

NAME : ADITYA DAS

CLASS : XII

ROLL :

SECTION : B

SUBJECT : COMPUTER SCIENCE

TOPIC : GROCERY SHOP MANAGEMENT

CERTIFICATE

This is to certify that ADITYA DAS ,a student of class - 12th SCIENCE has successfully completed his Computer Project on GROCERY SHOP MANAGEMENT SYSTEM under the guidance of MR. ROCHISNU SAHA.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

INTERNAL EXAMINER EXTERNAL EXAMINER



GROCERIA

DOCUMENTATION



CLASS : XII

SUBJECT : COMPUTER SCIENCE

PROJECT : GRCOERY SHOP MANAGMENT SYSTEM

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to all these individuals for mentoring and supporting me in completing this project GROCERIA  
My teacher Rochisnu Saha, for providing me with invaluable insights and direction.  
To my parents, their constant encouragement, patience, and understanding have been the pillars of my success.  
I am grateful to my friends who contributed ideas and perspectives that enriched the project.  
Thank you everyone for shaping this project and enhancing my learning experience.

**INDEX**

|  |  |
| --- | --- |
| TOPIC | PAGE |
| OBJECTIVE | 5 |
| INPUT SEGMENT | 6 - 78 |
| OUTPUT SEGMENT | 79 - 82 |
| FUNCTIONS AND FEATURES PROPOSED IN THE SYSTEM | 83 |
| FRONTEND AND BACKED USED | 84 |
| HARDWARE AND  SOFTWARE REQUIREMENT | 85 |
| ROLE OF STUDENTS | 86 |
| BIBLIOGRAPHY | 87 |

**OBJECTIVE**

The objective of our project is to design a grocery shop management system named **"GROCERIA”** using **Python, Tkinter** and **MySql**. The system aims to provide a dynamic view of a user friendly interface where users can buy any grocery items they wish from their homes itself.

KEY FEATURES :

* **ACCOUNT CREATION :** Users can create a new account by providing neccessary details such as username , real name and their desired password.
* **CART :** Users can wish to add different items from the store which they may decide to buy later.
* **CUSTOMER SUPPORT :** If the user is unable to understand anything , we have an exclusive customer support option which shows the layout of every corner of our app.
* **REMOTE ACCESSIBILITY :** Users can buy groceries right from their home which will be delivered within 30 mins without being externally present at the shop.
* **HISTORY :** We have a database for every user where users can see what products they bought in the recent history.

**INPUT SEGMENT**

import customtkinter as ctk

import tkinter as tk,tkinter

from tkinter import \*

from PIL import Image,ImageTk,ImageSequence

import mysql.connector

import time as time

import re

import itertools

from threading import Thread

import smtplib

import mysql.connector

import random

from hashlib import sha256

from tkinter import messagebox

from email.mime.text import MIMEText

from email.mime.multipart import MIMEMultipart

from datetime import datetime

global search\_frame

# SMTP Configuration

SMTP\_SERVER = "smtp.gmail.com"

SMTP\_PORT = 587

EMAIL\_ADDRESS = "groceria@gmail.com" # Replace with your email

EMAIL\_PASSWORD = "bbit@123" # Replace with your email password

def send\_email(to\_email, subject, body):

try:

with smtplib.SMTP(SMTP\_SERVER, SMTP\_PORT) as server:

server.starttls()

server.login(EMAIL\_ADDRESS, EMAIL\_PASSWORD)

message = MIMEMultipart()

message["From"] = EMAIL\_ADDRESS

message["To"] = to\_email

message["Subject"] = subject

message.attach(MIMEText(body, "plain"))

server.send\_message(message)

print(f"Email sent to {to\_email}")

except Exception as e:

print(f"Failed to send email to {to\_email}: {e}")

#### USER DATABASE [SERVER] ####

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS users (

email\_id VARCHAR PRIMARY KEY,

first\_name VARCHAR NOT NULL,

last\_name VARCHAR NOT NULL,

pass\_word VARCHAR NOT NULL,

address VARCHAR,

pincode VARCHAR)''')

def insert\_user():

email\_id = email\_address.get()

first\_name = fullname\_entry.get()

last\_name = lastname\_entry.get()

pass\_word = password.get()

# Hash the password before storing it

hashed\_password = sha256(pass\_word.encode()).hexdigest()

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute("INSERT INTO users (email\_id, first\_name, last\_name, pass\_word) VALUES (?, ?, ?, ?)",

(email\_id, first\_name, last\_name, hashed\_password))

messagebox.showinfo('Success','Successfully registered.')

except mysql.IntegrityError:

messagebox.showinfo('Error','Registered Email Id .')

print("Error: Email ID already exists.")

