PYTHON PROGRAMMING (INT 213)

PROJECT REPORT

ON

Courier Management System for LPU

Bachelor Of Technology (CSE)

Submitted by

NAME	REGISTRATION NUMBER	ROLL NUMBER
Prabhu Pathak	12105248	22
Kurapati Yaswanth Krishna Chowdary	12105417	23
Sonu Kumar Yadav	12110335	24

Submitted to Ishan Kumar

Introduction

A **Courier Management System or CMS** is a business software that simplifies courier management and routing. A CMS streamlines all of the tasks which include Planning and Optimizing delivery routes, Courier Tracking, and Scheduling.

This Project is a Customized Courier Management System for Lovely Professional University with the theme of the University itself.

The **Courier Management System** In Python here has a create account form and login page, under the creating account module includes such as username, password, reg. no, gender, mobile number and email id. And after the user creates his/her account, the user can now login to the system and monitor their items.

Role and Responsibility of Each Student

1.Sonu Kumar Yadav:

Designing the Login page , and some part of the New User page, Implementing the Login , New User Buttons using tkinter applications and Graphical User Interface(GUI), Implementing the photographs on the interface and contributed in report.

2. Kurapati Yaswanth Krishna Chowdary:

Designing the remaining part of the New User window with dialogue boxes, Create account button using tkinter applications and GUI, dialogue box, linking of all the modules, and Preparing the final report of the project.

3. Prabhu Pathak:

Designing the Track Consignment page and Shipment Status page with Track button using tkinter applications and GUI, Dialogue boxes, Making of Presentation of the project(PPT)

Module Wise Description

1.Login Page:

When the user uses the program, it displays the Login page which prompts the user to enter Username, Password options, to login to the account, along with Login and New User buttons present. After Logging in with correct details, it directs to the tracking page.

2.Create Account Page:

If the user is new to the application, then he/she needs to create a new account, which is where the New User option comes into the role. This module consists of fields which includes Username, Password, Reg Number, Gender (consists of Radio buttons Male, Female, other) , User's Mobile Number and Email ID to create a new account. Which after creation displays a dialogue box saying "Account Created", and the details of registration are stored in the database.

3.Track your Product

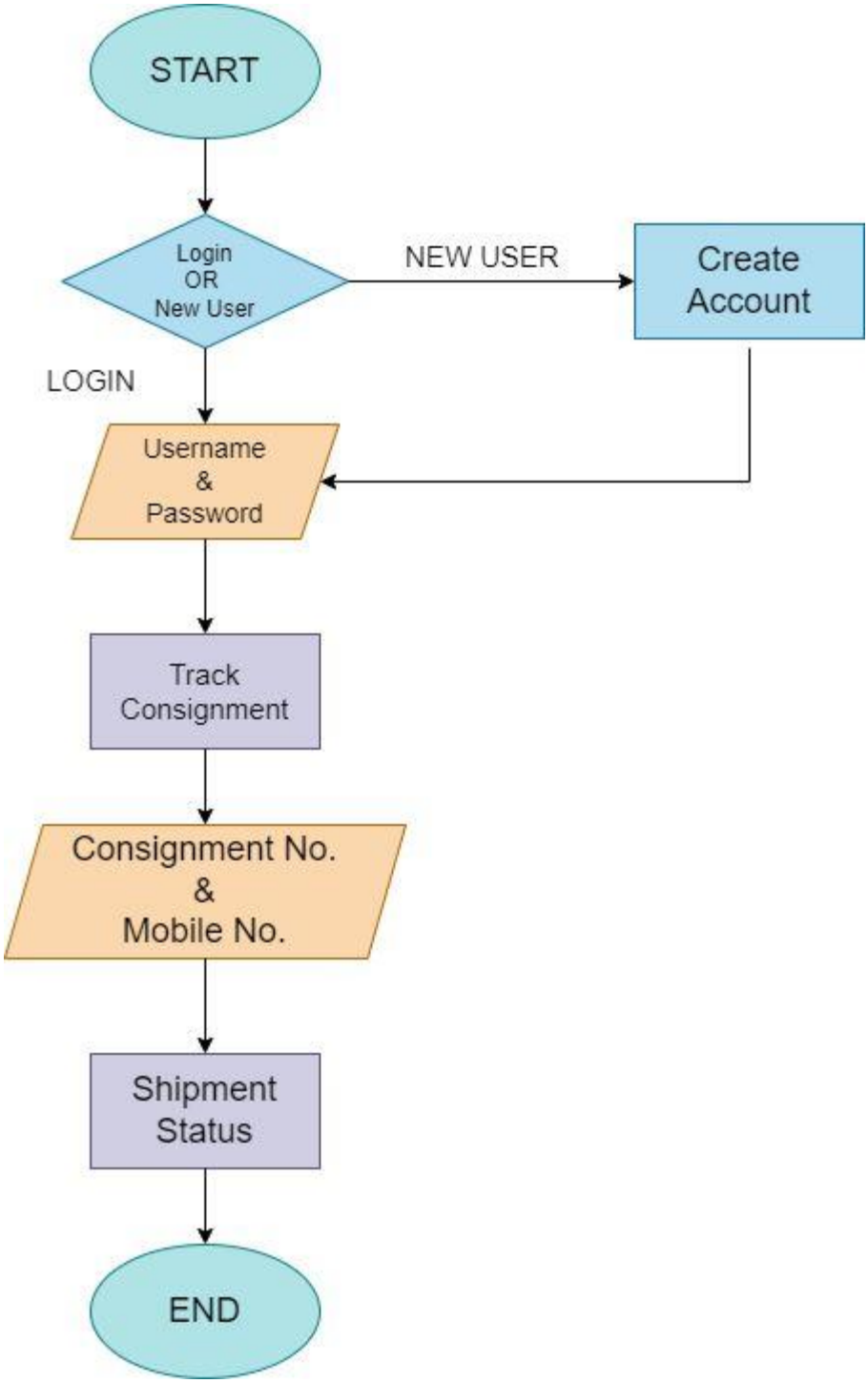
After the user creates a new account it asks to login, where the user enters the Login credentials to login which directs to the Track

