)
c.set('Select your State')
droplist.place(x=240,y=530)

label_10 = Label(root, text="Pincode",width=20,font=("bold", 10))
label_10.place(x=68,y=580)

entry_10 = Entry(root,textvar=pincode)
entry_10.place(x=240,y=580)

label_11 = Label(root, text="Password",width=20,font=("bold", 10))
label_11.place(x=68,y=630)

entry_11 = Entry(root,show="*",textvar=password)
entry_11.place(x=240,y=630)

Button(root, text='Submit',width=20,bg='green',fg='white', command=database_conn).place(x=120,y=680)
Button(root, text='Cancel',width=20,bg='red',fg='white', command=quit_func).place(x=280,y=680)

root.mainloop()
```


WELCOME
Food Chain Management
Farmer
New User
Registered
Proceed
Quit



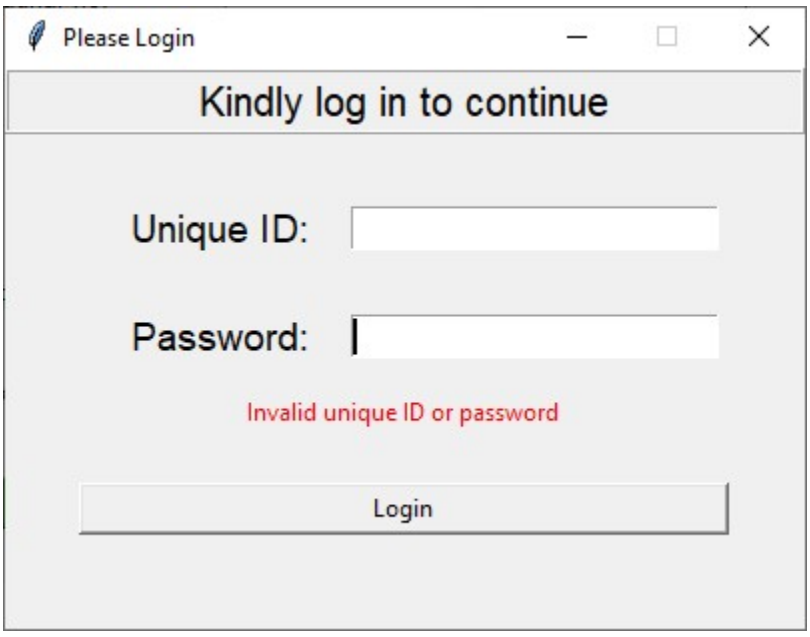
- **<u>Log In Page:</u>**

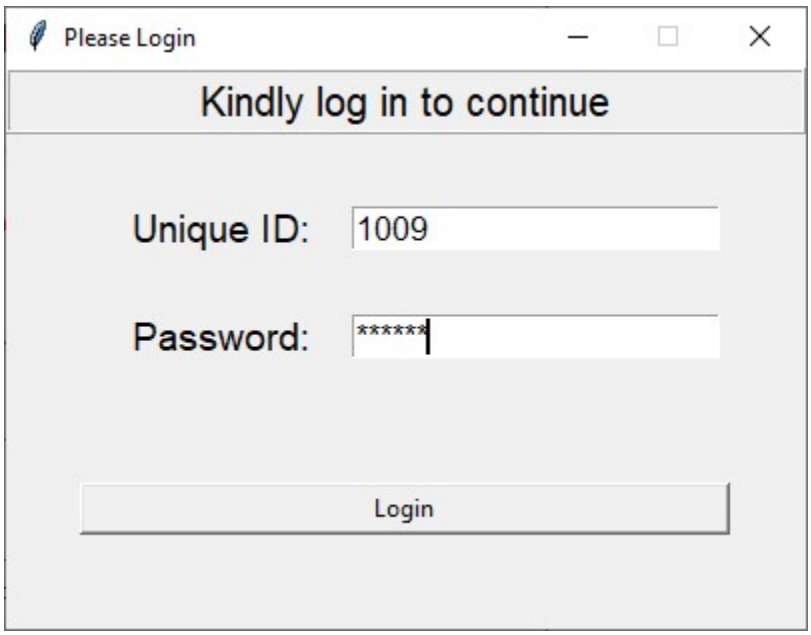
```
from tkinter import *
```

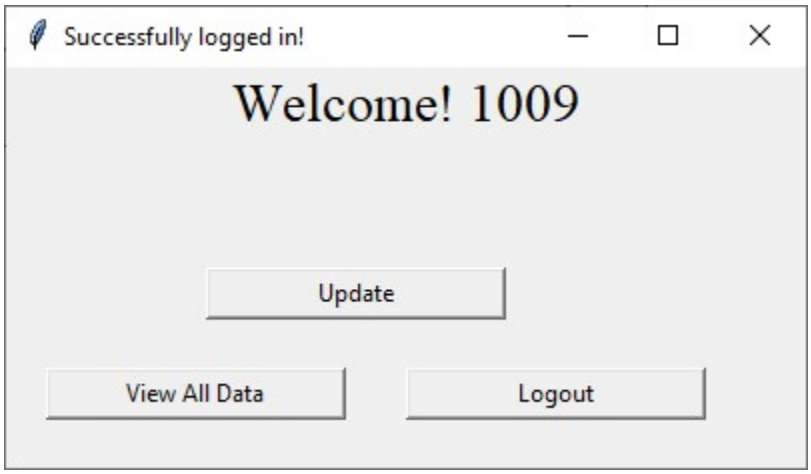
```
import pymysql.cursors
import PostLogin
import Update as U

