## Title: The Emergency Lane

## Abstract:

def __init__(self,root):

= it is magic function, using this we can make title visible.

def login(self):

= SQL connect with our app by this, if get right mail/pass they let us go, otherwise they will warn us.

def admin-window(self):

= Helps to go next page.

def map-window(self):

=Helps to go next page.

def add-info(self):

=It helps to add traffic police in SQL.

def fetch-data(self):

=it brings information to text box.

def remove(self):

=Remove a traffic police.

def update(self):

=Update a traffic police.

## Introduction:

We imported imagetk, pymysql, message box, ttk, PIL. Then in the project, we introduce a magic function to make our dashboard visible along with the login page. Here we have 2 boxes for REPORT inserting text us login user name and password. Then we are able to login here because we have data that can help us to log in and set-up previously by an Admin. Everything is connected to the SQL.

An admin can add, update, or delete data along with, creating the user. Login data will carry the information about an admin and traffic police. On the other hand, by login in as a traffic man, we can set our location by using GPS and zoom the road and started finding the emergency vehicle. If traffic faces a critical situation, he will ask the admin for emergency help. This is how we can make a chain of role in the road to introduce ourselves with a new emergency lane to save a life.

## Problem Statement:

Here in Bangladesh, we have fully failed to provide a safe road to the people of this country. Emergency lanes for emergency vehicles, particular lanes for different vehicles are the hidden features to us. Even the people who own the driving license didn't know about the emergency lane before 2028. 'Nirapod Shorok Andolon' opened our eyes and here is the problem of the solution to managing the emergency lane. Maintaining traffic in Dhaka is not an easy task for traffic police. Hence in this situation, we come forward to save a life, tired to ensure an emergency vehicle like an ambulance can get a chance to move where thousands of cars are already stuck in the jam. Our App will help the ambulance to be visible to the traffic police so that they could get a chance to rescue the ambulance and save their lives.

## Lit Review:

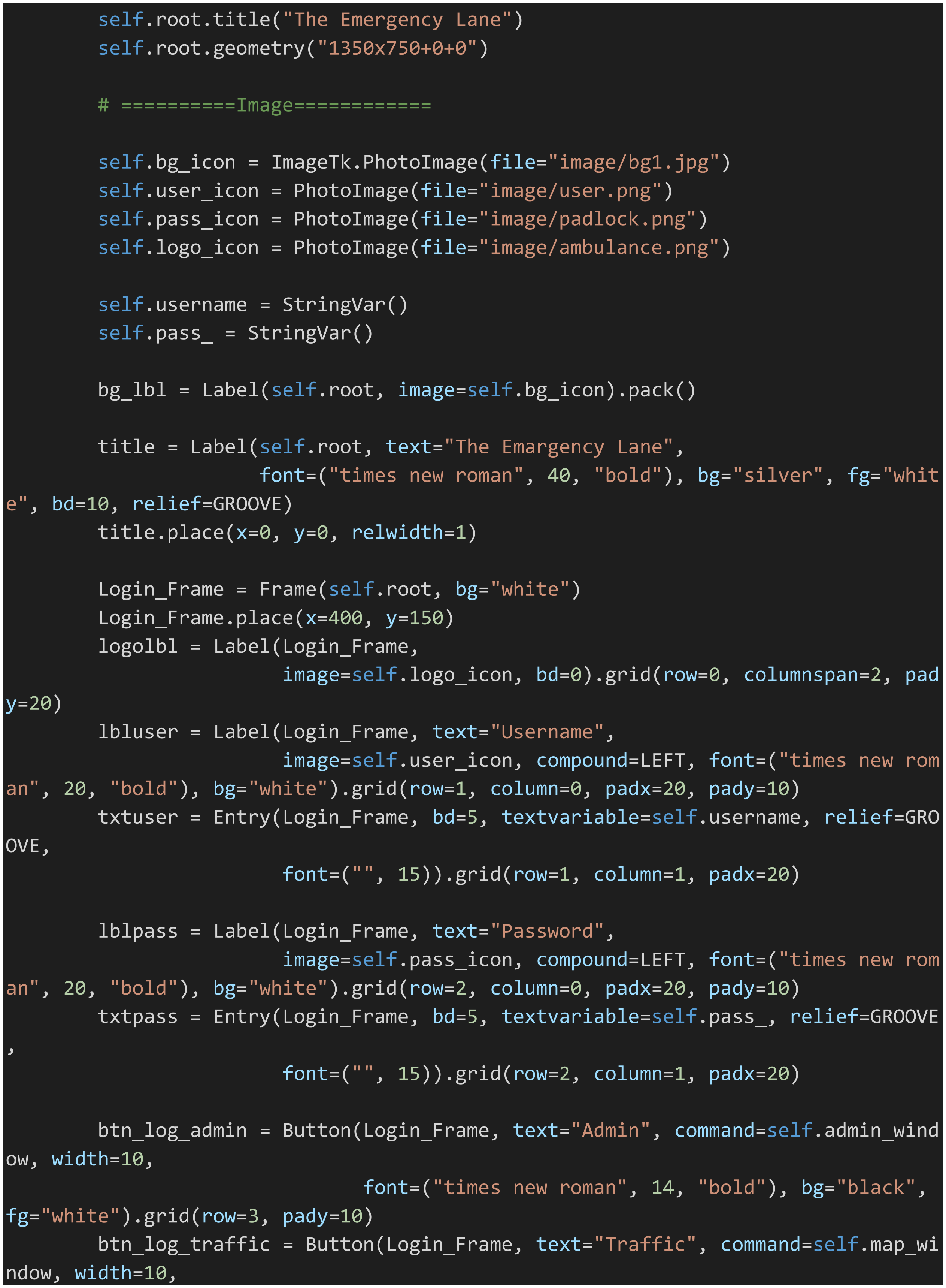
❖ Currently, we don't have this kind of APP concept. This is a new concept, we tried to introduce a new idea and want to do something for Bangladesh.