Shipment page. The user needs to enter the Consignment Number (which he after couriering the product) and his registered mobile number to get the Status of the shipment, Delivery date and Order details. Incase the entered mobile number is incorrect or is not the same as the one given while creating the account, it displays a dialogue box which says “Mobile Number does not exist”.

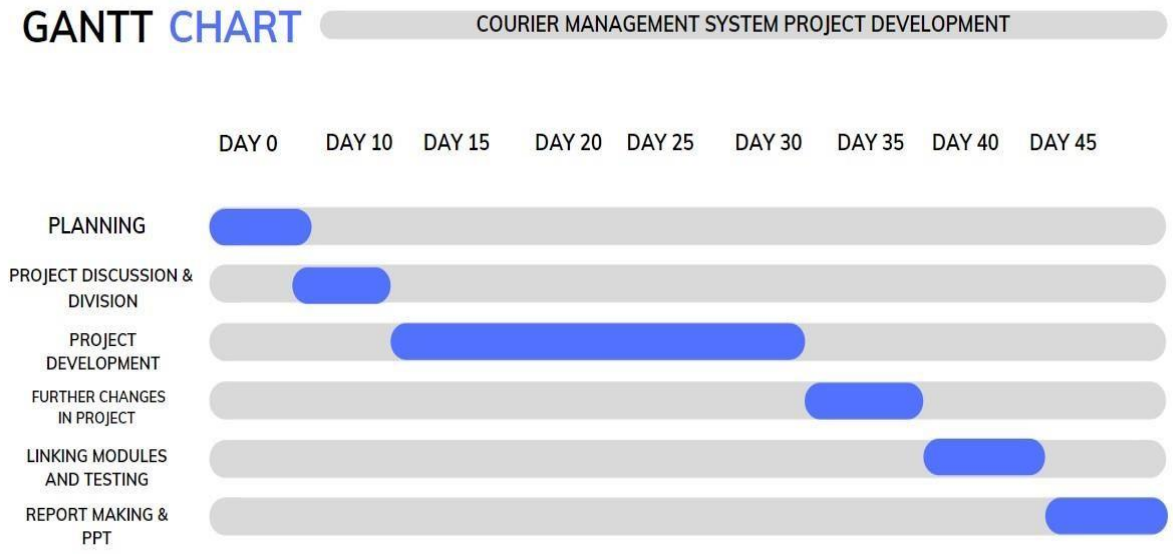
4.Shipment status:

The shipment status page displays the order details, Courier status and Expected delivery date with comment box for any queries.

