

Project No: 9**Student Details**

12106596 Manshi Kumari RK21STA24

12108408 Gaurav Kumar RK21STA25

12110095 Kul Deep Varma RK21STA35

Project Title: Design a Courier Management System for LPU using python.**Project Description:**

The minimum requirement of GUI as follows:

CMS

Login	New user	Track Consignment
-------	----------	----------------------

Register

Name	<input type="text"/>
Reg. No	<input type="text"/>
Gender	<input checked="" type="radio"/> Male <input type="radio"/> Female
Mobile no	<input type="text"/>
Email Id	<input type="text"/>
	<input type="button" value="Submit"/>

Login Page

User Name	<input type="text"/>
Pass Word	<input type="password"/>
	<input type="button" value="Login Now"/> <input type="button" value="New User"/>

**Track
Consignment**

Mobile No	<input type="text"/>
Consignment No	<input type="text"/>
	<input type="button" value="Track"/>

Rubrics:

- Pre-submission report - includes module wise description, role and responsibility of each group members and work plan as Gantt chart (5 Marks)
- Hard copy report and project submission- includes content relevance, methodology, Implementation (15 Marks)

- Peer evaluation (5 marks)
- Viva and presentation(5 Marks)
- Each group must submit one print out hard copy of report and one CD includes soft copy of report, running project folder.

Note: Above mentioned project description is minimum requirement, you can add new feature, new better look according to your requirement.

LPU COURIER FINDER





LOVELY
PROFESSIONAL
UNIVERSITY

PYTHON PROJECT
ON
LPU COURIER FINDER

Submitted by : Kul Deep Varma, Gaurav Kumar, Manshi Kumari

Registration no. : 12110095, 12108408, 12106596

Submitted To : Ajmer Singh

In partial fulfilment for the requirements of the award of the degree
of

“BACHELOR OF TECHNOLOGY”

“DEPARTMENT OF COMPUTER SCIENCE”

Lovely Professional University

Jalandhar (Punjab)

INTRODUCTION:

We have implemented a project named 'LPU Courier Finder' with a team of 3 members. In this application (project) you can join as a customer as well as a service provider. There are many service providers in LPU (UniMall) who provides the service to customers to easily find their product and get it. The service provider can add new arrived products by the provided functionalities and also can search for products in stock by the full of last digits of order id. If they want to deliver it then they can change the status of delivery in database as delivered and when they will search again then they will find it as delivered, also they can find how much products they have delivered and how much are left in stock. In another hand, you can join as a customer then you have to give the order id in input. If the product will found then you will get to know that your product is arrived in last destination and you will get order details also. If in any case order is not found then also you will get to know that your order is not at the last destination (LPU-UniMall).

PROBLEM STATEMENT:

If you have ordered any product from Amazon, Flipkart, Myntra etc. and on the day of delivery you have to reach in Unimall or Main Gate of University and then being in line for a long time might hurts you or its difficult to do this thing.

So, we have developed an application throw which you can directed see that your product is reached to Unimall or not. So that you can fix a time and go to take it on that time without any frustration.

WORKING OF PROJECT:

In this project there are end users and service providers of

university. End users will be able to login via their details and after it then will see an interface of find courier where they have to fill Mo. No. and courier no. then they will get to know about their product whether that is there or not and if found then they can fix their time of picking. Service providers will be able to add courier details and will get time details of picking product by the customers or end users.

ADVANTAGES:

- Customers can get details of product at their place instead of visiting Unimall.
- Customers don't have to be in line for long time.
- Customers can book time slot for picking the product.

SYSTEM DESCRIPTION:

The system comprise of two major modules with their sub modules as follows:

1) Service Provider Module:

1. Login: Service provider can log in into the system using username and password.
2. Sign Up: A new service provider can sign up into the system by providing the details.
3. Add: Service provider can add the product details that is reached in unimall.
4. Search: service provider can search product by product id to remove after delivery.
5. There is an option provided to delivery of product.
6. Service Provider can go to home menu.
7. Log out: Service provider can log out from the system.

2) Customer Module:

1. Login: Customers can log in into the system using username and password.
2. Sign Up: A new customer can sign up into the system by providing the details.
3. Find Order: Customer can find their order is reached to unimall or not by providing product id and contact no.
4. Home: Customer can go back to home menu.
5. Log out: Customer can log out from the system.

DESIGN:

Design of this project is too simple if we consider the algorithms because in this project we haven't used any algorithm as well as not any data structure. We just send the input data and fetched data from one python file to another file. Coming to database, we have used only two tables for storing login and sign in data and another for product details.

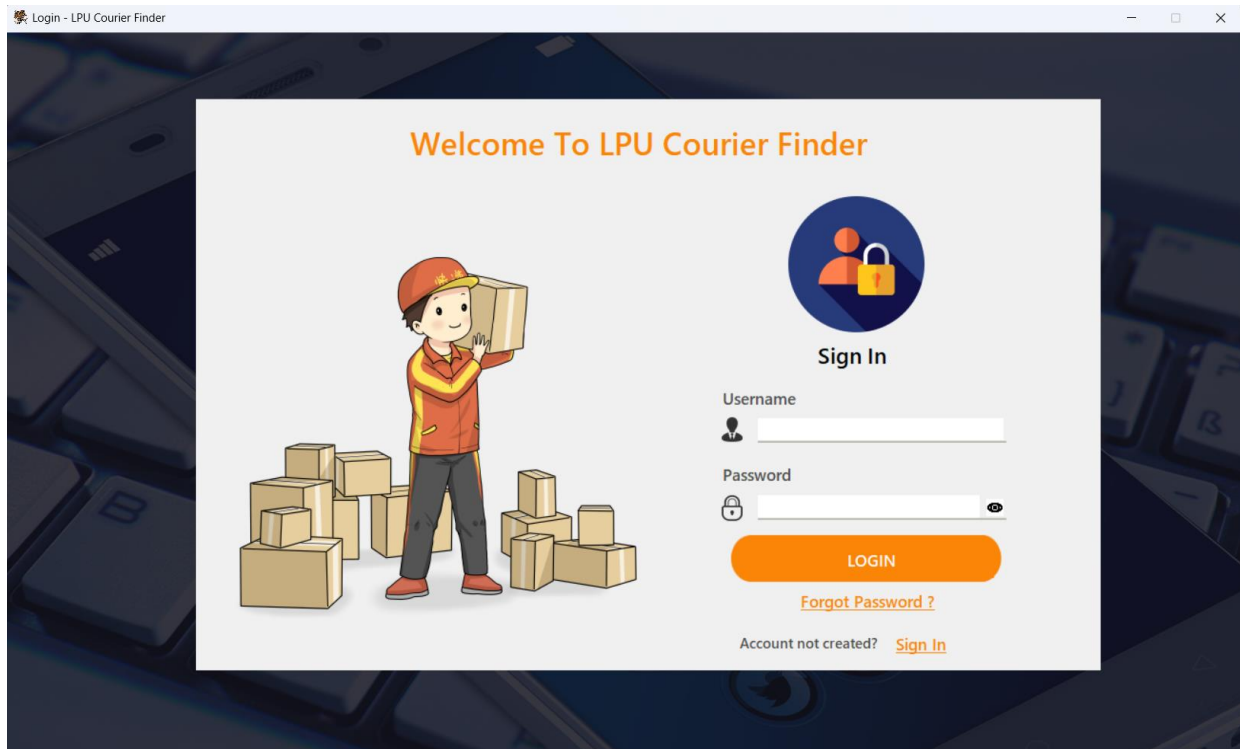
1. Tables used:

```
mysql> desc curiour_details;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| fullName   | varchar(50)   | NO   |     | NULL    |       |
| userName   | varchar(25)   | NO   | PRI | NULL    |       |
| password    | varchar(25)   | NO   |     | NULL    |       |
| regNo      | int           | NO   | UNI | NULL    |       |
| phoneNo    | varchar(10)   | NO   |     | NULL    |       |
| type       | char(1)       | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.02 sec)

mysql> desc courier_data;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name       | varchar(30)   | YES  |     | NULL    |       |
| order_id   | varchar(20)   | YES  |     | NULL    |       |
| address    | varchar(50)   | YES  |     | NULL    |       |
| phone_no   | varchar(15)   | YES  |     | NULL    |       |
| payment    | varchar(10)   | YES  |     | NULL    |       |
| status     | varchar(20)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```


Result Screenshots:

Log In Page:



Sign In Page

Sign In - LPU Courier Finder

Sign In To LPU Courier Finder

Full Name Registration No.

Username Phone Number

Password ☒ Customer ☐ Service Provider

[SIGN UP](#)

Already have an account? [Log In](#)

Find Product as customer

Login - LPU Courier Finder

Enter Your Details Here

Order Details


Order Id

Mobile No.

[Find](#)

Find Product as Service Provider

Dashboard - LPU Courier Finder



Kul Deep Varma

Add Products
Search Products

Log out

Dashboard

Search Here

Search

Name : kul deep varma

Status : Delivered

Order Id : 1234567890

Address : 377c/18, street-5, gurunakpura west, jalandhar

Mobile No. : 7413912366

Payment : Done

☐ Delivered

Submit

Details About Stock


Refresh

Total items left in stock : 0

Total items arrived in Branch : 1

Add product in stock (database)

Dashboard - LPU Courier Finder



Kul Deep Varma

Add Products
Search Products

Log out

Dashboard

Name

Order Id

Address

Mo. No.

Payment Status

Delivery Status

☒ Pending
☐ Done

☒ In Stock
☐ Delivered

Add

Details About Stock

Refresh

Total items left in stock : 0

Total items arrived in Branch : 1

SYSTEM REQUIREMENT:

❖ Hardware requirement:

- Processor – core i3
- Hard Disk – 160GB
- Memory – 1GB
- Monitor

❖ Software requirement:

- Windows 7 or higher
- Python
- MySQL Database

LIMITATIONS AND DISADVANTAGES:

- Wrong input can affect project output
- Can consume time if system crashes down.
- Internet connectivity is mandatory.

APPLICATIONS:

- Customers can find there product's information whether product to unimall or not and the can book a time slot of picking as well.
- Service provider can add details of new arrived products and can remove them after the product delivery.

REFERENCES:

1. <https://en.wikipedia.org/>
2. <https://www.google.com/>
3. <https://www.geeksforgeeks.org/>

TEAM:

Gaurav Kumar (12108408):

Work:

1. Order_Not_found page
2. Log in page

Manshi Kumari (12106596):

Work:

1. Order Found page
2. Sign page

Kul Deep Varma (12110095):

Work:

1. Service provider home page,
2. Add products page
3. Linked database

CODE:

Start.py