❖ The Ambulance or emergency vehicle tracking system tracker might be as similar as GPS Tracker.

## Code:

#Login page

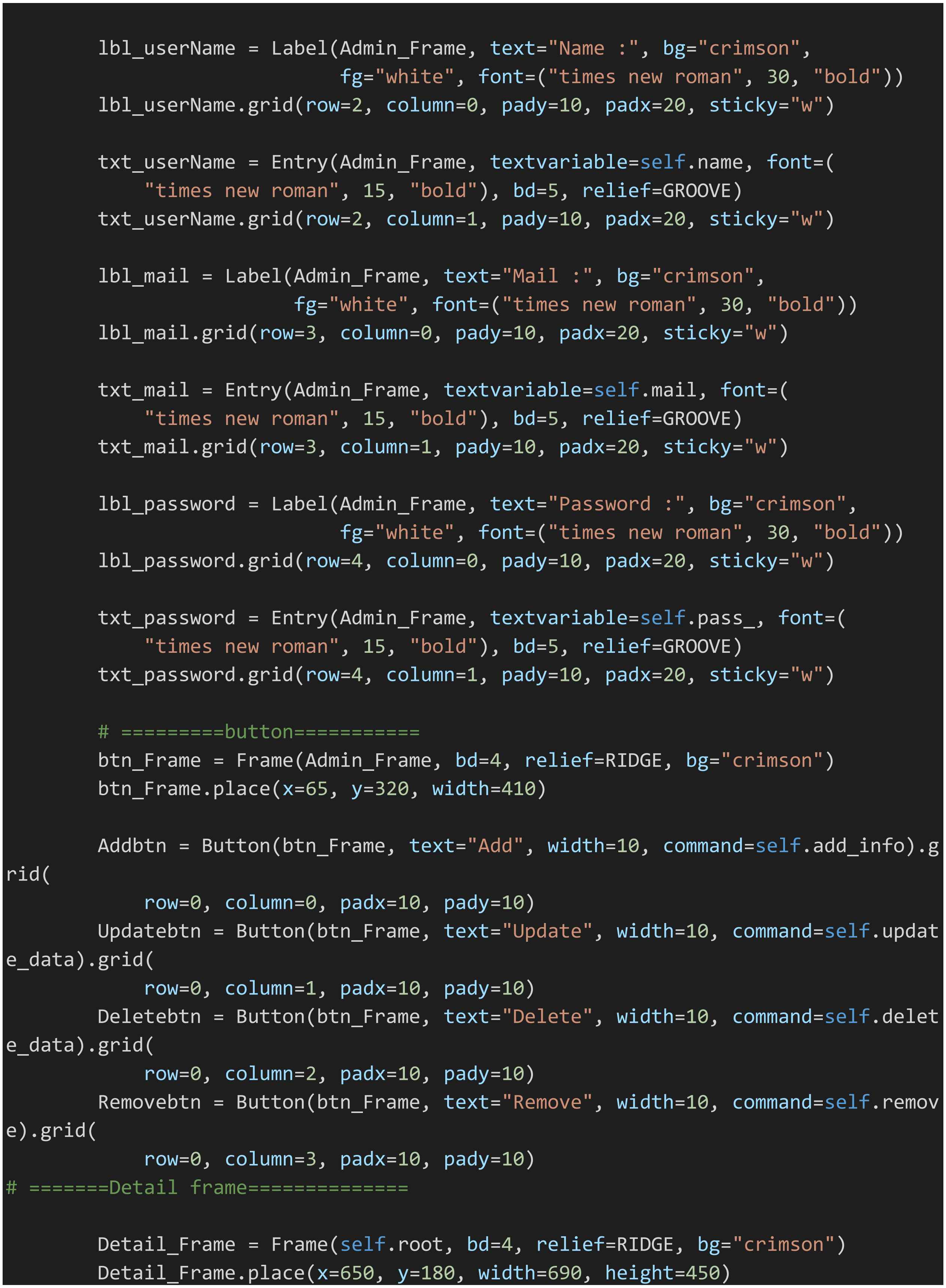
```
from tkinter import*
from PIL import ImageTk
import pymysql
from tkinter import messagebox


class The_Emargency_Lane:
    def __init__(self, root):
        self.root = root
```