FLOWCHART


START
Login
OR
New User
NEW USER
Create
Account
LOGIN
Username
&
Password
Track
Consignment
Consignment No.
&
Mobile No.
Shipment
Status
END

GHANTT CHART



SCREENSHOTS AND THE OUTPUT OF THE CODES

```python
from tkinter import *
from tkinter import messagebox as ms
from PIL import ImageTk
from tkinter import ttk
import sqlite3
import random
from tkinter import Button

# Database
with sqlite3.connect('SPY.db') as db:
    c = db.cursor()
try:
    c.execute('CREATE TABLE IF NOT EXISTS user (username TEXT NOT NULL ,password TEX  NOT NULL,mobile TEX NOT NULL);')
except:
    pass
db.commit()
db.close()

class main:
    def __init__(self, master):

        self.master = master

        self.username = StringVar()
        self.password = StringVar()
        self.n_username = StringVar()
        self.n_password = StringVar()
        self.n_reg = StringVar()
        self.n_mobile = StringVar()
        self.mobile11 = StringVar()
        self.widgets()
```

```python
    def login(self):

        with sqlite3.connect('SPY.db') as db:
            c = db.cursor()

        # Find user If there is any take proper action

        find_user = ('SELECT * FROM user WHERE username = ? and password = ?')
        c.execute(find_user, [(self.username.get()), (self.password.get())])
        result = c.fetchall()

        if result:
            self.track()
        else:
            ms.showerror('Oops!', 'Username Not Found.')

    def new_user(self):

        with sqlite3.connect('SPY.db') as db:
            c = db.cursor()
        if self.n_username.get() != ' ' and self.n_password.get() != ' ' and self.n_mobile.get() != ' ':
            find_user = ('SELECT * FROM user WHERE username = ?')
            c.execute(find_user, [(self.n_username.get())])

            if c.fetchall():
                ms.showerror('Error!', 'Username Taken Try a Diffrent One.')
            else:
                insert = 'INSERT INTO user(username,password,mobile) VALUES(?,?,?)'
                c.execute(insert, [(self.n_username.get()),
                           (self.n_password.get()), (self.n_mobile.get())])
                db.commit()
```

```
                ms.showinfo('Success!', 'Account Created!')
                self.log()
        else:
            ms.showerror('Error!', 'Please Enter the details.')

    def consignment(self):

        try:
            with sqlite3.connect('SPY.db') as db:
                c = db.cursor()

            # Find user If there is any take proper action

            find_user = ('SELECT * FROM user WHERE mobile= ?')
            c.execute(find_user, [(self.mobile11.get())])
            result = c.fetchall()

            if result:
                self.track()
                self.crff.pack_forget()
                self.head['text'] = self.username.get() + \
                    '\n Your Product Details'
                self.consi.pack()
            else:
                ms.showerror('Oops!', 'Mobile Number Not Found.')
        except:
            ms.showerror('Oops!', 'Mobile Number Not Found.')

    def track1(self):
        self.consi.pack_forget()
        self.head['text'] = self.username.get() + '\n Track your Product'
        self.crff.pack()
```

```python
def log(self):
    self.username.set('')
    self.password.set('')
    self.crf.pack_forget()
    self.head['text'] = 'RE-LOGIN'
    self.logf.pack()

def cr(self):
    self.n_username.set('')
    self.n_password.set('')
    self.logf.pack_forget()
    self.head['text'] = 'CREATE NEW ACCOUNT'
    self.crf.pack()

def track(self):
    self.logf.pack_forget()
    self.head['text'] = self.username.get() + '\n Track your Product'

    self.crff.pack()

def widgets(self):
    self.head = Label(self.master, text='LOGIN', font=("Times", 20), pady=10)
    self.head.pack()

    self.logf = Frame(self.master, padx=10, pady=10)

    self.logf.configure(background='light blue')


    Label(self.logf, text='Username: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
    Entry(self.logf, textvariable=self.username,bd=5, font=("Times", 15)).grid(row=0, column=1)
```