def LoginForm():
    def Database():
        conn=
pymysql.connect(host='localhost',user='root',password='WeRock',db='foodchainmanagem
ent',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)
        cursor = conn.cursor()
        return conn,cursor

    def Login(event=None):

            conn,cursor=Database()
            """
            if USERNAME.get() == "" or PASSWORD.get() == "":
                    lbl_text.config(text="Please complete the required field!", fg="red")
            else:
            """
            un=int(username.get())
            passwrd=str(password.get())
            sql = "SELECT Name FROM FARMER WHERE Unid= %d AND Password= '%s'" %
(un,passwrd)
            print("\n\n",un,"\n",passwrd,"\n",sql,"\n\n")
            cursor.execute(sql)
            if cursor.fetchone() is not None:
                    HomeWindow()
                    USERNAME.set("")
                    PASSWORD.set("")
                    lbl_text.config(text="")
            else:
                    lbl_text.config(text="Invalid unique ID or password", fg="red")
                    USERNAME.set("")
                    PASSWORD.set("")

            cursor.close()
            conn.close()

    def Continue():
            Home.withdraw()
            PostLogin.FarmerView()

    def HomeWindow():
```

```
        global Home
        root.withdraw()
        Home = Toplevel()
        Home.title("Successfully logged in!")
        width = 400
        height = 200
        screen_width = root.winfo_screenwidth()
        screen_height = root.winfo_screenheight()
        x = (screen_width/2) - (width/2)
        y = (screen_height/2) - (height/2)
        root.resizable(0, 0)
        Home.geometry("%dx%d+%d+%d" % (width, height, x, y))
        lbl_home = Label(Home, text="Welcome! %s" % str(username.get()), font=('times new roman', 20)).pack()

        btn_1= Button(Home, text='Update', width=20, command=U.UpdateValues).place(x=100,y=100)

        btn_cont = Button(Home, text='View All Data', width=20, command=PostLogin.FarmerView).place(x=20, y= 150)
        btn_back = Button(Home, text='Logout', width=20, command=Back).place(x=200, y=150)

    def Back():
        Home.destroy()
        root.deiconify()


    root = Tk()
    root.title("Please Login")
    width = 400
    height = 280
    screen_width = root.winfo_screenwidth()
    screen_height = root.winfo_screenheight()
    x = (screen_width/2) - (width/2)
    y = (screen_height/2) - (height/2)
    root.geometry("%dx%d+%d+%d" % (width, height, x, y))
    root.resizable(0, 0)


    USERNAME = IntVar()
    PASSWORD = StringVar()
```

```
Top = Frame(root, bd=2,  relief=RIDGE)
Top.pack(side=TOP, fill=X)
Form = Frame(root, height=200)
Form.pack(side=TOP, pady=20)


lbl_title = Label(Top, text = "Kindly log in to continue", font=('arial', 15))
lbl_title.pack(fill=X)
lbl_username = Label(Form, text = "Unique ID:", font=('arial', 14), bd=15)
lbl_username.grid(row=0, sticky="e")
lbl_password = Label(Form, text = "Password:", font=('arial', 14), bd=15)
lbl_password.grid(row=1, sticky="e")
lbl_text = Label(Form)
lbl_text.grid(row=2, columnspan=2)


username = Entry(Form, textvariable=USERNAME, font=(14))
username.grid(row=0, column=1)
password = Entry(Form, textvariable=PASSWORD, show="*", font=(14))
password.grid(row=1, column=1)


btn_login = Button(Form, text="Login", width=45, command=Login)
btn_login.grid(pady=25, row=3, columnspan=2)
btn_login.bind('<Return>', Login)