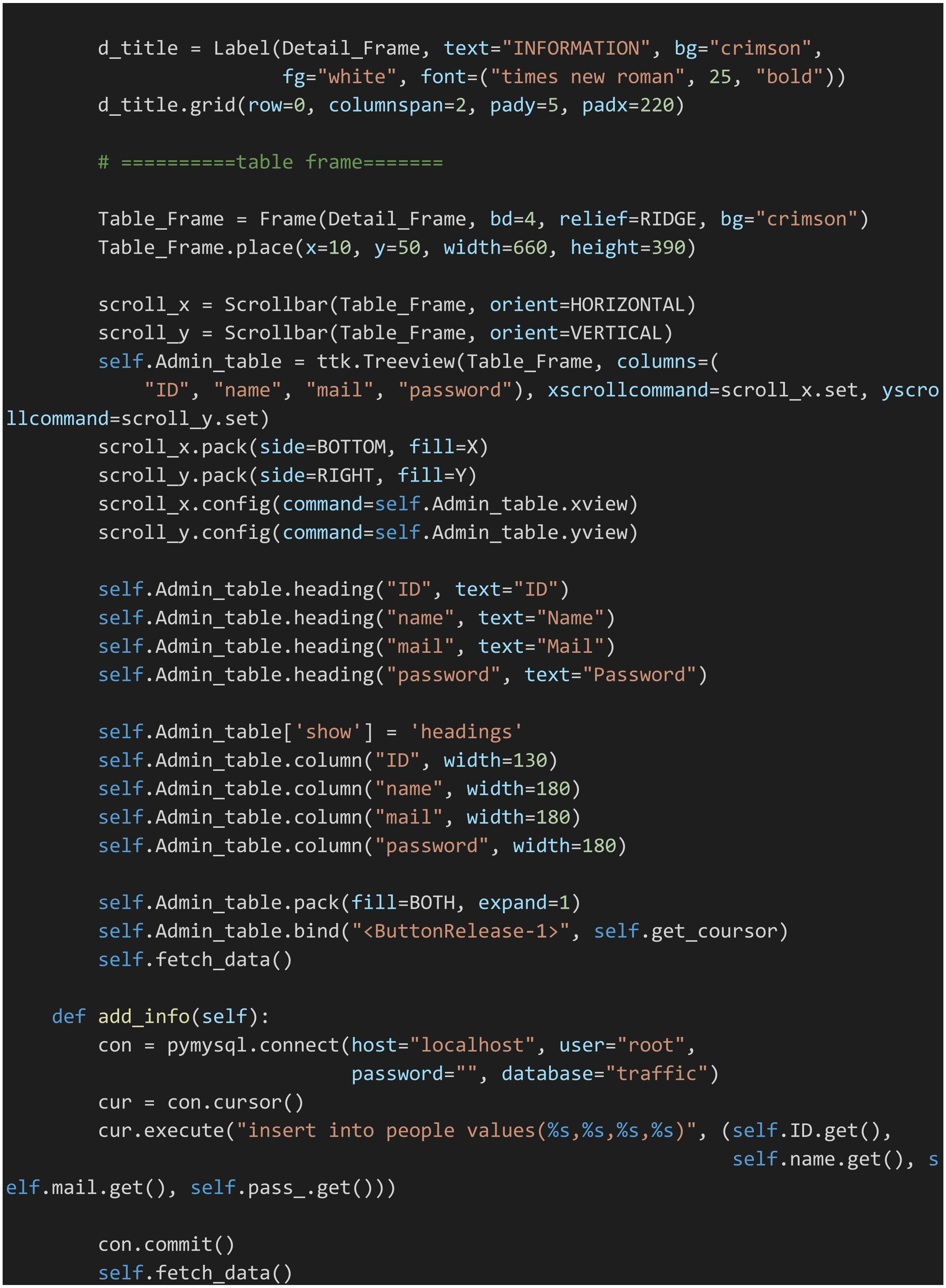
```
        self.root.title("The Emergency Lane")
        self.root.geometry("1350x750+0+0")

        # ==========Image============

        self.bg_icon = ImageTk.PhotoImage(file="image/bg1.jpg")
        self.user_icon = PhotoImage(file="image/user.png")
        self.pass_icon = PhotoImage(file="image/padlock.png")
        self.logo_icon = PhotoImage(file="image/ambulance.png")

        self.username = StringVar()
        self.pass_ = StringVar()

        bg_lbl = Label(self.root, image=self.bg_icon).pack()

        title = Label(self.root, text="The Emargency Lane",
                      font=("times new roman", 40, "bold"), bg="silver", fg="whit
e", bd=10, relief=GROOVE)
        title.place(x=0, y=0, relwidth=1)

        Login_Frame = Frame(self.root, bg="white")
        Login_Frame.place(x=400, y=150)
        logolbl = Label(Login_Frame,
                        image=self.logo_icon, bd=0).grid(row=0, columnspan=2, pad
y=20)
        lbluser = Label(Login_Frame, text="Username",
                        image=self.user_icon, compound=LEFT, font=("times new rom
an", 20, "bold"), bg="white").grid(row=1, column=0, padx=20, pady=10)
        txtuser = Entry(Login_Frame, bd=5, textvariable=self.username, relief=GRO
OVE,
                        font=("", 15)).grid(row=1, column=1, padx=20)

        lblpass = Label(Login_Frame, text="Password",
                        image=self.pass_icon, compound=LEFT, font=("times new rom
an", 20, "bold"), bg="white").grid(row=2, column=0, padx=20, pady=10)
        txtpass = Entry(Login_Frame, bd=5, textvariable=self.pass_, relief=GROOVE
,
                        font=("", 15)).grid(row=2, column=1, padx=20)

        btn_log_admin = Button(Login_Frame, text="Admin", command=self.admin_wind
ow, width=10,
                               font=("times new roman", 14, "bold"), bg="black",
fg="white").grid(row=3, pady=10)
        btn_log_traffic = Button(Login_Frame, text="Traffic", command=self.map_wi
ndow, width=10,
                                 font=("times new roman", 14, "bold"), bg="black"
, fg="white").grid(row=3, column=1, pady=10)

    def login(self):
        if self.username.get() == "" or self.pass_.get() == "":
            messagebox.showerror(
                "Error", "All fields are required", parent=self.root)
        else:
            try:
                con = pymysql.connect(
                    host="localhost", user="root", password="", database="traffic
")
                cur = con.cursor()
                cur.execute("select * from admin where userName=%s and password=%
s",
                            (self.username.get(), self.pass_.get()))
                row = cur.fetchone()
                if row == None:
                    messagebox.showerror(
                        "Error", "Invalid USERNAME OR PASSWORD", parent=self.root
)
                else:
                    messagebox.showinfo("Welcome", parent=self.root)
                con.close()
            except Exception as es:
                messagebox.showerror(
                    "Error", f"Error Due to: {str(es)}", parent=self.root)

    def admin_window(self):
        self.root.destroy()
        import admin

    def map_window(self):
        self.root.destroy()
        import map


root = Tk()
obj = The_Emargency_Lane(root)
root.mainloop()
```