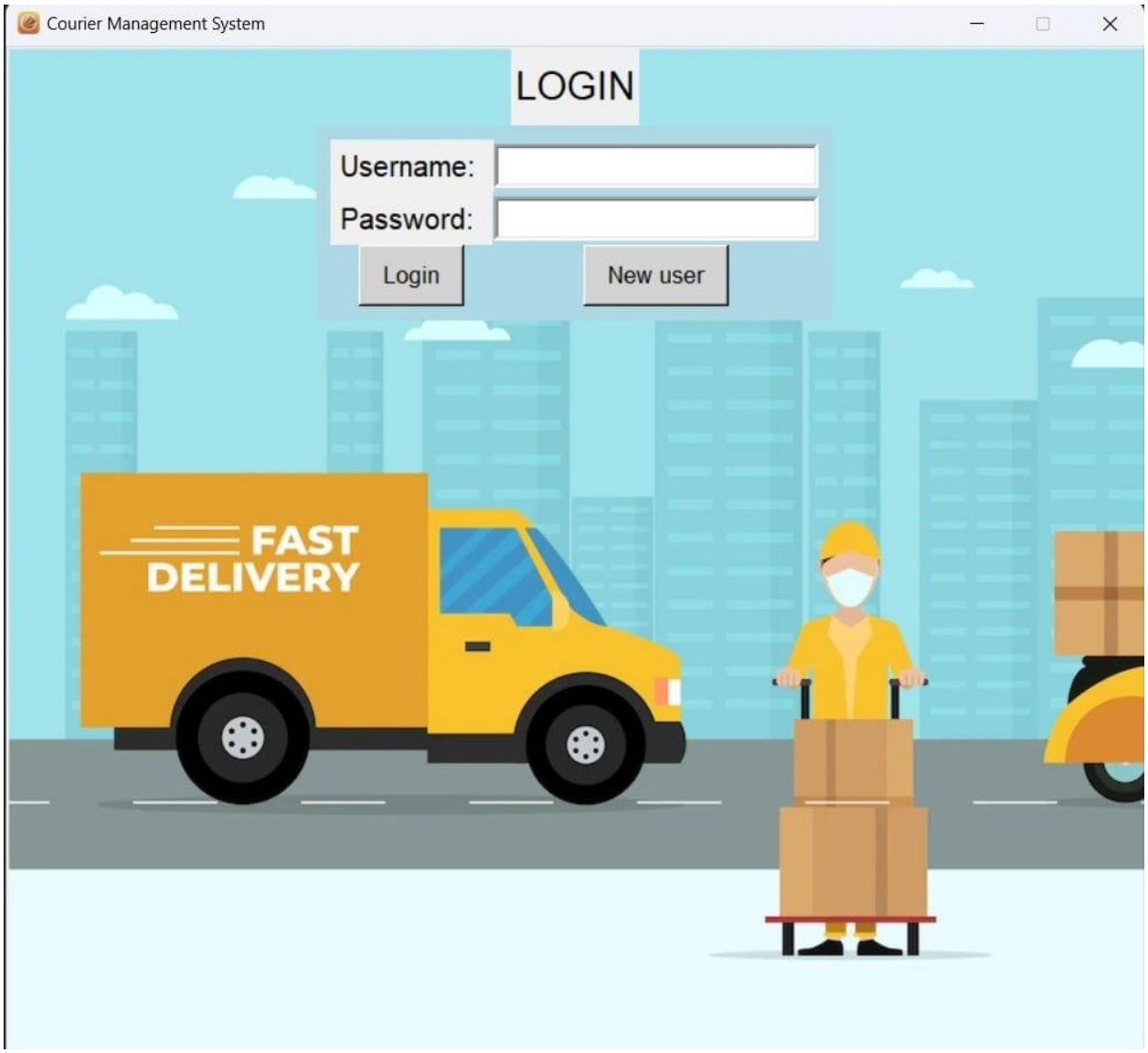
```
    Label(self.logf, text='Password: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
    Entry(self.logf, textvariable=self.password, bd=5,font=("Times", 15), show='*').grid(row=1, column=1)

    Button(self.logf, text=' Login ', background='#FFFFE4', bd=5, font=("Times", 13), padx=6, pady=6, command=self.login).grid(row=8, c
    Button(self.logf, text=' New user ', background='#FFFFE4', bd=5, font=("Times", 13), padx=6, pady=6, command=self.cr).grid(row=8, c

    self.logf.pack()

    self.crf = Frame(self.master, padx=10, pady=10)
    Label(self.crf, text='Username: ', font=('Arial', 15), pady=5, padx=5).grid(sticky=W)
    Entry(self.crf, textvariable=self.n_username,bd=5, font=("Times", 15)).grid(row=0, column=1)

    Label(self.crf, text='Password: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
    Entry(self.crf, textvariable=self.n_password, bd=5,font=("Times", 15), show='*').grid(row=1, column=1)

    Label(self.crf, text='Reg No.: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
    Entry(self.crf, textvariable=self.n_reg, bd=5,font=("Times", 15)).grid(row=2, column=1)

    Label(self.crf, text='Gender: ', font=( "Times", 15), pady=5, padx=5).grid(sticky=W)

    var = IntVar()
    R1 = Radiobutton(self.crf, text="Male", variable=var, value=1).grid(sticky=W)
    R2 = Radiobutton(self.crf, text="Female", variable=var, value=2).grid(row=4, column=1)

    Label(self.crf, text='Mobile No.: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
    Entry(self.crf, textvariable=self.n_mobile, bd=5, font=("Times", 15)).grid(row=5, column=1)

    Label(self.crf, text='Email Id: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
    Entry(self.crf, bd=5, font=("Times", 15)).grid(row=6, column=1)

    Button(self.crf, text='Create Account', background='#FFFFE4', bd=5, font=("Times", 13), padx=6, pady=6, command=self.new_user).grid