root.mainloop()
```



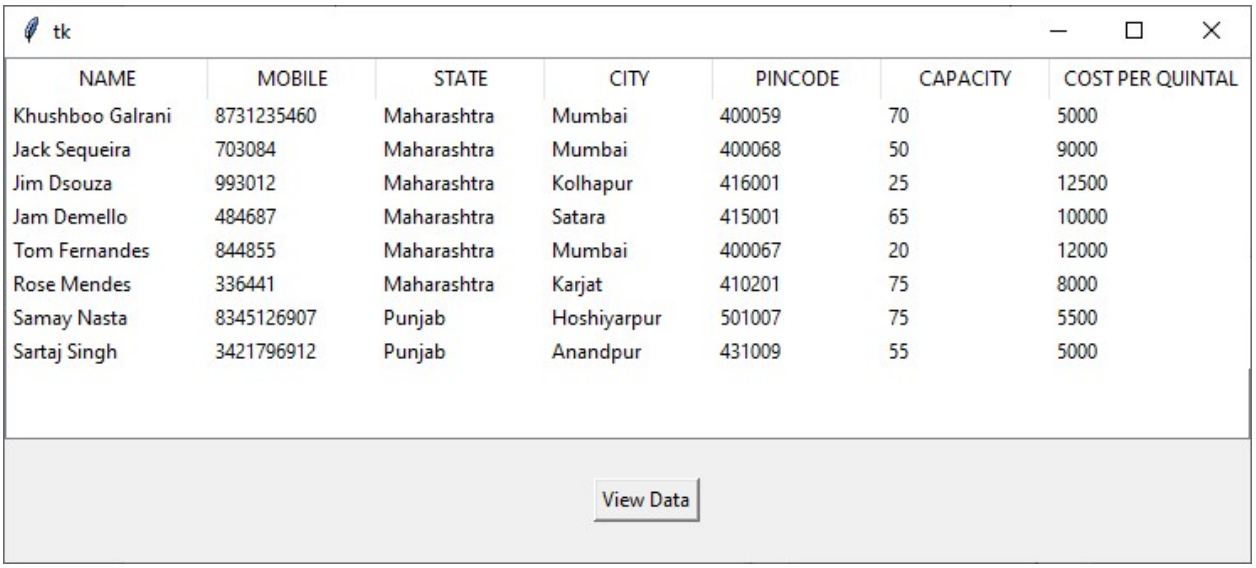




- **<u>View Section (Post Login):</u>**

```
from tkinter import *
import pymysql.cursors
from tkinter import ttk

def FarmerView():
    def PL():
            conn = pymysql.connect(host='localhost',user='root',password='WeRock',db='foodchainmanagement',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)
            try:
```

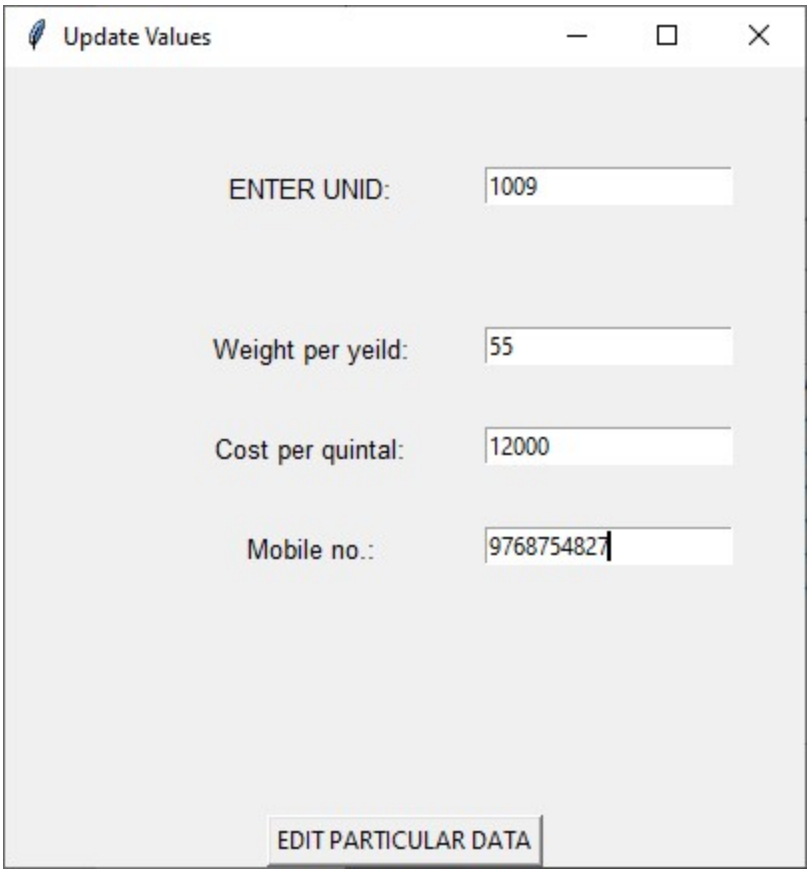
```
                with conn.cursor() as cursor:
                        sql = "SELECT Name,Mobile,State,City,Pincode,Capacity,Cost_per_quintal FROM Distributor"
                        cursor.execute(sql)
                        rows=cursor.fetchall()
                        for row in rows:
                                print(row)

    tree.insert("",END,values=(row["Name"],row["Mobile"],row["State"],row["City"],row["Pincode"],row["Capacity"],row["Cost_per_quintal"]))
            finally:
                    conn.close()

    FView = Tk()
    FView.geometry("740x300")

    tree= ttk.Treeview(FView, selectmode="extended", column=("Name", "","Mobile", "State","City","Pincode","Capacity","Cost_per_quintal"), show='headings')

    tree.heading("#1", text="NAME")
    tree.column("#1",minwidth=0,width=120)
    tree.heading("#2", text="MOBILE")
    tree.column("#2",minwidth=0,width=100)
    tree.heading("#3", text="STATE")
    tree.column("#3",minwidth=0,width=100)
    tree.heading("#4", text="CITY")
    tree.column("#4",minwidth=0,width=100)
    tree.heading("#5", text="PINCODE")
    tree.column("#5",minwidth=0,width=100)
    tree.heading("#6", text="CAPACITY")
    tree.column("#6",minwidth=0,width=100)
    tree.heading("#7", text="COST PER QUINTAL")
    tree.column("#7",minwidth=0,width=120)

    tree.pack(fill=BOTH)

    b2 = Button(FView, text="View Data", command=PL)
    b2.place(x=350,y=250)