#admin page

```
from tkinter import *
from PIL import ImageTk
import pymysql
from tkinter import messagebox
from tkinter import ttk


class admin:
    def __init__(self, root):
        self.root = root
        self.root.title("The Emergency Lane")
        self.root.geometry("1650x800+0+0")

        self.bg_icon = ImageTk.PhotoImage(file="image/bg1.jpg")

        bg_lbl = Label(self.root, image=self.bg_icon).pack()
        title = Label(self.root, text="Admin Panel",
                      font=("times new roman", 40, "bold"), bg="silver", fg="whit
e", bd=10, relief=GROOVE)
        title.place(x=0, y=0, relwidth=1)

# ========variable======

        self.ID = StringVar()
        self.name = StringVar()
        self.mail = StringVar()
        self.pass_ = StringVar()

# ==========admin frame===============

        Admin_Frame = Frame(self.root, bd=4, relief=RIDGE, bg="crimson")
        Admin_Frame.place(x=40, y=180, width=550, height=450)

        # a_title = Label(Admin_Frame, text="ID :", bg="crimson",
        # fg="white", font=("times new roman", 30, "bold"))
        # a_title.grid(row=20, columnspan=2, pady=20)
        lbl_id = Label(Admin_Frame, text="ID :", bg="crimson",
                       fg="white", font=("times new roman", 30, "bold"))
        lbl_id.grid(row=1, column=0, pady=10, padx=20, sticky="w")

        txt_id = Entry(Admin_Frame, textvariable=self.ID, font=(
            "times new roman", 15, "bold"), bd=5, relief=GROOVE)
        txt_id.grid(row=1, column=1, pady=10, padx=20, sticky="w")
```