```
from tkinter import *  
import LoginPage as loggingapp  
def page():  
    window = Tk()  
    loggingapp.Login_Page(window)  
    window.mainloop()  
page()
```

LoginPage.py

```

from tkinter import *

from tkinter import Tk

from tkinter import messagebox

from PIL import ImageTk, Image

import main_page as home

import signin

import service_provider

import mysql.connector


class Login_Page:

    def loginDb(self):

self.db=mysql.connector.connect(host="localhost",user='root',password='Kuldeep@7413',db='college')

    def __init__(self, window):

        self.window = window

        self.window.geometry('1300x760')

        self.window.resizable(0, 0)

        #self.window.state('zoomed')

        self.window.title('Login - LPU Courier Finder')


        #

=====

=====

=====


        #

=====fevi

```

```

con=====
=====

#
=====
=====
=====

self.photoicon = PhotoImage(file =
"images\\delivery-man.png")

self.window.iconphoto(False,
self.photoicon)

#
=====
=====
=====

#
=====background
kground
image=====
==

#
=====
=====
=====

self.bg_frame =
Image.open('images\\background1.png')

photo =
ImageTk.PhotoImage(self.bg_frame)

self.bg_panel = Label(self.window,
image=photo)

self.bg_panel.image = photo

self.bg_panel.pack(fill='both',
expand='yes')

# ===== Login Frame
=====

```



```
self.lgn_frame = Frame(self.window,  
bg='#FFFFFF', width=950, height=600)
```

```
self.lgn_frame.place(x=200, y=70)
```

```
#
```

```
=====
```

```
#
```

```
=====
```

```
#
```

```
=====
```

```
self.txt = "Welcome To LPU Courier  
Finder"
```

```
self.heading = Label(self.lgn_frame,  
text=self.txt, font=('yu gothic ui', 25,  
"bold"), bg="#FFFFFF", fg='#fa8507',  
bd=5, relief=FLAT)
```

```
self.heading.place(x=165, y=30,  
width=600, height=30)
```

```
#
```

```
=====
```

```
# ===== Left Side Image
```

```
=====
```

```
#
```

```
=====
```

```
self.side_image =  
Image.open('images\\vector1.png')  
  
photo =  
ImageTk.PhotoImage(self.side_image)  
  
self.side_image_label =  
Label(self.lgn_frame, image=photo,  
bg='#FFFFFF')  
  
self.side_image_label.image = photo  
  
self.side_image_label.place(x=5,  
y=100)
```

```
#
```

```
=====
```

```
# ===== Sign In Image
```

```
=====
```

```
#
```

```
=====
```

```
self.sign_in_image =  
Image.open('images\\hyy.png')  
  
photo =  
ImageTk.PhotoImage(self.sign_in_image)  
  
self.sign_in_image_label =  
Label(self.lgn_frame, image=photo,  
bg='#FFFFFF')  
  
self.sign_in_image_label.image =  
photo  
  
self.sign_in_image_label.place(x=620,  
y=100)
```

#

```
=====
=====
=====
```

===== Sign In label

```
=====
=====
```

#

```
=====
=====
=====
```

```
self.sign_in_label =
Label(self.lgn_frame, text="Sign In",
bg="#FFFFFF", fg="white",

font=("yu gothic ui",
17, "bold"))
```

```
self.sign_in_label.place(x=650,
y=250)
```

#

```
=====
=====
=====
```

#

```
=====user
name=====
=====
```

#

```
=====
=====
=====
```

```
self.username_label =
Label(self.lgn_frame, text="Username",
bg="#FFFFFF", fg="#4f4e4d",

font=("yu gothic ui",
13, "bold"))
```

```
self.username_label.place(x=550,  
y=300)
```

```
self.username_entry =  
Entry(self.lgn_frame, highlightthickness=0,  
relief=FLAT, bg="#FFFFFF",  
fg="#6b6a69",
```

```
font=("yu gothic ui  
", 12, "bold"))
```

```
self.username_entry.place(x=590,  
y=335, width=258)
```

```
self.username_line =  
Canvas(self.lgn_frame, width=261,  
height=2.0, bg="#bdb9b1",  
highlightthickness=0)
```

```
self.username_line.place(x=590,  
y=359)
```

```
# ===== Username icon =====
```

```
self.username_icon =  
Image.open('images\\username_icon.png')
```

```
photo =  
ImageTk.PhotoImage(self.username_icon)
```

```
self.username_icon_label =  
Label(self.lgn_frame, image=photo,  
bg='#FFFFFF')
```

```
self.username_icon_label.image =  
photo
```

```
self.username_icon_label.place(x=550,  
y=332)
```

```
#
```

```
=====
=====
=====
```

#

```
=====pass
word=====
=====
```

#

```
=====
=====
=====
```

```
self.password_label =
Label(self.lgn_frame, text="Password",
bg="#FFFFFF", fg="#4f4e4d",

font=("yu gothic ui",
13, "bold"))
```

```
self.password_label.place(x=550,
y=380)
```

```
self.password_entry =
Entry(self.lgn_frame, highlightthickness=0,
relief=FLAT, bg="#FFFFFF",
fg="#6b6a69",
```