    FView.mainloop()
```



- **Update Table:**

```
from tkinter import *
import pymysql

def UpdateValues():
    def Update():
            UNID = int(unid.get())
            data1 = int(e1.get())
            data2 = int(e2.get())
            data3 = e3.get()

            conn=
pymysql.connect(host='localhost',user='root',password='WeRock',db='foodchainmanagem
ent',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)
            cur = conn.cursor()
            cur.execute("UPDATE FARMER SET Weight_per_yeild=%d, Cost_per_quintal=%d,
Mobile='%s' WHERE Unid=%d" % (data1, data2, data3,UNID))
            print("VALUES UPDATED SUCCESSFULLY")
            conn.commit()
            conn.close()
            root.destroy()


    root = Tk()
    root.geometry("400x400")
    root.title("Update Values")
```

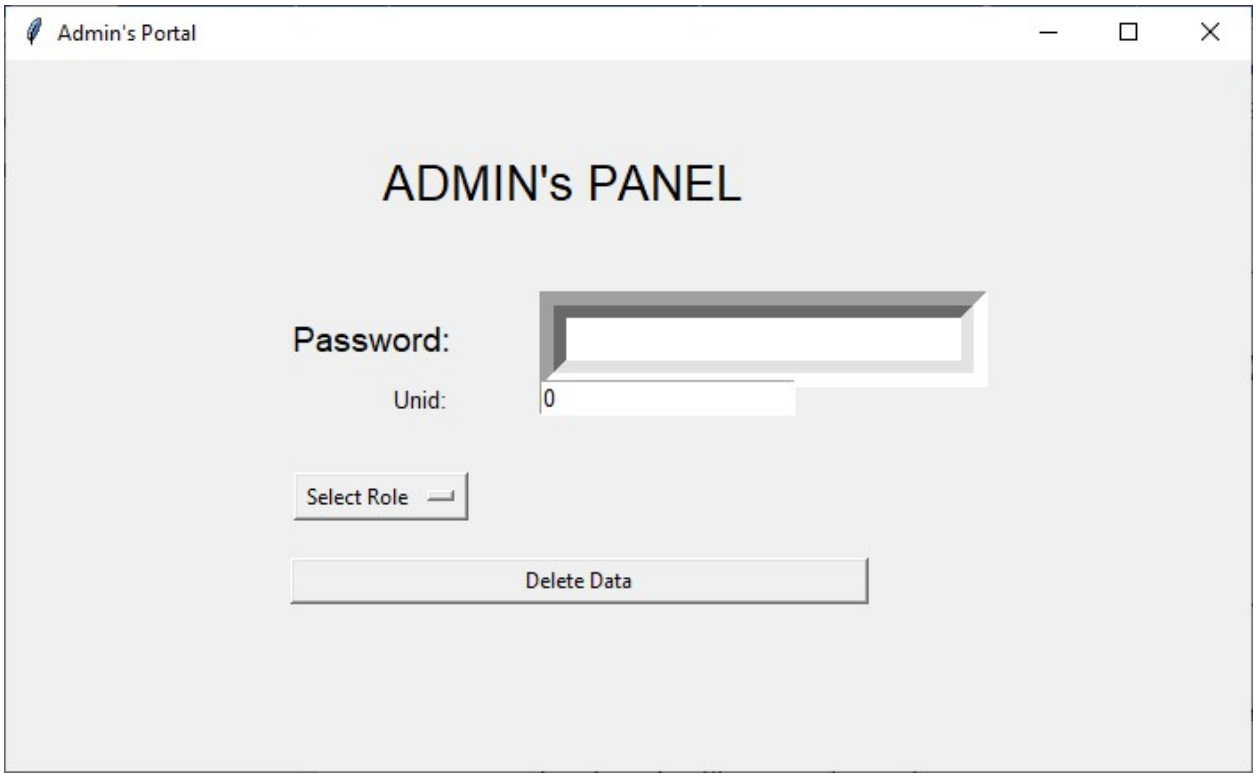
```
label_0 = Label(root, text="ENTER UNID:",width=20,font=("bold", 10))
label_0.place(x=70,y=50)

unid = IntVar()
unid = Entry(root, textvariable=unid)
unid.place(x=240,y=50)


label_1 = Label(root, text="Weight per yeild:",width=20,font=("bold", 10))
label_1.place(x=70,y=130)

wpy = IntVar()
e1 = Entry(root, textvariable=wpy)
e1.place(x=240,y=130)

label_2 = Label(root, text="Cost per quintal:",width=20,font=("bold", 10))
label_2.place(x=70,y=180)