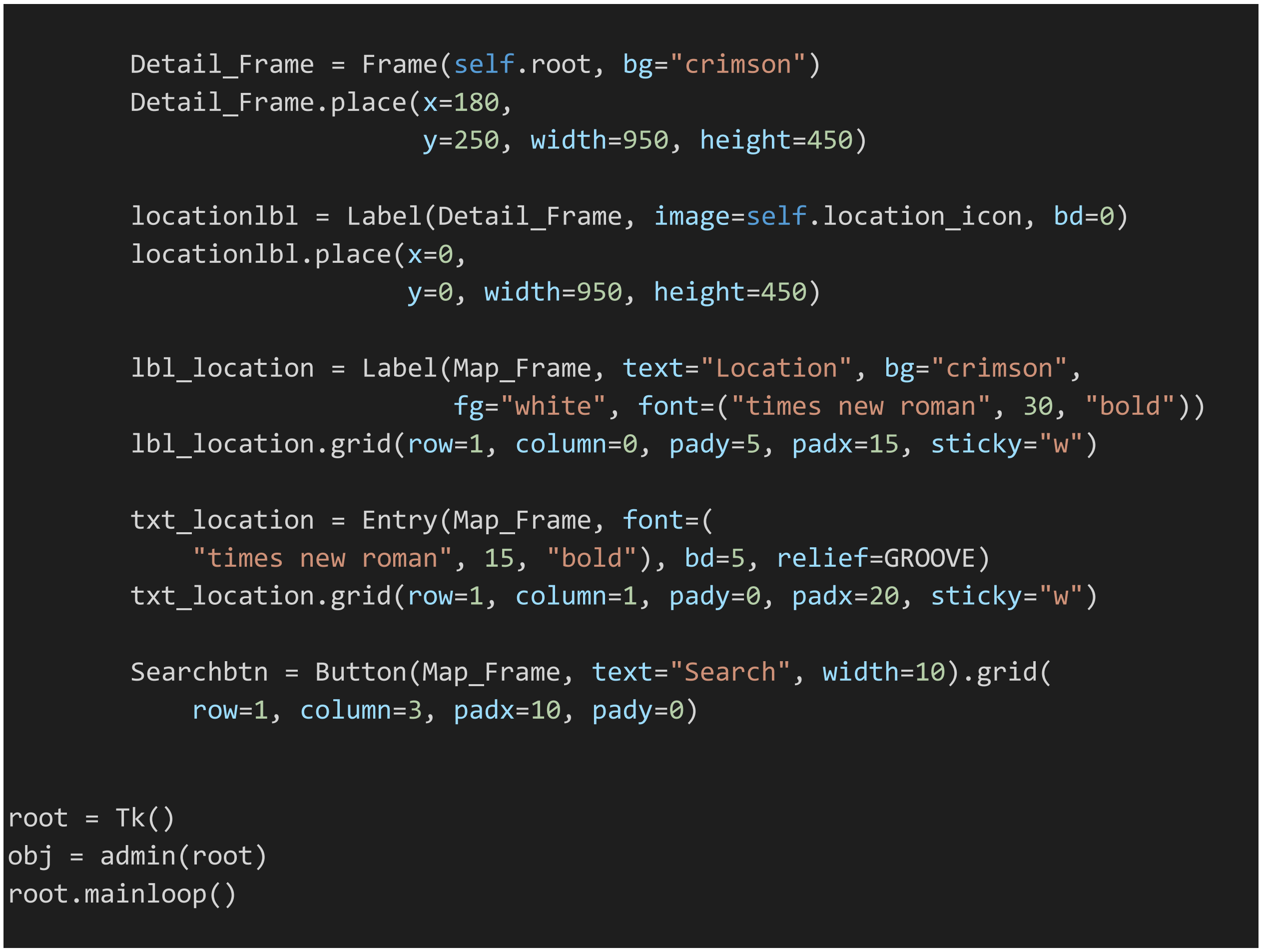
```
        lbl_userName = Label(Admin_Frame, text="Name :", bg="crimson",
                             fg="white", font=("times new roman", 30, "bold"))
        lbl_userName.grid(row=2, column=0, pady=10, padx=20, sticky="w")

        txt_userName = Entry(Admin_Frame, textvariable=self.name, font=(
            "times new roman", 15, "bold"), bd=5, relief=GROOVE)
        txt_userName.grid(row=2, column=1, pady=10, padx=20, sticky="w")

        lbl_mail = Label(Admin_Frame, text="Mail :", bg="crimson",
                         fg="white", font=("times new roman", 30, "bold"))
        lbl_mail.grid(row=3, column=0, pady=10, padx=20, sticky="w")

        txt_mail = Entry(Admin_Frame, textvariable=self.mail, font=(
            "times new roman", 15, "bold"), bd=5, relief=GROOVE)
        txt_mail.grid(row=3, column=1, pady=10, padx=20, sticky="w")

        lbl_password = Label(Admin_Frame, text="Password :", bg="crimson",
                             fg="white", font=("times new roman", 30, "bold"))
        lbl_password.grid(row=4, column=0, pady=10, padx=20, sticky="w")

        txt_password = Entry(Admin_Frame, textvariable=self.pass_, font=(
            "times new roman", 15, "bold"), bd=5, relief=GROOVE)
        txt_password.grid(row=4, column=1, pady=10, padx=20, sticky="w")

        # =========button===========
        btn_Frame = Frame(Admin_Frame, bd=4, relief=RIDGE, bg="crimson")
        btn_Frame.place(x=65, y=320, width=410)

        Addbtn = Button(btn_Frame, text="Add", width=10, command=self.add_info).g
rid(
            row=0, column=0, padx=10, pady=10)
        Updatebtn = Button(btn_Frame, text="Update", width=10, command=self.updat
e_data).grid(
            row=0, column=1, padx=10, pady=10)
        Deletebtn = Button(btn_Frame, text="Delete", width=10, command=self.delet
e_data).grid(
            row=0, column=2, padx=10, pady=10)
        Removebtn = Button(btn_Frame, text="Remove", width=10, command=self.remov
e).grid(
            row=0, column=3, padx=10, pady=10)
# =======Detail frame==============

        Detail_Frame = Frame(self.root, bd=4, relief=RIDGE, bg="crimson")
        Detail_Frame.place(x=650, y=180, width=690, height=450)
```



```
        d_title = Label(Detail_Frame, text="INFORMATION", bg="crimson",
                        fg="white", font=("times new roman", 25, "bold"))
        d_title.grid(row=0, columnspan=2, pady=5, padx=220)

        # ==========table frame=======

        Table_Frame = Frame(Detail_Frame, bd=4, relief=RIDGE, bg="crimson")
        Table_Frame.place(x=10, y=50, width=660, height=390)

        scroll_x = Scrollbar(Table_Frame, orient=HORIZONTAL)
        scroll_y = Scrollbar(Table_Frame, orient=VERTICAL)
        self.Admin_table = ttk.Treeview(Table_Frame, columns=(
            "ID", "name", "mail", "password"), xscrollcommand=scroll_x.set, yscro
llcommand=scroll_y.set)
        scroll_x.pack(side=BOTTOM, fill=X)