    Button(self.crf, text='Go to Login', background='#FFFFE4', bd=5, font=("Times", 13), padx=6, pady=6, command=self.log).grid(row=11,
```

```
Button(self.crf, text='Create Account', background='#FFFFE4', bd=5, font=("Times", 13), padx=6, pady=6, command=self.new_user).grid
Button(self.crf, text='Go to Login', background='#FFFFE4', bd=5, font=("Times", 13), padx=6, pady=6, command=self.log).grid(row=11,

self.crff = Frame(self.master, padx=10, pady=10)

Label(self.crff, text='Consignment No: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
Entry(self.crff, bd=5, font=("Times", 15)).grid(row=0, column=1)

Label(self.crff, text='Mobile no:', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
Entry(self.crff, bd=5, textvariable=self.mobile11,font=("Times", 15)).grid(row=1, column=1)

Button(self.crff, text='Track', background='#FFFFE4', bd=5, font=("Times", 13), padx=6, pady=6, command=self.consignment).grid(row=

self.consi = Frame(self.master, padx=10, pady=10)

Label(self.consi, text=' Product ID:', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
Label(self.consi, text=random.randint(565154, 99994216),font=("Times", 13), pady=5, padx=5).grid(row=0, column=1)

L = ['Travel-Bag', 'Python Book', 'Redmi Note 8', 'Heater', 'MACBOOK-AIR',
     'Wedding-Ring', 'Car-Cover', 'Ipad', 'Haldiram Bhujiya', 'Water-Bottle', 'Placements-Series Bundle']

f = random.randint(0, 10)
Label(self.consi, text='Product Name: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
Label(self.consi, text=L[f], font=("Times", 13),pady=5, padx=5).grid(row=1, column=1)
Label(self.consi, text='Product Status: ', font=("Times", 15), pady=5, padx=5).grid(sticky=W)
Label(self.consi, text='Pending', font=("Times", 13),pady=5, padx=5).grid(row=2, column=1)
Label(self.consi, font=("Times", 13),text='Thanks for Exploring!').grid(row=4, column=0)

Label(self.consi, text='Comments:', font=("Times", 13)).grid(row=5, column=0, padx=5, sticky='sw')
Entry(self.consi, bd=5, font=("Times", 15)).grid(row=5, column=1)
```