cpq = IntVar()
e2 = Entry(root, textvariable=cpq)
e2.place(x=240,y=180)

label_3 = Label(root, text="Mobile no.:",width=20,font=("bold", 10))
label_3.place(x=70,y=230)

mobile = StringVar()
e3 = Entry(root, textvariable=mobile)
e3.place(x=240,y=230)

b2 = Button(root, text="EDIT PARTICULAR DATA", command=Update)
b2.pack(side=BOTTOM)

root.mainloop()
```

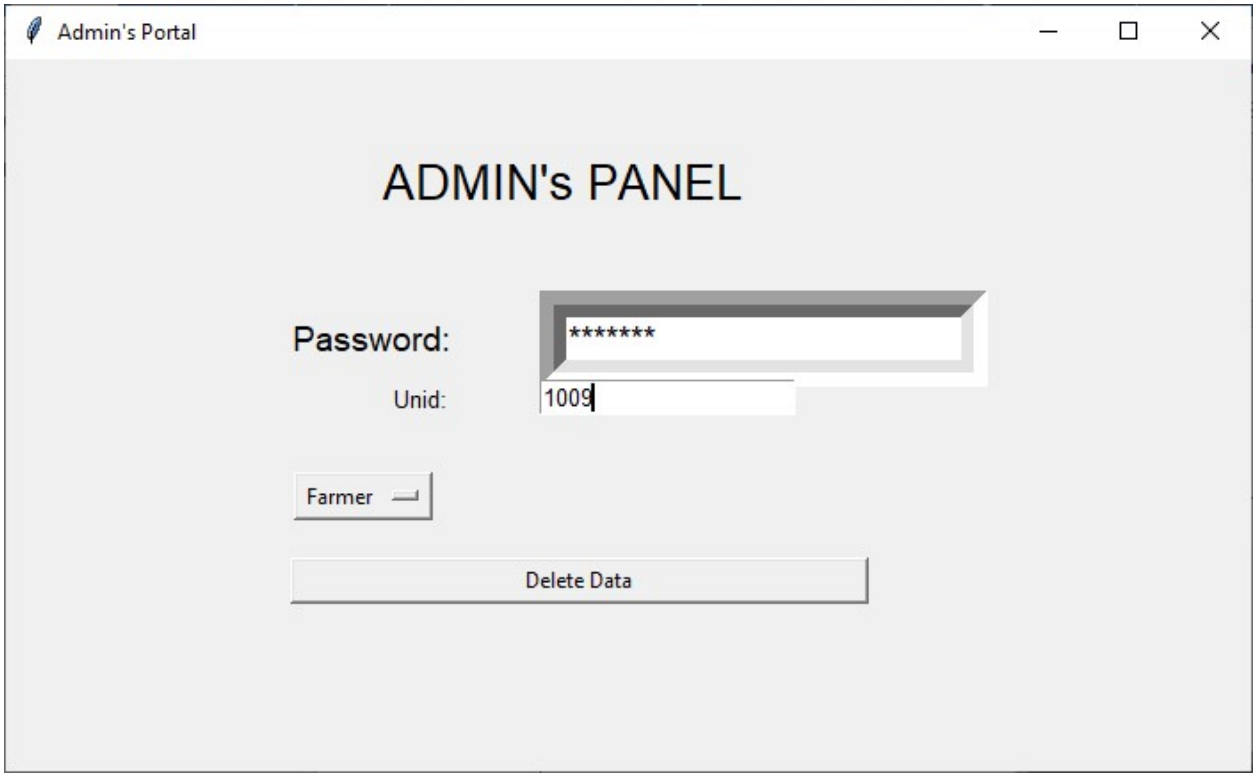


- **ADMIN PANEL:**

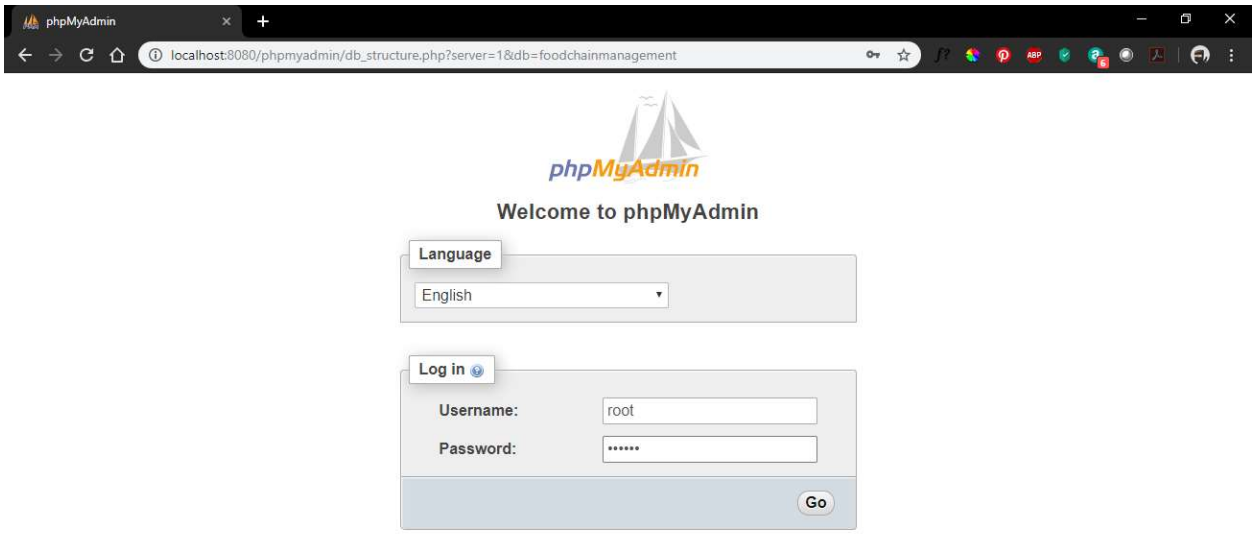
```
from tkinter import *
import pymysql.cursors