        scroll_y.pack(side=RIGHT, fill=Y)
        scroll_x.config(command=self.Admin_table.xview)
        scroll_y.config(command=self.Admin_table.yview)

        self.Admin_table.heading("ID", text="ID")
        self.Admin_table.heading("name", text="Name")
        self.Admin_table.heading("mail", text="Mail")
        self.Admin_table.heading("password", text="Password")

        self.Admin_table['show'] = 'headings'
        self.Admin_table.column("ID", width=130)
        self.Admin_table.column("name", width=180)
        self.Admin_table.column("mail", width=180)
        self.Admin_table.column("password", width=180)

        self.Admin_table.pack(fill=BOTH, expand=1)
        self.Admin_table.bind("<ButtonRelease-1>", self.get_coursor)
        self.fetch_data()

    def add_info(self):
        con = pymysql.connect(host="localhost", user="root",
                              password="", database="traffic")
        cur = con.cursor()
        cur.execute("insert into people values(%s,%s,%s,%s)", (self.ID.get(),
                                                               self.name.get(), s
elf.mail.get(), self.pass_.get()))

        con.commit()
        self.fetch_data()
        self.remove()
        con.close()

    def fetch_data(self):
        con = pymysql.connect(host="localhost", user="root",
                              password="", database="traffic")
        cur = con.cursor()
        cur.execute("select * from people")
        rows = cur.fetchall()
        if len(rows) != 0:
            self.Admin_table.delete(*self.Admin_table.get_children())
            for row in rows:
                self.Admin_table.insert('', END, values=row)
            con.commit()
        con.close()

    def remove(self):
        self.ID.set("")
        self.name.set("")
        self.mail.set("")
        self.pass_.set("")

    def get_coursor(self, ev):
        coursor_row = self.Admin_table.focus()
        contents = self.Admin_table.item(coursor_row)
        row = contents['values']
        self.ID.set(row[0])
        self.name.set(row[1])
        self.mail.set(row[2])
        self.pass_.set(row[3])

    def update_data(self):
        con = pymysql.connect(host="localhost", user="root",
                              password="", database="traffic")
        cur = con.cursor()
        cur.execute("update people set name=%s,mail=%s,pass=%s where ID=%s",
                    (self.name.get(), self.mail.get(), self.pass_.get(), self.ID.
get()))

        con.commit()
        self.fetch_data()
        self.remove()
        con.close()

    def delete_data(self):
        con = pymysql.connect(host="localhost", user="root",
                              password="", database="traffic")
        cur = con.cursor()
        cur.execute("delete from people where ID=%s", self.ID.get())
        con.commit()
        con.close()
        self.fetch_data()
        self.remove()


root = Tk()
obj = admin(root)
root.mainloop()
```