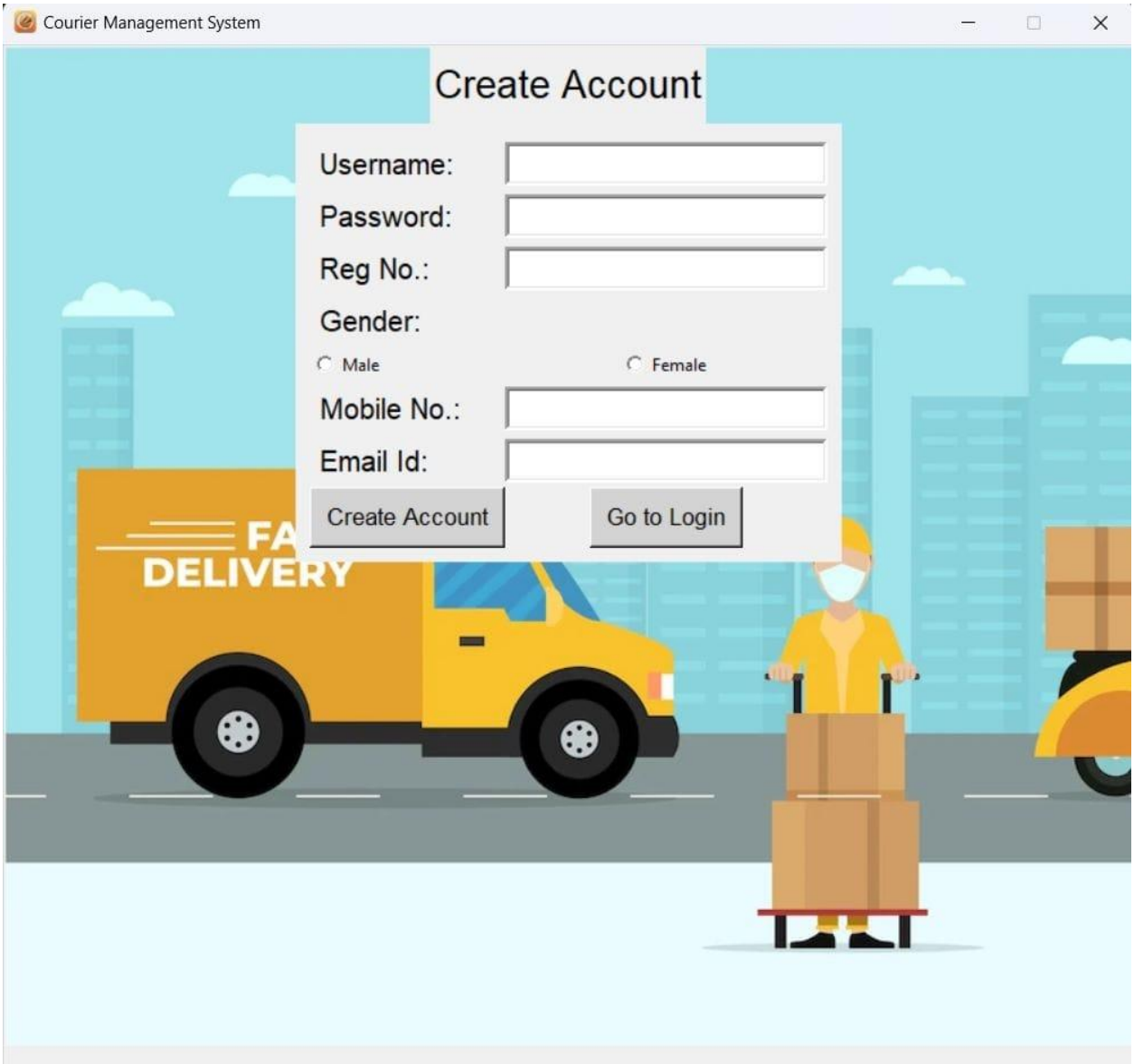
```
        Button(self.consi, text='Back', background='#FFFFE4', bd=5, font=("Times", 13), padx=6, pady=6, command=self.track1).grid(row=6, co

if __name__ == '__main__':

    courier_system = Tk()

    courier_system.resizable(0,0)
    bgImage = ImageTk.PhotoImage(file='7.webp')

    bgLabel = Label(courier_system, image=bgImage)
    bgLabel.place(x=0, y=0)
    courier_system.iconbitmap('lpu.ico')
    courier_system.title('Courier Management System')
    courier_system.geometry('800x1000+300+300')
    main(courier_system)

    courier_system.mainloop()
```