def Login():
    if(str(password.get())=="qwerty"):
            unid=int(entry_2.get())
            variable=str(var.get())
            conn=
pymysql.connect(host='localhost',user='root',password='WeRock',db='foodchainmanagem
ent',charset='utf8mb4',cursorclass=pymysql.cursors.DictCursor)
            cur = conn.cursor()
            cur.execute("DELETE FROM %s WHERE Unid=%d" % (variable,unid))
            print("VALUES DELETED SUCCESSFULLY")
            conn.commit()
            conn.close()
            root.destroy()

root = Tk()
```

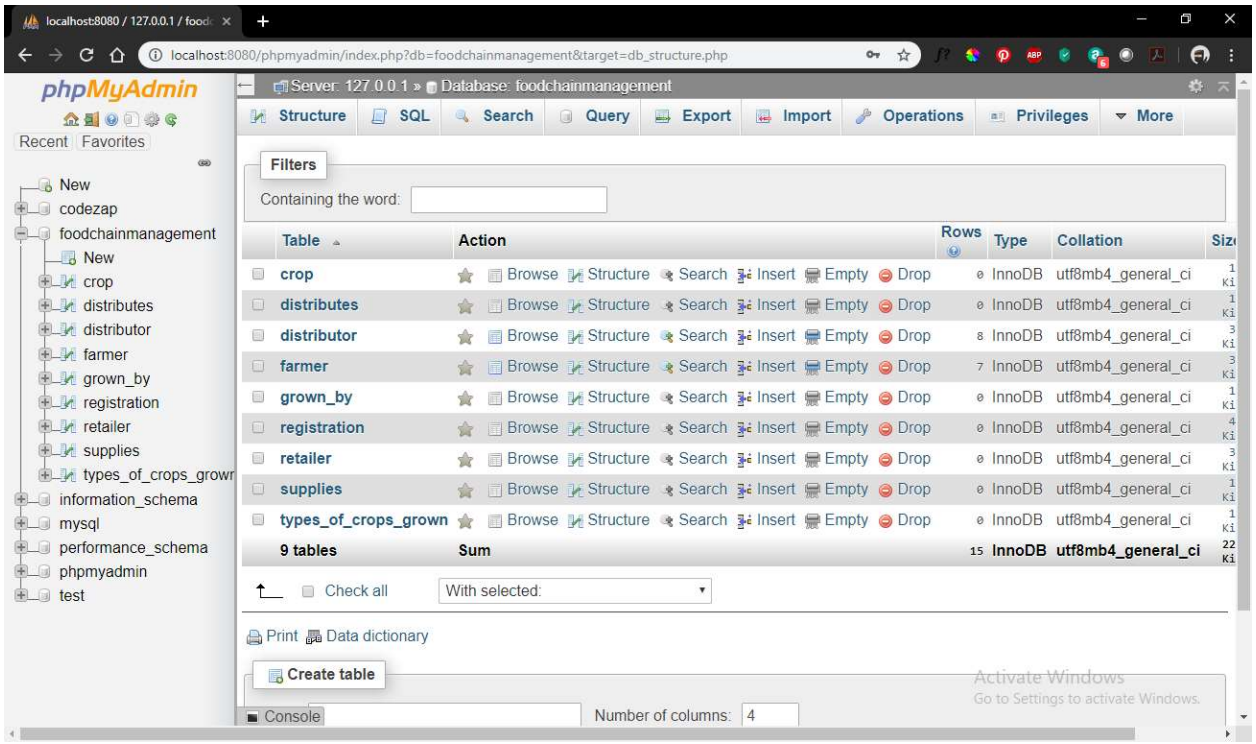
```
root.geometry("700x400")
root.title("Admin's Portal")

label_0 = Label(root, text="ADMIN's PANEL",width=20,font=("bold", 20))
label_0.place(x=150,y=50)


label_1 = Label(root, text="Password:",width=20,font=('arial', 14), bd=15)
label_1.place(x=80,y=130)

password=StringVar()
unid=IntVar()
var=StringVar(root)

entry_1 = Entry(root,textvar=password,show="*",font=('arial', 14), bd=15)
entry_1.place(x=300,y=130)