```
font=("yu gothic ui",
12, "bold"), show="*")
```

```
self.password_entry.place(x=590,
y=416, width=233)
```

```
self.password_line =
Canvas(self.lgn_frame, width=261,
height=2.0, bg="#bdb9b1",
highlightthickness=0)
```

```
self.password_line.place(x=590,
y=440)
```

```

# ===== Password icon
=====

self.password_icon =
Image.open('images\\password_icon.png')

photo =
ImageTk.PhotoImage(self.password_icon)

self.password_icon_label =
Label(self.lgn_frame, image=photo,
bg='#FFFFFF')

self.password_icon_label.image =
photo

self.password_icon_label.place(x=550,
y=414)

# ===== show/hide password
=====
=====
=====

self.show_image =
ImageTk.PhotoImage \

(file='images\\show.png')

self.hide_image =
ImageTk.PhotoImage \

(file='images\\hide.png')

self.show_button =
Button(self.lgn_frame,
image=self.show_image,
command=self.show, relief=FLAT,

activebackground="white"

, borderwidth=0,

```

```
background="white", cursor="hand2")

self.show_button.place(x=830, y=420)
```

```
#
```

```
=====
=====
=====
```

```
#
```

```
=====logi
n
button=====
=====
```

```
#
```

```
=====
=====
=====
```

```
self.lgn_button =
Image.open('images\\btn1.png')

photo =
ImageTk.PhotoImage(self.lgn_button)

self.lgn_button_label =
Label(self.lgn_frame, image=photo,
bg='#FFFFFF')

self.lgn_button_label.Image = photo

self.lgn_button_label.place(x=560,
y=455)

self.login =
Button(self.lgn_button_label,
text='LOGIN', font=("yu gothic ui", 13,
"bold"), width=25, bd=0, bg='#fa8507',
cursor='hand2',
activebackground='#fa8507', fg='white',
command=self.open_file)
```

```
self.login.pack()
```

```
self.login.place(x=20, y=10)
```

```
#
```

```
=====
=====
=====
```

```
#
```

```
=====For
got
password=====
=====
```

```
#
```

```
=====
=====
=====
```

```
self.forgot_button =
Button(self.lgn_frame, text="Forgot
Password ?",
```

```
font=("yu gothic ui",
13, "bold underline"), fg="#fa8507",
relief=FLAT,
```

```
activebackground="#FFFFFF"
```

```
, borderwidth=0,
background="#FFFFFF", cursor="hand2")
```

```
self.forgot_button.place(x=630,
y=510)
```

```
# ===== Sign Up
```

```
=====
=====
```

```
self.sign_label = Label(self.lgn_frame,
text='Account not created?', font=("yu
gothic ui", 11, "bold"),
```



```
        relief=FLAT,  
borderwidth=0, background="#FFFFFF",  
fg='#4f4e4d')
```

```
self.sign_label.place(x=570, y=560)
```

```
self.sign_button =  
Button(self.lgn_frame,font=("yu gothic ui",  
13, "bold underline"), text="Sign In",  
bg='#4f4e4d',fg="#fa8507",  
cursor="hand2", borderwidth=0,  
background="#FFFFFF",  
activebackground="#FFFFFF",  
command=self.signup )
```

```
self.sign_button.place(x=730, y=555)
```

```
self.loginDb()
```

```
def show(self):
```

```
self.hide_button =  
Button(self.lgn_frame,  
image=self.hide_image,  
command=self.hide, relief=FLAT,
```

```
activebackground="white"
```

```
        , borderwidth=0,  
background="white", cursor="hand2")
```

```
self.hide_button.place(x=830, y=420)
```

```
self.password_entry.config(show="")
```

```

def hide(self):

    self.show_button =
    Button(self.lgn_frame,
    image=self.show_image,
    command=self.show, relief=FLAT,

    activebackground="white"

    , borderwidth=0,
    background="white", cursor="hand2")

    self.show_button.place(x=830, y=420)

    self.password_entry.config(show='*')


def open_file(self):

    if self.username_entry.get() and
    self.password_entry.get():

username=self.username_entry.get()

        password=self.password_entry.get()

        #data current window

if(self.checking(self.window,username,pas
sword)):

        if(self.tpl[0][5]=='C'):

            home.details(self.window)

        elif(self.tpl[0][5]=='S'):

            for widgets in
self.lgn_frame.winfo_children():

                widgets.destroy()

service_provider.service(self.window)

```

else:

```
        messagebox.showerror('Login  
Error', 'Username or Password is wrong!')
```

else:

```
        messagebox.showerror('Login  
Error', 'Enter username and password')
```

```
def signup(self):
```

```
    signin.SignUp(self.window)
```

```
def loginUser(self,userName,password):
```

```
    try:
```

```
        queryy="select * from  
curiour_details
```

```
        where userName='{0}' and  
password='{1}';".format(userName,passwo  
rd)
```

```
        cursr=self.db.cursor()
```

```
        cursr.execute(queryy)
```

```
        self.tpl=cursr.fetchall()
```

```
        return True
```

```
    except:
```

```
        messagebox.ERROR('Database  
error','Failed to login the database.')
```

```
        return False
```

```
def
```

```
checking(self,window,username,password)  
:
```

```
    try:
```

```
        if
self.loginUser(username,password):

        return True

    else:

        return False

except :

    messagebox.ERROR('Database
error','Failed to login the database.')

    raise
```

Signin.py

```
from sqlite3 import DatabaseError

from tkinter import *

from tkinter import Tk

from tkinter import messagebox

from PIL import ImageTk, Image

import main_page as home

import LoginPage as logging

import mysql.connector

class SignUp:

    def __init__(self,win):

        self.loginDb()

        self.window = win
```

```

self.window.geometry('1300x760')

self.window.resizable(0, 0)

self.window.title('Sign In - LPU
Courier Finder')

#
=====fevi
con=====
=====

self.photoicon = PhotoImage(file =
"images\\delivery-man.png")

self.window.iconphoto(False,
self.photoicon)

#
=====bac
kground
image=====
==

self.bg_frame =
Image.open('images\\background1.png')

photo =
ImageTk.PhotoImage(self.bg_frame)

self.bg_panel = Label(self.window,
image=photo)

self.bg_panel.image = photo

self.bg_panel.pack(fill='both',
expand='yes')

#
=====
=== Login Frame
=====

self.lgn_frame = Frame(self.window,
bg='#FFFFFF', width=950, height=600)

```

```
self.lgn_frame.place(x=200, y=70)
```

```
#
```

```
=====L
```

```
PU COURIER
```

```
LABEL=====
```

```
txt = "Sign In To LPU Courier Finder"
```

```
heading = Label(self.lgn_frame,  
text=txt, font=('yu gothic ui', 25, "bold"),  
bg="#FFFFFF", fg='#fa8507', bd=5,  
relief=FLAT)
```

```
heading.place(x=165, y=50,  
width=600, height=50)
```

```
#
```

```
=====Full
```

```
Name=====
```

```
=====
```

```
name_label = Label(self.lgn_frame,  
text="Full Name", bg="#FFFFFF",  
fg="#4f4e4d", font=("yu gothic ui", 13,  
"bold"))
```

```
name_label.place(x=150, y=140)
```

```
self.name_entry =  
Entry(self.lgn_frame, highlightthickness=0,  
relief=FLAT, bg="#FFFFFF",  
fg="#6b6a69",
```

```
font=("yu gothic ui",  
12, "bold"))
```

```
self.name_entry.place(x=190, y=175,  
width=258)
```

```
name_line = Canvas(self.lgn_frame,
width=261, height=2.0, bg="#bdb9b1",
highlightthickness=0)
```

```
name_line.place(x=190, y=199)
```

```
#
```

```
=====
Username icon
=====
```

```
name_icon =
Image.open('images\\username_icon.png')
```

```
photo =
ImageTk.PhotoImage(name_icon)
```

```
name_icon_label =
Label(self.lgn_frame, image=photo,
bg='#FFFFFF')
```

```
name_icon_label.image = photo
```

```
name_icon_label.place(x=150, y=172)
```

```
#
```

```
=====Reg
istration
No.=====
=
```

```
reg_label = Label(self.lgn_frame,
text="Registration No.", bg="#FFFFFF",
fg="#4f4e4d", font=("yu gothic ui", 13,
"bold"))
```