OUTPUT OF THE CODE

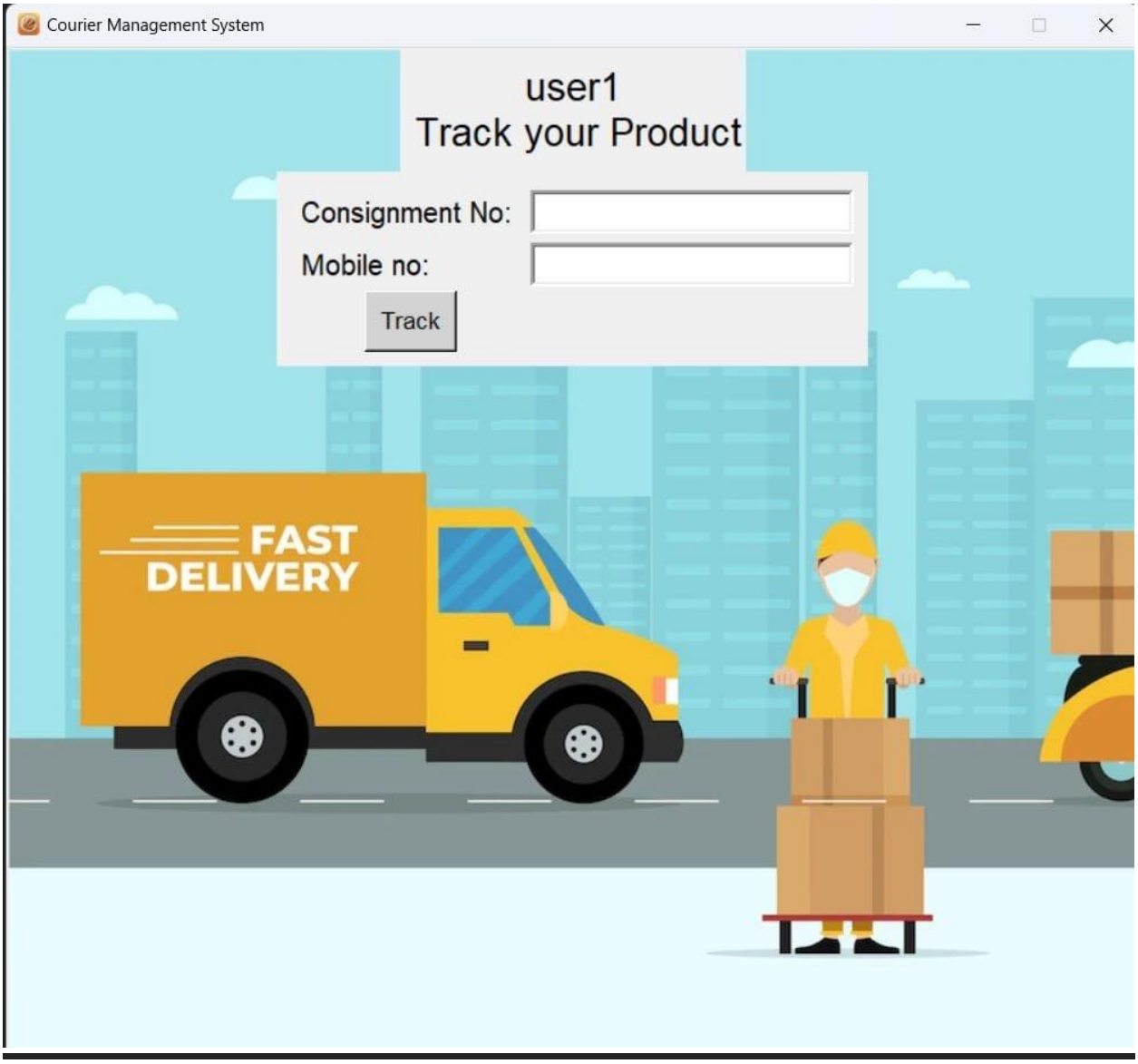
LOGIN



CREATE NEW ACCOUNT



TRACK SHIPMENT



SHIPMENT STATUS