label_2 = Label(root, text="Unid:",width=20,font=("bold", 10))
label_2.place(x=150,y=180)

entry_2 = Entry(root,textvar=unid,font=("bold", 10))
entry_2.place(x=300,y=180)

var.set("Select Role")
option_1=OptionMenu(root, var, "Farmer", "Distributor", "Retailer")
option_1.place(x=160,y=230)

btn_login = Button(root, text="Delete Data", width=45, command=Login)
btn_login.place(x=160,y=280)
btn_login.bind('<Return>', Login)

root.mainloop()
```

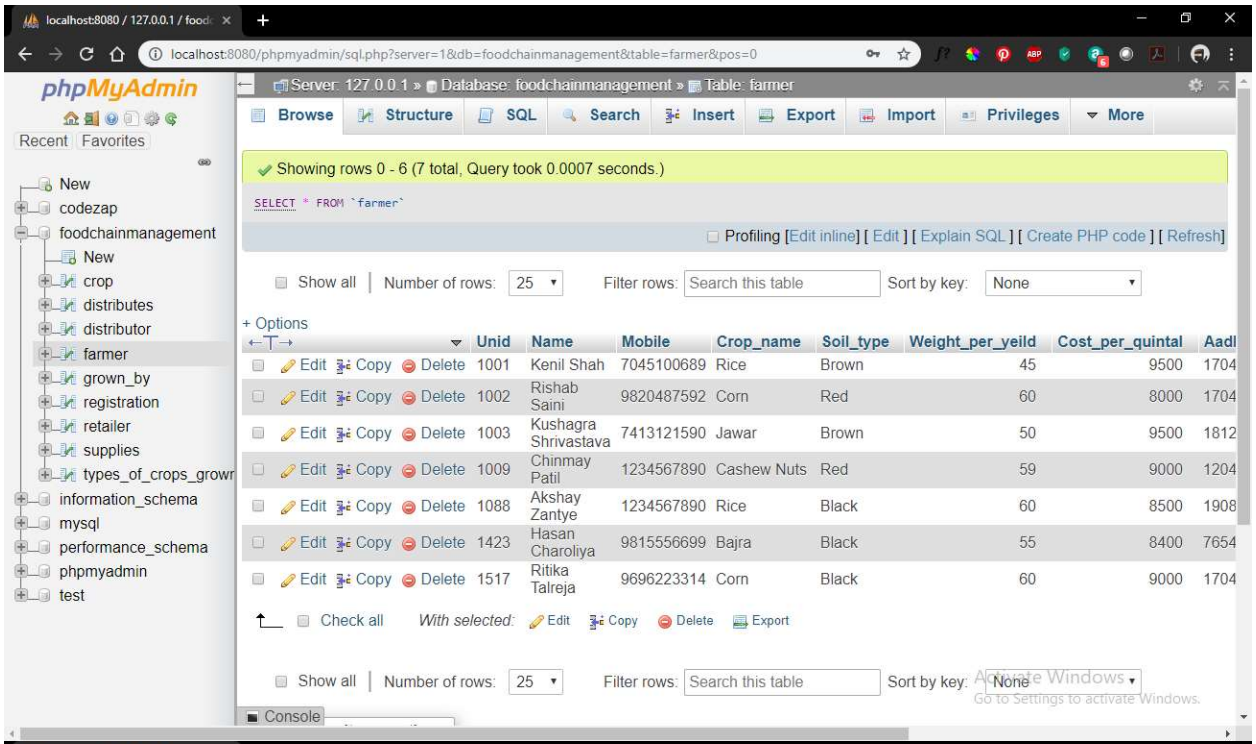

Admin's Portal
ADMIN's PANEL
Password:
Unid:
0
Select Role
Delete Data

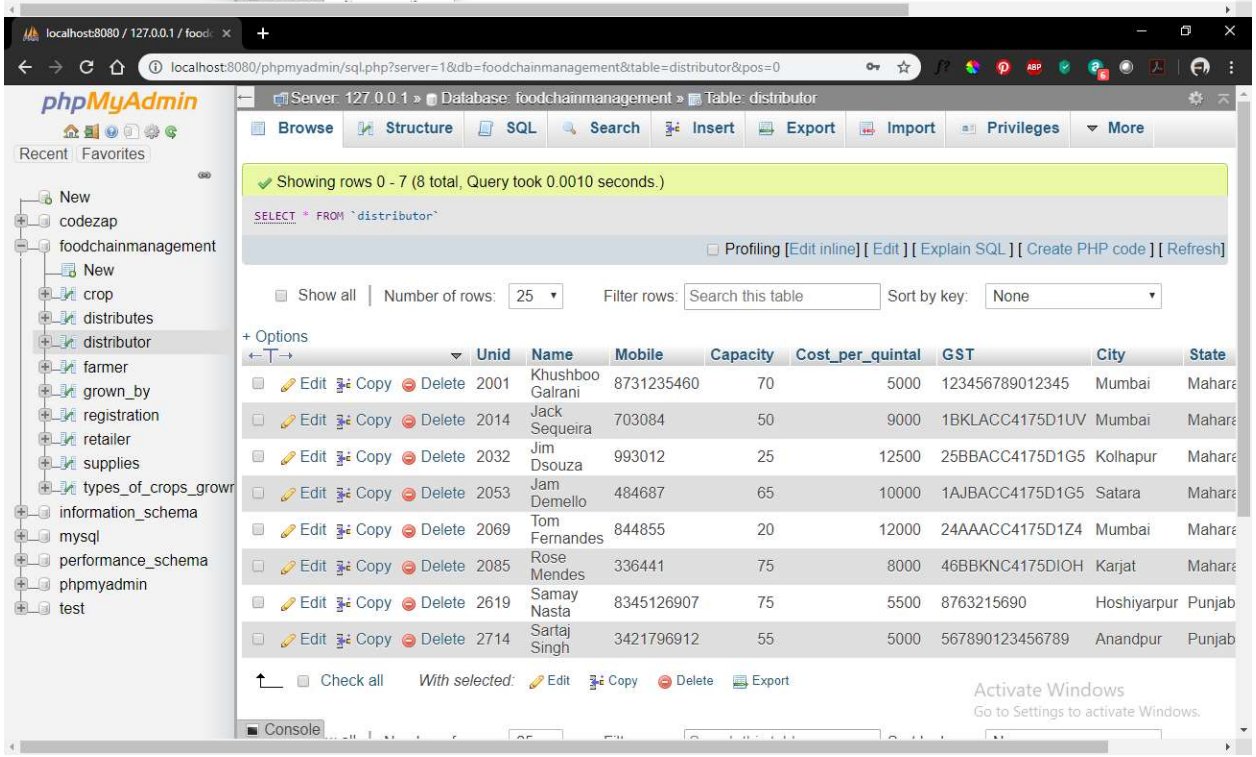

Admin's Portal
ADMIN's PANEL
Password:
****
Unid:
1009
Farmer
Delete Data

➢ SCREENSHOT OF THE DATABASE:






localhost:8080 / 127.0.0.1 / food
localhost:8080/phpmyadmin/sql.php?server=1&db=foodchainmanagement&table=farmer&pos=0
phpMyAdmin
Recent Favorites
New
codezap
foodchainmanagement
New
crop
distributes
distributor
farmer
grown_by
registration
retailer
supplies
types_of_crops_growr
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: foodchainmanagement » Table: farmer
Browse Structure SQL Search Insert Export Import Privileges More
Showing rows 0 - 6 (7 total, Query took 0.0007 seconds.)
SELECT * FROM `farmer`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
Unid Name Mobile Crop_name Soil_type Weight_per_yeild Cost_per_quintal Aadl
Edit Copy Delete 1001 Kenil Shah 7045100689 Rice Brown 45 9500 1704
Edit Copy Delete 1002 Rishab Saini 9820487592 Corn Red 60 8000 1704
Edit Copy Delete 1003 Kushagra Shrivastava 7413121590 Jawar Brown 50 9500 1812
Edit Copy Delete 1009 Chinmay Patil 1234567890 Cashew Nuts Red 59 9000 1204