```
reg_label.place(x=500, y=140)
```

```
self.reg_entry = Entry(self.lgn_frame,
highlightthickness=0, relief=FLAT,
bg="#FFFFFF", fg="#6b6a69",
```

```
font=("yu gothic ui",  
12, "bold"))
```

```
self.reg_entry.place(x=540, y=175,  
width=258)
```

```
reg_line = Canvas(self.lgn_frame,  
width=261, height=2.0, bg="#bdb9b1",  
highlightthickness=0)
```

```
reg_line.place(x=540, y=199)
```

```
#
```

```
=====Username  
icon
```

```
=====
```

```
reg_icon =  
Image.open('images\\username_icon.png')
```

```
photo =  
ImageTk.PhotoImage(reg_icon)
```

```
reg_icon_label =  
Label(self.lgn_frame, image=photo,  
bg='#FFFFFF')
```

```
reg_icon_label.image = photo
```

```
reg_icon_label.place(x=500, y=172)
```

```
#
```

```
=====Typ  
e of  
user=====
```

```
=====
```

```
self.var1=IntVar()
```

```
self.var2=IntVar()
```

```
self.t1 = Checkbutton(win,  
text="Customer", variable=self.var1,
```



```
onvalue=1, offvalue=0,  
command=self.discard2)
```

```
self.t1.pack()
```

```
self.t1.place(x=700,y=400)
```

```
self.t1.select()
```

```
self.t2 = Checkbutton(win,  
text="Service Provider",  
variable=self.var2, onvalue=1, offvalue=0,  
command=self.discard1)
```

```
self.t2.pack()
```

```
self.t2.place(x=800, y=400)
```

```
#
```

```
=====
```

```
==Username=====
```

```
=====
```

```
username_label =  
Label(self.lgn_frame, text="Username",  
bg="#FFFFFF", fg="#4f4e4d",font=("yu  
gothic ui", 13, "bold"))
```

```
username_label.place(x=150, y=220)
```

```
self.username_entry =  
Entry(self.lgn_frame, highlightthickness=0,  
relief=FLAT, bg="#FFFFFF",  
fg="#6b6a69",
```

```
font=("yu gothic ui",  
12, "bold"))
```

```
self.username_entry.place(x=190,  
y=255, width=258)
```

```
username_line =
```

```
Canvas(self.lgn_frame, width=261,  
height=2.0, bg="#bdb9b1",  
highlightthickness=0)
```

```
username_line.place(x=190, y=279)
```

```
# =====
```

```
Username icon
```

```
=====
```

```
username_icon =  
Image.open('images\\username_icon.png')
```

```
photo =  
ImageTk.PhotoImage(username_icon)
```

```
username_icon_label =  
Label(self.lgn_frame, image=photo,  
bg='#FFFFFF')
```

```
username_icon_label.image = photo
```

```
username_icon_label.place(x=150,  
y=252)
```

```
#
```

```
=====Mo  
bile  
no=====
```

```
mo_label = Label(self.lgn_frame,  
text="Phone Number", bg="#FFFFFF",  
fg="#4f4e4d", font=("yu gothic ui", 13,  
"bold"))
```

```
mo_label.place(x=500, y=210)
```

```
self.mo_entry = Entry(self.lgn_frame,  
highlightthickness=0, relief=FLAT,  
bg="#FFFFFF", fg="#6b6a69",
```

```
font=("yu gothic ui",
```

12, "bold"))

self.mo_entry.place(x=540, y=258,
width=258)

mo_line = Canvas(self.lgn_frame,
width=261, height=2.0, bg="#bdb9b1",
highlightthickness=0)

mo_line.place(x=540, y=278)

#

=====
== Username icon
=====

mo_icon =
Image.open('images\\username_icon.png')

photo =
ImageTk.PhotoImage(mo_icon)

mo_icon_label =
Label(self.lgn_frame, image=photo,
bg='#FFFFFF')

mo_icon_label.image = photo

mo_icon_label.place(x=500, y=250)

#

=====
===== password
=====

self.password_label =
Label(self.lgn_frame, text="Password",
bg="#FFFFFF", fg="#4f4e4d",

font=("yu gothic ui",
13, "bold"))

self.password_label.place(x=150,

y=300)

```
self.password_entry =  
Entry(self.lgn_frame, highlightthickness=0,  
relief=FLAT, bg="#FFFFFF",  
fg="#6b6a69",
```

```
font=("yu gothic ui",  
12, "bold"), show="*")
```

```
self.password_entry.place(x=190,  
y=336, width=233)
```

```
self.password_line =  
Canvas(self.lgn_frame, width=261,  
height=2.0, bg="#bdb9b1",  
highlightthickness=0)
```

```
self.password_line.place(x=190,  
y=360)
```

```
#
```

```
=====
```

```
== Password icon
```

```
=====
```

```
===
```

```
self.password_icon =  
Image.open('images\\password_icon.png')
```

```
photo =  
ImageTk.PhotoImage(self.password_icon)
```

```
self.password_icon_label =  
Label(self.lgn_frame, image=photo,  
bg='#FFFFFF')
```

```
self.password_icon_label.image =  
photo
```

```
self.password_icon_label.place(x=150,  
y=334)
```

```
#
```

```
=====
== show/hide password
=====
```

```
self.show_image =
ImageTk.PhotoImage \

    (file='images\\show.png')
```

```
self.hide_image =
ImageTk.PhotoImage \

    (file='images\\hide.png')
```

```
self.show_button =
Button(self.lgn_frame,
image=self.show_image,
command=self.show, relief=FLAT,
```

```
activebackground="white"
```

```
        , borderwidth=0,
background="white", cursor="hand2")
```

```
self.show_button.place(x=430, y=340)
```

```
#
```

```
=====fevi
con=====
=====
```

```
photoicon = PhotoImage(file =
"images\\delivery-man.png")
```

```
self.window.iconphoto(False,
photoicon)
```

```
# ===== show/hide
password
=====
=====
```

```
self.show_image =
ImageTk.PhotoImage \

    (file='images\\show.png')
```

```
self.hide_image =
ImageTk.PhotoImage \

    (file='images\\hide.png')
```

```
self.show_button =
Button(self.lgn_frame,
image=self.show_image,
command=self.show, relief=FLAT,
```

```
activebackground="white"
```

```
        , borderwidth=0,
background="white", cursor="hand2")
```

```
self.show_button.place(x=430, y=340)
```

```
#
=====sign
up
button=====
=====
```

```
sign_button =
Image.open('images\\btn1.png')
```

```
photo =
ImageTk.PhotoImage(sign_button)
```

```
sign_button_label =  
Label(self.lgn_frame, image=photo,  
bg='#FFFFFF')
```

```
sign_button_label.Image = photo
```

```
sign_button_label.place(x=330,  
y=405)
```

```
signup = Button(sign_button_label,  
text='SIGN UP', font=("yu gothic ui", 13,  
"bold"), width=25, bd=0, bg='#fa8507',  
cursor='hand2',  
activebackground='#fa8507', fg='white',  
command=self.open_file)
```

```
signup.pack()
```

```
signup.place(x=20, y=10)
```

```
# =====  
log in  
=====
```

```
sign_label = Label(self.lgn_frame,  
text='Already have an account?', font=("yu  
gothic ui", 11, "bold"), relief=FLAT,  
borderwidth=0, background="#FFFFFF",  
fg='#4f4e4d')
```

```
sign_label.place(x=360, y=530)
```

```
sign_button =  
Button(self.lgn_frame, font=("yu gothic ui",  
13, "bold underline"), text="Log In",  
bg='#4f4e4d', fg='#fa8507',  
cursor="hand2", borderwidth=0,  
background="#FFFFFF",  
activebackground="#FFFFFF",  
command=self.openloginpage)
```