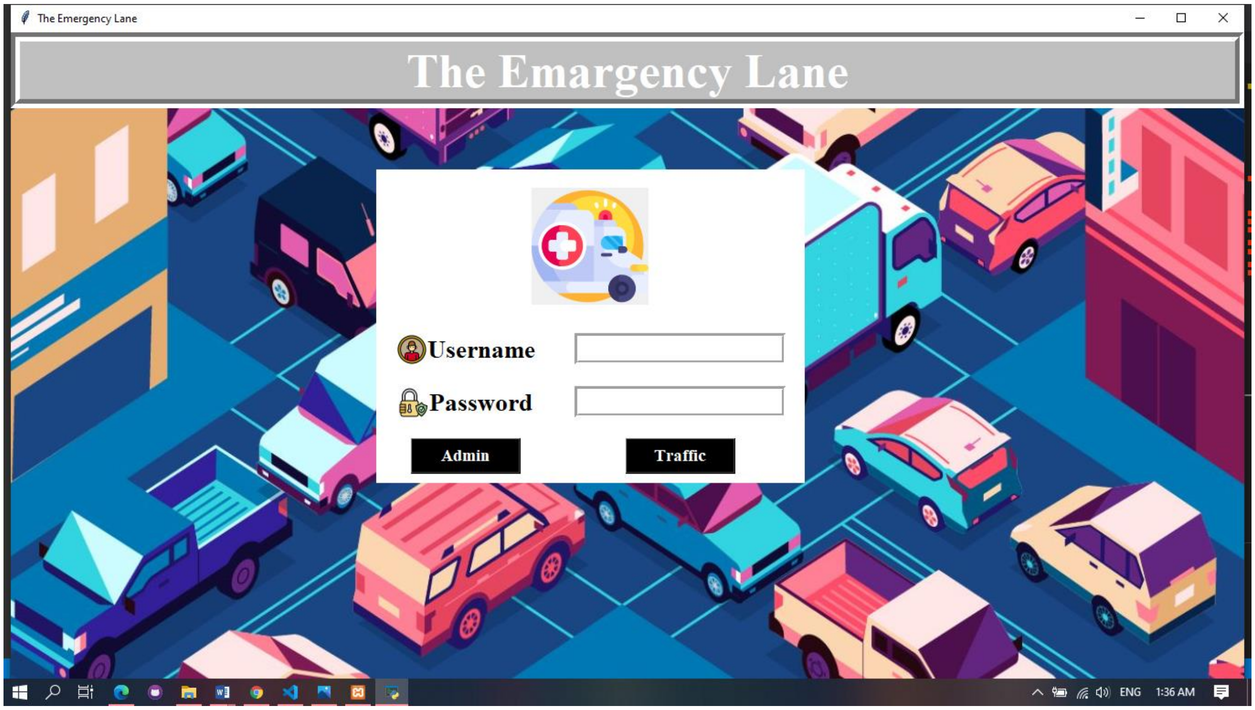
#map page

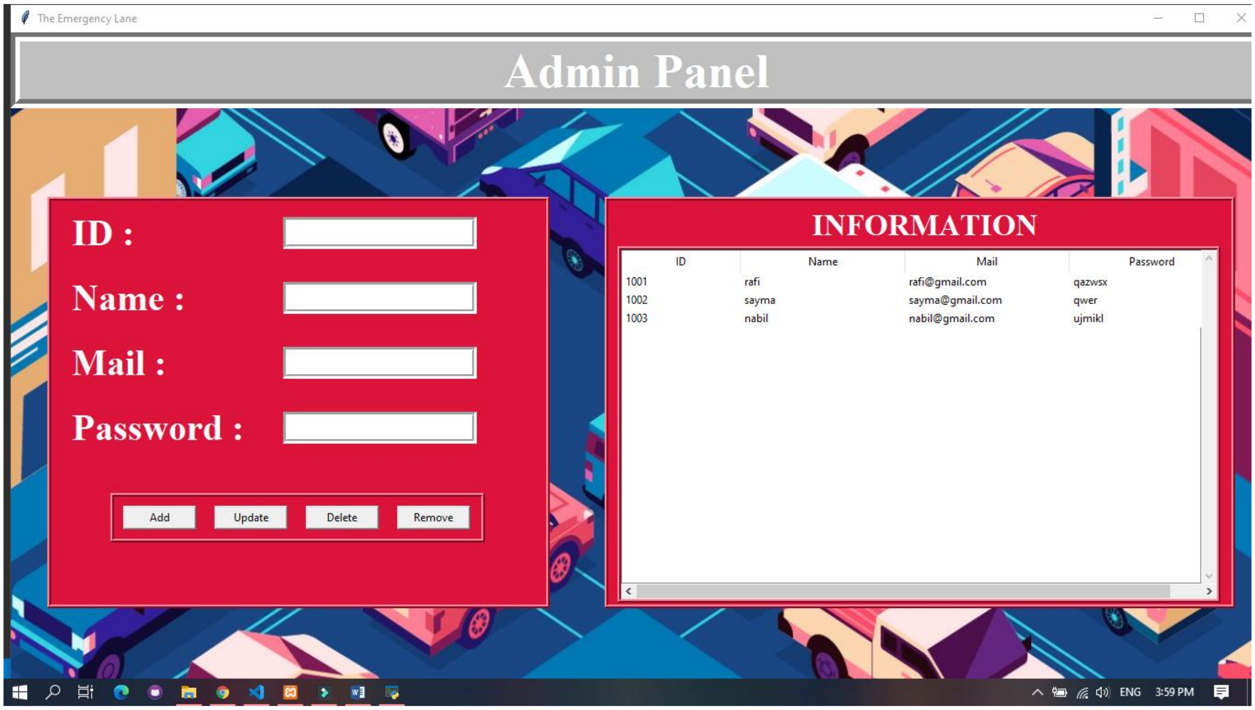
```
from tkinter import*
from PIL import ImageTk, Image
import pymysql
from tkinter import messagebox
from tkinter import ttk


class admin:
    def __init__(self, root):
        self.root = root
        self.root.title("The Emergency Lane")
        self.root.geometry("1650x800+0+0")

        self.bg_icon = ImageTk.PhotoImage(file="image/bg1.jpg")

        self.location_icon = ImageTk.PhotoImage(file="image/Location1.jpg")

        bg_lbl = Label(self.root, image=self.bg_icon).pack()

        title = Label(self.root, text="Map",
                      font=("times new roman", 40, "bold"), bg="silver", fg="whit
e", bd=10, relief=GROOVE)
        title.place(x=0, y=0, relwidth=1)

        Map_Frame = Frame(self.root, bd=4, relief=RIDGE, bg="crimson")
        Map_Frame.place(x=400, y=150, width=550, height=70)
```