Edit Copy Delete 1088 Akshay Zantye 1234567890 Rice Black 60 8500 1908
Edit Copy Delete 1423 Hasan Charoliya 9815556699 Bajra Black 55 8400 7654
Edit Copy Delete 1517 Ritika Talreja 9696223314 Corn Black 60 9000 1704
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
Console


localhost:8080 / 127.0.0.1 / food
localhost:8080/phpmyadmin/sql.php?server=1&db=foodchainmanagement&table=distributor&pos=0
phpMyAdmin
Recent Favorites
New
codezap
foodchainmanagement
New
crop
distributes
distributor
farmer
grown_by
registration
retailer
supplies
types_of_crops_growr
information_schema
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: foodchainmanagement » Table: distributor
Browse Structure SQL Search Insert Export Import Privileges More
Showing rows 0 - 7 (8 total, Query took 0.0010 seconds.)
SELECT * FROM `distributor`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options
Unid Name Mobile Capacity Cost_per_quintal GST City State
Edit Copy Delete 2001 Khushboo Galrani 8731235460 70 5000 123456789012345 Mumbai Mahara
Edit Copy Delete 2014 Jack Sequeira 703084 50 9000 1BKLACC4175D1UV Mumbai Mahara
Edit Copy Delete 2032 Jim Dsouza 993012 25 12500 25BBACC4175D1G5 Kolhapur Mahara
Edit Copy Delete 2053 Jam Demello 484687 65 10000 1AJBACC4175D1G5 Satara Mahara
Edit Copy Delete 2069 Tom Fernandes 844855 20 12000 24AAACC4175D1Z4 Mumbai Mahara
Edit Copy Delete 2085 Rose Mendes 336441 75 8000 46BBKNC4175DIOH Karjat Mahara
Edit Copy Delete 2619 Samay Nasta 8345126907 75 5500 8763215690 Hoshiyarpur Punjab
Edit Copy Delete 2714 Sartaj Singh 3421796912 55 5000 567890123456789 Anandpur Punjab
Check all With selected: Edit Copy Delete Export
Console