```
sign_button.place(x=540, y=525)
```

```
def show(self):
```

```
    hide_button = Button(self.lgn_frame,  
image=self.hide_image,  
command=self.hide, relief=FLAT,  
activebackground="white" ,  
borderwidth=0, background="white",  
cursor="hand2")
```

```
    hide_button.place(x=430, y=340)
```

```
    self.password_entry.config(show="")
```

```
def hide(self):
```

```
    show_button = Button(self.lgn_frame,  
image=self.show_image,  
command=self.show, relief=FLAT,  
activebackground="white" ,  
borderwidth=0, background="white",  
cursor="hand2")
```

```
    show_button.place(x=430, y=340)
```

```
    self.password_entry.config(show='*')
```

```
def discheck1(self):
```

```
    self.t1.deselect()
```

```
def discheck2(self):
```

```
    self.t2.deselect()
```

```
def openloginpage(self):
```

```
    logging.Login_Page(self.window)
```

```
def open_file(self):
```



```
#data current window

try:

username=self.username_entry.get()

    password=self.password_entry.get()

    fullname=self.name_entry.get()

    Mobile_no=self.mo_entry.get()

    reg_no=int(self.reg_entry.get())

    if self.var2.get():

        typeofuser='S'

    else:

        typeofuser='C'

self.insertRecord(fullname,username,password,reg_no,Mobile_no,typeofuser)

    home.details(self.window)

except DatabaseError:

    messagebox.showerror('record error','User already exist / database error!')

except:

    messagebox.showerror('record error','Incomplete details.')

#database

def loginDb(self):

self.db=mysql.connector.connect(host="localhost",user='root',password='Kuldeep@7413',db='college')
```

```

def
insertRecord(self,fullName,userName,pass
word,regNo,phoneNo,type):

    createTablee=""

    create table if not exists
curiour_details(

    fullName varchar(50) not
null,userName varchar(25) primary
key,password varchar(25) not null,regNo
int unique not null,phoneNo varchar(10)
not null,type char(1) not null

);

""

self.db.cursor().execute(createTablee)

queryInsert=""

insert into curiour_details
values('{0}','{1}','{2}','{3}','{4}','{5}')".for
mat(fullName,userName,password,regNo,p
honeNo,type)

try:

self.db.cursor().execute(queryInsert)

    self.db.commit()

except:

    raise DatabaseError('User already
exist or database error!')

```

main_page.py

```
from tkinter import *
from tkinter import Tk
from PIL import ImageTk, Image
import mysql.connector
import courier_details as found
import order_not_found as not_found
```

```
class details:
```

```
    def __init__(self, window):
```

```
        self.window = window
```

```
        #self.window.state('zoomed')
```

```
        # self.window.title('LPU Courier
Finder')
```

```
        #
```

```
=====
=====
=====
```

```
        #
```

```
=====fevi
con=====
=====
```

```
        #
```

```
=====
=====
=====
```

```
        self.photoicon = PhotoImage(file =
"images\\delivery-man.png")
```

```
self.window.iconphoto(False,  
self.photoicon)
```

```
#
```

```
=====
```

```
#
```

```
=====background  
kground  
image=====
```

```
#
```

```
=====
```

```
self.bg_frame =  
Image.open('images\\background1.png')
```

```
photo =  
ImageTk.PhotoImage(self.bg_frame)
```

```
self.bg_panel = Label(self.window,  
image=photo)
```

```
self.bg_panel.image = photo
```

```
self.bg_panel.pack(fill='both',  
expand='yes')
```

```
# ===== Login Frame
```

```
=====
```

```
self.lgn_frame = Frame(self.window,  
bg='FFFFFF', width=950, height=600)
```

```
self.lgn_frame.place(x=200, y=70)
```

```
#
```

```
=====
```

#

=====

=====

#

=====

=====

=====

self.txt = "Enter Your Details Here"

self.heading = Label(self.lgn_frame,
text=self.txt, font=('yu gothic ui', 25,
"bold"), bg="#FFFFFF", fg='#fa8507',
bd=5, relief=FLAT)

self.heading.place(x=165, y=80,
width=600, height=30)

#

=====

=====

=====

=====Order details label

=====

=====

#

=====

=====

=====

self.order_details =
Label(self.lgn_frame, text="Order Details",
bg="#FFFFFF", fg="#fa8507",

font=("yu gothic ui",
17, "bold"))

self.order_details.place(x=400,
y=150)

#

```
=====
=====
=====
```

#

```
=====Ord
er
Id=====
=====
```

#

```
=====
=====
=====
```

```
self.order_id = Label(self.lgn_frame,
text="Order Id", bg="#FFFFFF",
fg="#4f4e4d",
```

```
font=("yu gothic ui",
13, "bold"))
```

```
self.order_id.place(x=300, y=280)
```

```
self.order_entry =
Entry(self.lgn_frame, highlightthickness=0,
relief=FLAT, bg="#FFFFFF",
fg="#6b6a69",
```

```
font=("yu gothic ui", 12, "bold"))
```

```
self.order_entry.place(x=375, y=280,
width=258)
```

```
self.order_entry.focus_set()
```

```
self.order_line =
Canvas(self.lgn_frame, width=261,
height=2.0, bg="#bdb9b1",
```

```
highlightthickness=0)
```

```
self.order_line.place(x=375, y=302)
```

```
#
```

```
=====
=====
=====
```

```
#
```

```
=====login
n
button=====
=====
```

```
#
```

```
=====
=====
=====
```

```
self.find_btn =
Image.open('images\\btn1.png')
```

```
photo =
ImageTk.PhotoImage(self.find_btn)
```

```
self.find_button =
Label(self.lgn_frame, image=photo,
bg='#FFFFFF')
```

```
self.find_button.Image = photo
```

```
self.find_button.place(x=330, y=415)
```

```
self.login = Button(self.find_button,
text='Find', font=("yu gothic ui", 13,
"bold"), width=25, bd=0, bg='#fa8507',
cursor='hand2',
activebackground='#fa8507', fg='white',
command=self.open_file)
```

```
self.login.pack()
```

```
self.login.place(x=20, y=10)
```

#

```
=====
=====
=====
```

#

```
=====pass
word=====
=====
```

#

```
=====
=====
=====
```

```
self.mo_label = Label(self.lgn_frame,
text="Mobile No.", bg="#FFFFFF",
fg="#4f4e4d",
```

```
font=("yu gothic ui",
13, "bold"))
```

```
self.mo_label.place(x=300, y=330)
```

```
self.mo_entry = Entry(self.lgn_frame,
highlightthickness=0, relief=FLAT,
bg="#FFFFFF", fg="#6b6a69",
```

```
font=("yu gothic ui",
12, "bold"))
```

```
self.mo_entry.place(x=390, y=330,
width=233)
```

```
self.mo_line = Canvas(self.lgn_frame,
width=245, height=2.0, bg="#bdb9b1",
highlightthickness=0)
```

```
self.mo_line.place(x=390, y=352)
```

```
def loginDb(self):
```



```
self.db=mysql.connector.connect(host="localhost",user='root',password='Kuldeep@7413',db='college')
```

```
def open_file(self):
```

```
    self.loginDb()
```

```
    #take values
```

```
    order_id=self.order_entry.get()
```

```
    mo_no=self.mo_entry.get()
```

```
    try:
```

```
        query="select * from courier_data  
where order_id={0} and  
phone_no={1}";".format(order_id,mo_no)
```

```
        cursr=self.db.cursor()
```

```
        cursr.execute(query)
```

```
        tpl=cursr.fetchall()
```

```
found.courier_details(self.window,tpl)
```

```
    except:
```

```
        #messagebox.showinfo('Order not  
found','Order not found!\nTry again after  
sometime.')
```

```
not_found.courier_details(self.window)
```

Courier_details.py

```

from tkinter import *
from tkinter import Tk
from PIL import ImageTk, Image
import main_page

class courier_details:

    def __init__(self, window,data):

        self.window = window

        self.window.geometry('1300x760')

        self.window.resizable(0, 0)