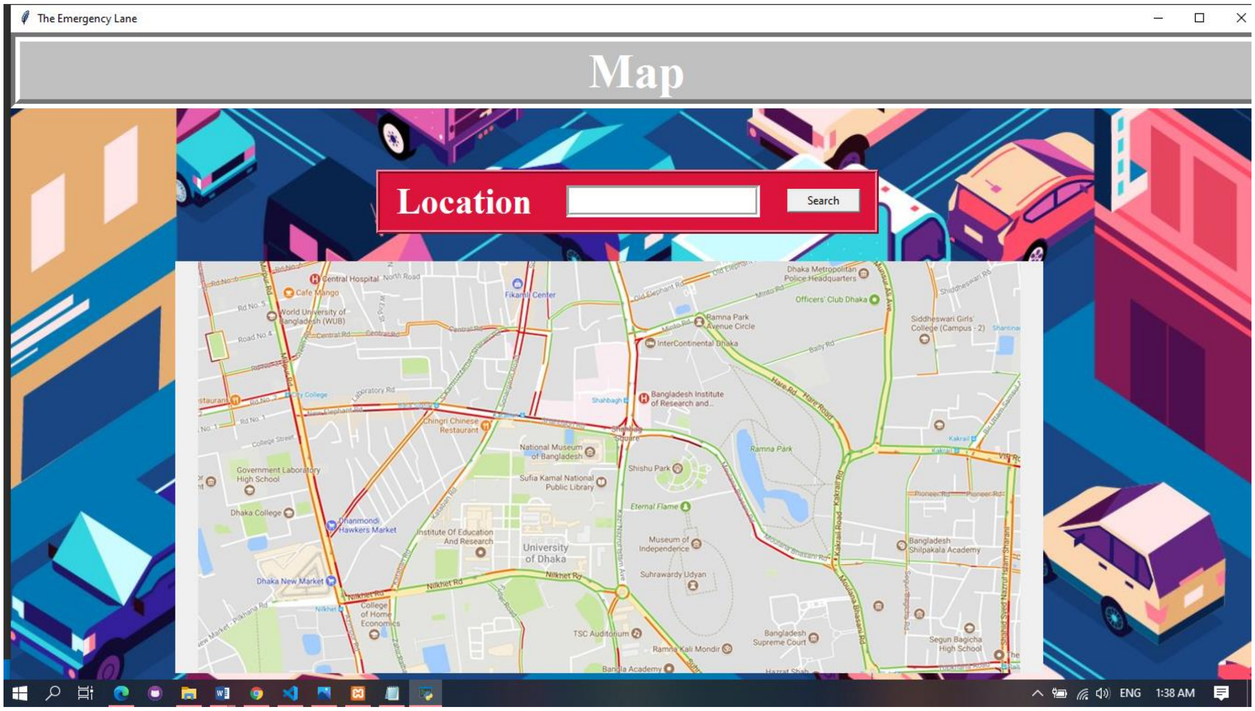
```
        Detail_Frame = Frame(self.root, bg="crimson")
        Detail_Frame.place(x=180,
                           y=250, width=950, height=450)

        locationlbl = Label(Detail_Frame, image=self.location_icon, bd=0)
        locationlbl.place(x=0,
                          y=0, width=950, height=450)

        lbl_location = Label(Map_Frame, text="Location", bg="crimson",
                             fg="white", font=("times new roman", 30, "bold"))
        lbl_location.grid(row=1, column=0, pady=5, padx=15, sticky="w")

        txt_location = Entry(Map_Frame, font=(
            "times new roman", 15, "bold"), bd=5, relief=GROOVE)
        txt_location.grid(row=1, column=1, pady=0, padx=20, sticky="w")

        Searchbtn = Button(Map_Frame, text="Search", width=10).grid(
            row=1, column=3, padx=10, pady=0)


root = Tk()
obj = admin(root)
root.mainloop()
```

## Output:





| ID | Name | Mail | Password |
|---|---|---|---|
| 1001 | rafi | rafi@gmail.com | qazwsx |
| 1002 | sayma | sayma@gmail.com | qwer |
| 1003 | nabil | nabil@gmail.com | ujmikl |



## Summery:

Here we have many more limitations and have a lot of things to be improved. As we don’t able to show real time GPS in XAMPP. In future we have to connect it. Besides, we have to put a Tracker to the emergency vehicle to visible to our traffic police. Overall, we need a slight improve to be done hundred percent.