        #self.window.state('zoomed')

        self.window.title('LPU Courier
Finder')

        #
=====
=====
=====

        #
=====fevi
con=====
=====

        #
=====
=====
=====

        self.photoicon = PhotoImage(file =
"images\\delivery-man.png")

```

```
self.window.iconphoto(False,  
self.photoicon)
```

```
#
```

```
=====
```

```
#
```

```
=====background  
kground  
image=====
```

```
#
```

```
=====
```

```
self.bg_frame =  
Image.open('images\\background1.png')
```

```
photo =  
ImageTk.PhotoImage(self.bg_frame)
```

```
self.bg_panel = Label(self.window,  
image=photo)
```

```
self.bg_panel.image = photo
```

```
self.bg_panel.pack(fill='both',  
expand='yes')
```

```
# ===== Login Frame
```

```
=====
```

```
self.lgn_frame = Frame(self.window,  
bg='FFFFFF', width=950, height=600)
```

```
self.lgn_frame.place(x=200, y=70)
```

```
#
```

```
=====
```

#

=====
=====

#

=====
=====

self.txt = "Courier Details"

self.heading = Label(self.lgn_frame,
text=self.txt, font=('yu gothic ui', 25,
"bold"), bg="#FFFFFF", fg='#fa8507',
bd=5, relief=FLAT)

self.heading.place(x=165, y=80,
width=600, height=30)

#

=====
=====

=====Order details label

=====
=====

#

=====
=====

self.order_details =
Label(self.lgn_frame, text="Product
Found", bg="#FFFFFF", fg="#fa8507",
font=("yu gothic ui",
17, "bold"))

self.order_details.place(x=390,
y=150)

```
text1='Name : '+data[0][0]+'\\nContact  
no. : '+data[0][3]+'\\nAddress : '+data[0][1]
```

```
self.print_details =  
Label(self.lgn_frame, text=text1,  
bg="#FFFFFF", fg="#fa8507",  
  
font=("yu gothic ui",  
17, "bold"))  
  
self.print_details.place(x=350, y=150)
```

```
self.find_btn =  
Image.open('images\\btn1.png')  
  
photo =  
ImageTk.PhotoImage(self.find_btn)
```

```
self.login = Button(self.window,  
text='Print', font=("yu gothic ui", 13,  
"bold"), bd=0, bg='#fa8507',  
cursor='hand2',  
activebackground='#fa8507', fg='white',  
command=self.open_file)
```

```
self.login.pack()
```

```
self.login.place(x=500, y=500,  
width=100)
```

```
self.back = Button(self.window,  
text='Home', font=("yu gothic ui", 13,  
"bold"), bd=0, bg='#fa8507',  
cursor='hand2',  
activebackground='#fa8507', fg='white',  
command=self.backing)
```

```
self.back.pack()

self.back.place(x=700, y=500,
width=100)

def open_file(self):

    pass

def backing(self):

    main_page.details(self.window)
```

Order_not_found.py

```
from tkinter import *
from tkinter import Tk
from PIL import ImageTk, Image
import main_page

class courier_details:

    def __init__(self, window):

        self.window = window

        self.window.geometry('1300x760')

        self.window.resizable(0, 0)

        #self.window.state('zoomed')

        self.window.title('LPU Courier
Finder')
```

#

```
=====
=====
=====
```

#

```
=====fevi
con=====
=====
```

#

```
=====
=====
=====
```

```
self.photoicon = PhotoImage(file =
"images\\delivery-man.png")
```

```
self.window.iconphoto(False,
self.photoicon)
```

#

```
=====
=====
=====
```

#

```
=====bac
kground
image=====
==
```

#

```
=====
=====
=====
```

```
self.bg_frame =
Image.open('images\\background1.png')
```

```
photo =
ImageTk.PhotoImage(self.bg_frame)
```

```
self.bg_panel = Label(self.window,
image=photo)
```

```
self.bg_panel.image = photo
```

```
self.bg_panel.pack(fill='both',  
expand='yes')
```

```
# ===== Login Frame
```

```
=====
```

```
self.lgn_frame = Frame(self.window,  
bg='#FFFFFF', width=950, height=600)
```

```
self.lgn_frame.place(x=200, y=70)
```

```
#
```

```
=====
```

```
=====
```

```
=====
```

```
#
```

```
=====
```

```
=====
```

```
#
```

```
=====
```

```
=====
```

```
=====
```

```
self.txt = "Courier Details"
```

```
self.heading = Label(self.lgn_frame,  
text=self.txt, font=('yu gothic ui', 25,  
"bold"), bg="#FFFFFF", fg='#fa8507',  
bd=5, relief=FLAT)
```

```
self.heading.place(x=165, y=80,  
width=600, height=30)
```

```
#
```

```
=====
```

```
=====
```

```
=====
```

```
# =====sad
```

```
=====
```



```
=====
#
=====
=====
=====
```

```
self.find_btn =
Image.open('images\\sad.png')

photo =
ImageTk.PhotoImage(self.find_btn)

self.find_button =
Label(self.lgn_frame, image=photo,
bg='#FFFFFF')

self.find_button.Image = photo

self.find_button.place(x=370,
y=160,width=200)
```

```
#
=====
=====
=====
```

```
# =====Order details label
=====
=====
```

```
#
=====
=====
=====
```

```
self.order_details =
Label(self.lgn_frame, text="Product Not
Found", bg="#FFFFFF", fg="red",

font=("yu gothic ui",
17, "bold"))

self.order_details.place(x=360,
```

y=330)

```
self.order_details =  
Label(self.lgn_frame, text="Wait for some  
days. May be your order is on the way to  
our branch.\nThank you for using LPU  
Courier Finder.\nStay safe and Healthy",  
bg="#FFFFFF", fg="#fa8507",  
  
font=("yu gothic ui",  
13, "bold"))
```

```
self.order_details.place(x=200,  
y=390)
```

```
self.find_btn =  
Image.open('images\\btn1.png')
```

```
photo =  
ImageTk.PhotoImage(self.find_btn)
```

```
self.find_button =  
Label(self.lgn_frame, image=photo,  
bg='FFFFFF')
```

```
self.find_button.Image = photo
```

```
self.find_button.place(x=370,  
y=475,width=200)
```

```
self.login = Button(self.find_button,  
text='Home', font=("yu gothic ui", 13,  
"bold"), width=25, bd=0, bg='#fa8507',  
cursor='hand2',  
activebackground='#fa8507', fg='white',  
command=self.open_file)
```

```
self.login.pack()
```

```
self.login.place(x=-28, y=10)
```

```
def open_file(self):  
    main_page.details(self.window)
```

Service_provider.py

```
from asyncio.windows_events import  
NULL  
  
from distutils.cmd import Command  
  
from sqlite3 import DatabaseError  
  
from tkinter import *  
  
from tkinter import messagebox  
  
from turtle import color, left  
  
from PIL import ImageTk, Image  
  
import mysql.connector  
  
import LoginPage  
  
  
class service:  
    def __init__(self,window):  
        self.window = window  
  
        self.window.geometry('1300x760')  
  
        self.window.resizable(0, 0)  
  
        #self.window.state('zoomed')  
  
        self.window.title('Dashboard - LPU  
Courier Finder')
```

```
#===== Login Database
```

```
=====
```

```
self.db=mysql.connector.connect(host="localhost",user='root',password='Kuldeep@7413',db='college')
```

```
#
```

```
=====
```

```
=====
```

```
=====
```

```
#
```

```
=====fevi
```

```
con=====
```

```
=====
```

```
#
```

```
=====
```

```
=====
```

```
=====
```

```
self.photoicon = PhotoImage(file =  
"images\\delivery-man.png")
```

```
self.window.iconphoto(False,  
self.photoicon)
```

```
#
```

```
=====
```

```
=====
```

```
=====
```

```
#
```

```
=====
```

```
Header
```

```
=====
```

```
=====
```

```
#
```

```
=====
```

```
=====
```

=====

```
self.header=Frame(self.window,bg='#1c355e')
```

```
    self.header.place(x=300, y=0, width=1000, height=60)
```

```
self.logout_text=Button(self.header,text='Logout', bg='#fa9c05', font=("",13,"bold"),bd=0,fg='white', cursor='hand2', activebackground='#32cf8e', command=self.Logout)
```

```
self.logout_text.place(x=780,y=15,width=150)
```

#

=====

#

=====

Sidebar

=====

=====

#

=====

=====

=====

```
self.sidebar=Frame(self.window,bg='ffffff')
```

```
self.sidebar.place(x=0,y=0,width=300,height=750)
```

```
#===== logo
```

```
=====
```

```
self.logoImage=Image.open('images\\hyy.png')
```

```
photo=ImageTk.PhotoImage(self.logoImage)
```

```
self.logo=Label(self.sidebar, image=photo, bg='#ffffff')
```

```
self.logo.image=photo;
```

```
self.logo.place(x=70,y=80)
```

```
#===== User Details
```

```
=====
```

```
self.heading=Label(self.sidebar,text='Kul Deep Varma', font=("",15,"bold"), fg='black', bg='#eff5f6')
```

```
self.heading.place(x=70, y=250)
```

```
self.heading2=Button(self.sidebar,text='Add Products', bg='#fa9c05', font=("",13,"bold"),bd=0,fg='white', cursor='hand2', activebackground='#1c355e', command=self.adding)
```

```
self.heading2.place(x=0, y=400,
width=300, height=45)
```

```
self.heading2=Button(self.sidebar,text='Sea
rch Products', bg='#fa9c05',
font=("",13,"bold"),bd=0,fg='white',
cursor='hand2',
activebackground='#1c355e',
command=self.frame2)
```

```
self.heading2.place(x=0, y=450,
width=300, height=45)
```

```
#
```

```
=====
=====
=====
```

```
#
```

```
=====
Body
=====
=====
```

```
#
```

```
=====
=====
=====
```

```
#===== Frame 2
```

```
=====
```

```
self.bodyFrame2=Frame(self.window,bg='
#ffffff')
```

```
self.bodyFrame2.place(x=328,y=495,
width=940, height=220)
```

```
#===== details of stock
```

```
=====
```

```
self.refresh_button =  
Button(self.bodyFrame2,text='Refresh',  
bg='#fa9c05',  
font=("",13,"bold"),bd=0,fg='white',  
cursor='hand2', activebackground='white',  
command=self.total_data_info)
```

```
self.refresh_button.place(x=750,y=15,  
width=70)
```

```
self.total_info=Label(self.bodyFrame2,  
text='Details About Stock', fg='black',  
bg='#ffffff',font=("yu gothic ui ", 15,  
"bold"))
```

```
self.total_info.place(x=350, y=20)
```

```
self.frame2()
```

```
# Functions
```

```
def frame2(self):
```

```
#===== Dashboard
```

```
=====
```

```
self.heading=Label(self.window,text='Dash  
board', font=("",13,"bold"), fg='black',  
bg='#eff5f6')
```

```
self.heading.place(x=325, y=70)
```


#===== Frame 1

=====

```
self.bodyFrame1=Frame(self.window,bg='
#ffffff')
```

```
self.bodyFrame1.place(x=328,y=110,
width=940, height=350)
```

#===== Search

=====

```
self.search_label=Label(self.bodyFrame1,
text='Search Here', fg='black',
bg='#eff5f6',font=("yu gothic ui ", 13,
"bold"))
```

```
self.search_label.place(x=20, y=20)
```

```
self.search_entry=Entry(self.bodyFrame1,
highlightthickness=0,width=540 ,
relief=FLAT, bg="#FFFFFF",
fg="#6b6a69",
```

```
font=("yu gothic ui
", 12, "bold"))
```

```
self.search_entry.place(x=170, y=20)
```

```
self.search_entry.focus_set()
```

```
self.search_canvas =
Canvas(self.bodyFrame1, width=540,
height=0.5, bg="#bdb9b1",
highlightthickness=0)
```

```
self.search_canvas.place(x=170,
```

y=40)

```
self.search_button =  
Button(self.bodyFrame1,text='Search',  
bg='#fa9c05',  
font=("",13,"bold"),bd=0,fg='white',  
cursor='hand2', activebackground='white',  
command=self.search)
```

```
self.search_button.place(x=750,y=15,  
width=150)
```

```
def total_data_info(self):
```

```
    queryy="select count(*) from  
courier_data where status='In Stock' ;"
```

```
    self.cursr=self.db.cursor()
```

```
    self.cursr.execute(queryy)
```

```
    tpl=self.cursr.fetchone()
```

```
    total_stock_data="Total items left in  
stock : {0} ".format(tpl[0])
```

```
self.total_info=Label(self.bodyFrame2,  
text=total_stock_data, fg='black',  
bg='#ffffff',font=("yu gothic ui ", 13,  
"bold"))
```

```
self.total_info.place(x=100, y=70)
```

```
    queryy="select count(*) from  
courier_data;"
```

```
    self.cursr.execute(queryy)
```

```
tpl1=self.cursr.fetchone()
```

```
total_data=""Total items arrived in  
Branch : {0}"".format(tpl1[0])
```

```
self.total_info=Label(self.bodyFrame2,  
text=total_data, fg='black',  
bg='#ffffff',font=("yu gothic ui ", 13,  
"bold"))
```

```
self.total_info.place(x=100, y=100)
```

```
def search(self):
```

```
order_id=self.search_entry.get()
```

```
if not (self.search_entry.get()):
```

```
messagebox.showinfo('Invalid  
Input','Please Enter product id.')
```

```
else:
```

```
try:
```

```
queryy=""select * from  
courier_data where order_id LIKE '%{0}';  
"".format(order_id)
```

```
self.cursr=self.db.cursor()
```

```
self.cursr.execute(queryy)
```

```
tpl=self.cursr.fetchall()
```

```
name=tpl[0][0]
```

```
self.orderid=tpl[0][1]
```

```
address=tpl[0][2]
```

```
phone_no=tpl[0][3]
```

```
payment=tpl[0][4]
```

```
status=tpl[0][5]
```

```
self.info="Name : {0}\n\nOrder  
Id : {1}\n\nAddress : {2}\n\nMobile No. :  
{3}\n\nPayment :  
{4}"".format(name,self.orderid,address,pho  
ne_no,payment)
```

```
info2="Status :  
{0}"".format(status)
```

```
self.showdata_label1=Label(self.bodyFram  
e1,text=self.info, fg='black',  
bg='#ffffff',font=("yu gothic ui ", 13,  
"bold"),justify=LEFT)
```

```
self.showdata_label1.place(x=30,  
y=80)
```

```
self.showdata_label2=Label(self.bodyFram  
e1,text=info2, fg='black',  
bg='#ffffff',font=("yu gothic ui ", 13,  
"bold"),justify=LEFT)
```

```
self.showdata_label2.place(x=500, y=80)
```

```
self.var1=IntVar()
```

```
self.status_pending =  
Checkbutton(self.bodyFrame1,  
text="Delivered", variable=self.var1,
```

```
onvalue=1,  
offvalue=0).place(x=330,y=290)
```

```
        self.submit =  
Button(self.bodyFrame1,text='Submit',  
bg='#fa9c05',  
font=("",13,"bold"),bd=0,fg='white',  
cursor='hand2',  
command=self.change).place(x=450,y=287  
,width=100)
```

```
        #self.Submit =  
Button(self.bodyFrame1,text='Submit',  
bg='#32cf8e',  
font=("",13,"bold"),bd=0,fg='white',  
cursor='hand2',command=self.change(orde  
rid)).place(x=700, y=290)
```

```
except:
```

```
        messagebox.showinfo('Item Not  
Found','No product is in record related to  
this id.')
```

```
def change(self):
```

```
    #status of product change
```

```
    if(self.var1.get()):
```

```
        queryy="update courier_data set  
status='Delivered' where  
order_id={0}";".format(self.orderid)
```

```
        self.cursr=self.db.cursor()
```

```
        self.cursr.execute(queryy)
```

```
        self.db.commit()
```

```
        messagebox.showinfo('Delivery  
status','Order delivered sucessfully.')
```

```
    else:
```

```
        pass
```

```
def Logout(self):
```

```
    LoginPage.Login_Page(self.window)
```

```
def discheck1(self):
```

```
    self.status_pending.deselect()
```

```
def discheck2(self):
```

```
    self.status_done.deselect()
```

```
def discheck11(self):
```

```
    self.status_instock.deselect()
```

```
def discheck22(self):
```

```
    self.status_delivered.deselect()
```

```
def adding(self):
```

```
    for widget in  
self.bodyFrame1.winfo_children():
```

```
        widget.destroy()
```

```
#taking details of new product
```

```
self.name_label =  
Label(self.bodyFrame1, text="Name",  
bg="#FFFFFF", fg="#4f4e4d", font=("yu  
gothic ui", 13, "bold")).place(x=50, y=50)
```

```
self.orderid_label =  
Label(self.bodyFrame1, text="Order Id",  
bg="#FFFFFF", fg="#4f4e4d", font=("yu  
gothic ui", 13, "bold")).place(x=350, y=50)
```

```
self.address_label =  
Label(self.bodyFrame1, text="Address",  
bg="#FFFFFF", fg="#4f4e4d", font=("yu  
gothic ui", 13, "bold")).place(x=650, y=50)
```

```
self.mo_no_label =  
Label(self.bodyFrame1, text="Mo. No.",  
bg="#FFFFFF", fg="#4f4e4d", font=("yu  
gothic ui", 13, "bold")).place(x=50, y=150)
```

```
self.payment_label =  
Label(self.bodyFrame1, text="Payment  
Status", bg="#FFFFFF",  
fg="#4f4e4d", font=("yu gothic ui", 13,  
"bold")).place(x=350, y=150)
```

```
self.status_label =  
Label(self.bodyFrame1, text="Delivery  
Status", bg="#FFFFFF",  
fg="#4f4e4d", font=("yu gothic ui", 13,  
"bold")).place(x=600, y=150)
```

```
#payment status
```

```
self.var1=IntVar()
```

```
self.var2=IntVar()
```

```
self.status_pending =  
Checkbutton(self.bodyFrame1,  
text="Pending", variable=self.var1,  
onvalue=1, offvalue=0,  
command=self.discard2)
```

```
self.status_pending.pack()
```

```
self.status_pending.place(x=350,y=200)
```

```
self.status_pending.select()
```

```
self.status_done =  
Checkbutton(self.bodyFrame1,  
text="Done", variable=self.var2,  
onvalue=1, offvalue=0,  
command=self.discheck1)
```

```
self.status_done.pack()
```

```
self.status_done.place(x=450, y=200)
```

```
#delivery status
```

```
self.var3=IntVar()
```

```
self.var4=IntVar()
```

```
self.status_instock =  
Checkbutton(self.bodyFrame1, text="In  
Stock", variable=self.var3, onvalue=1,  
offvalue=0, command=self.discheck22)
```

```
self.status_instock.pack()
```

```
self.status_instock.place(x=600,y=200)
```

```
self.status_instock.select()
```

```
self.status_delivered =  
Checkbutton(self.bodyFrame1,  
text="Delivered", variable=self.var4,  
onvalue=1, offvalue=0,  
command=self.discheck11)
```

```
self.status_delivered.pack()
```

```
self.status_delivered.place(x=700,  
y=200)
```


#entries

```
self.name_entry =  
Entry(self.bodyFrame1,  
highlightthickness=0, relief=FLAT,  
bg="#FFFFFF", fg="#6b6a69",font=("yu  
gothic ui", 12, "bold"))
```

```
self.name_entry.pack()
```

```
self.name_entry.place(x=55, y=85,  
width=258)
```

```
self.orderid_entry =  
Entry(self.bodyFrame1,  
highlightthickness=0, relief=FLAT,  
bg="#FFFFFF", fg="#6b6a69",font=("yu  
gothic ui", 12, "bold"))
```

```
self.orderid_entry.pack()
```

```
self.orderid_entry.place(x=355, y=85,  
width=258)
```

```
self.address_entry =  
Entry(self.bodyFrame1,  
highlightthickness=0, relief=FLAT,  
bg="#FFFFFF", fg="#6b6a69",font=("yu  
gothic ui", 12, "bold"))
```

```
self.address_entry.pack()
```

```
self.address_entry.place(x=655, y=85,  
width=258)
```

```
self.mo_no_entry =  
Entry(self.bodyFrame1,  
highlightthickness=0, relief=FLAT,  
bg="#FFFFFF", fg="#6b6a69",font=("yu  
gothic ui", 12, "bold"))
```

```
self.mo_no_entry.pack()
```

```
self.mo_no_entry.place(x=55, y=185,  
width=258)
```

```
#focus on name
```

```
self.name_entry.focus_set()
```

```
#canvas
```

```
self.name_line =
```

```
Canvas(self.bodyFrame1, width=261,  
height=2.0, bg="#bdb9b1",  
highlightthickness=0).place(x=55, y=110)
```

```
self.orderid_line =
```

```
Canvas(self.bodyFrame1, width=261,  
height=2.0, bg="#bdb9b1",  
highlightthickness=0).place(x=355, y=110)
```

```
self.address_line =
```

```
Canvas(self.bodyFrame1, width=261,  
height=2.0, bg="#bdb9b1",  
highlightthickness=0).place(x=655, y=110)
```

```
self.mo_no_line =
```

```
Canvas(self.bodyFrame1, width=261,  
height=2.0, bg="#bdb9b1",  
highlightthickness=0).place(x=55, y=210)
```

```
#add button
```

```
self.add_button =
```

```
Button(self.bodyFrame1, text='Add',  
bg='#fa9c05',  
font=("", 13, "bold"), bd=0, fg='white',  
command=self.addproduct,  
relief=FLAT, activebackground="#1c355e",  
borderwidth=0, cursor="hand2")
```

```
self.add_button.place(x=320, y=270,  
width=250)
```

```
def addproduct(self):
```

```
try:
```

```
    name=self.name_entry.get()
```

```
    orderid=self.orderid_entry.get()
```

```
    address=self.address_entry.get()
```

```
    mo_no=self.mo_no_entry.get()
```

```
    if self.var1.get():
```

```
        payment='Pending'
```

```
    else:
```

```
        payment='Done'
```

```
    if self.var3.get():
```

```
        delivery='In Stock'
```

```
    else:
```

```
        delivery='Delivered'
```

```
self.insertRecord(name,orderid,address,mo  
_no,payment,delivery)
```

```
    messagebox.showinfo('Product  
info','Product added sucessfully.')
```

```
except DatabaseError:
```

```
    messagebox.showerror('record  
error','Product already exist / database  
error!')
```

```
except:
```

```
    messagebox.showerror('record  
error','Incomplete details.')
```

```
def
```

```
insertRecord(self,name,orderid,address,mo  
_no,payment,delivery):
```

```
    createTablee=""
```

```
    create table if not exists courier_data(
```

```
        Name varchar(50) not null,order_id  
varchar(30) primary key,address  
varchar(100) not null,phone_no int unique  
not null,payment varchar(10) not  
null,status char(10) not null
```

```
    );
```

```
    ""
```

```
    self.db.cursor().execute(createTablee)
```

```
    queryInsert=""insert into courier_data  
values('{0}','{1}','{2}','{3}','{4}','{5}')".for  
mat(name,orderid,address,mo_no,payment,  
delivery)
```

```
    try:
```

```
self.db.cursor().execute(queryInsert)
```

```
    self.db.commit()
```

```
    except:
```

```
        raise DatabaseError('User already  
exist or database error!')
```