### SALES TABLE ###

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS sales (

product\_id VARCHAR PRIMARY KEY,

exp\_date VARCHAR NOT NULL,

price VARCHAR NOT NULL,

quantity VARCHAR NOT NULL,

discount VARCHAR NOT NULL,

product\_name VARCHAR NOT NULL,

description VARCHAR NOT NULL)''')

#### CATEGORY PRODUCT TABLES ####

## VEGETABLES ##

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS veges (

product\_id VARCHAR PRIMARY KEY,

exp\_date VARCHAR NOT NULL,

price VARCHAR NOT NULL,

quantity VARCHAR NOT NULL,

discount VARCHAR NOT NULL,

product\_name VARCHAR NOT NULL,

description VARCHAR NOT NULL)''')

## FRUITS ##

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS fruits (

product\_id VARCHAR PRIMARY KEY,

exp\_date VARCHAR NOT NULL,

price VARCHAR NOT NULL,

quantity VARCHAR NOT NULL,

discount VARCHAR NOT NULL,

product\_name VARCHAR NOT NULL,

description VARCHAR NOT NULL)''')

## DAIRY ##

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS dairy (

product\_id VARCHAR PRIMARY KEY,

exp\_date VARCHAR NOT NULL,

price VARCHAR NOT NULL,

quantity VARCHAR NOT NULL,

discount VARCHAR NOT NULL,

product\_name VARCHAR NOT NULL,

description VARCHAR NOT NULL)''')

## ESSENTIALS ##

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute('''CREATE TABLE IF NOT EXISTS supplies (

product\_id VARCHAR PRIMARY KEY,

exp\_date VARCHAR NOT NULL,

price VARCHAR NOT NULL,

quantity VARCHAR NOT NULL,

discount VARCHAR NOT NULL,

product\_name VARCHAR NOT NULL,

description VARCHAR NOT NULL)''')

# Dairy products

dairy\_products = [

('d001', '15/01/2025', '250', '250g', '0', 'Paneer', 'Fresh Indian cheese'),

('d002', '18/01/2025', '125', '1 liter', '0', 'Milk', 'Fresh cow milk'),

('d003', '20/01/2025', '180', '200g', '0', 'Butter', 'Creamy butter'),

('d004', '22/01/2025', '200', '200g', '0', 'Cheese', 'Sliced cheddar cheese'),

('d005', '24/01/2025', '80', '500g', '0', 'Yogurt', 'Fresh, natural yogurt'),

('d006', '26/01/2025', '550', '500g', '0', 'Ghee', 'Pure clarified butter'),

('d007', '28/01/2025', '120', '200ml', '0', 'Cream', 'Thick, fresh cream'),

('d008', '30/01/2025', '70', '500g', '0', 'Curd', 'Fresh, natural curd'),

('d009', '01/02/2025', '40', '1 liter', '0', 'Buttermilk', 'Refreshing buttermilk'),

('d010', '03/02/2025', '30', '200ml', '0', 'Lassi', 'Sweetened lassi')

]

# Veges products

veges\_products = [

('v001', '20/01/2025', '300', '1 kg', '0', 'Bell Pepper', 'Fresh and crunchy bell peppers'),

('v002', '22/01/2025', '80', '1 bunch', '0', 'Green Onion', 'Fresh green onions'),

('v003', '24/01/2025', '210', '1 kg', '0', 'Eggplant', 'Fresh eggplants'),

('v004', '26/01/2025', '120', '1 bunch', '0', 'Spinach', 'Fresh spinach leaves'),

('v005', '28/01/2025', '100', '1 head', '0', 'Lettuce', 'Fresh and crisp lettuce'),

('v006', '30/01/2025', '170', '1 piece', '0', 'Cabbage', 'Fresh green cabbage'),

('v007', '01/02/2025', '160', '1 kg', '0', 'Potatoes', 'Fresh potatoes'),

('v008', '03/02/2025', '250', '1 piece', '0', 'Cauliflower', 'Fresh cauliflower heads'),

('v009', '05/02/2025', '65', '1 bunch', '0', 'Coriander', 'Fresh coriander leaves'),

('v010', '07/02/2025', '90', '1 kg', '0', 'Carrot', 'Fresh carrots')

]

# Fruits products

fruits\_products = [

('f001', '07/02/2025', '400', '1 piece', '0', 'Watermelon', 'Sweet and juicy watermelon'),

('f002', '09/02/2025', '330', '1 kg', '0', 'Apple', 'Fresh, crisp apples'),

('f003', '11/02/2025', '250', '1 kg', '0', 'Oranges', 'Fresh and tangy oranges'),

('f004', '13/02/2025', '80', '1 dozen', '0', 'Banana', 'Ripe bananas'),

('f005', '15/02/2025', '200', '1 kg', '0', 'Grapes', 'Fresh grapes'),

('f006', '17/02/2025', '300', '1 kg', '0', 'Mango', 'Sweet mangoes'),

('f007', '19/02/2025', '150', '1 piece', '0', 'Papaya', 'Fresh papaya'),

('f008', '21/02/2025', '200', '1 piece', '0', 'Pineapple', 'Sweet pineapple'),

('f009', '23/02/2025', '350', '1 kg', '0', 'Pomegranate', 'Juicy pomegranates'),

('f010', '25/02/2025', '250', '250g', '0', 'Strawberries', 'Fresh strawberries')

]

# Supplies products

supplies\_products = [

('s001', '13/02/2025', '50', '500g', '0', 'Salt', 'Iodized salt, good for health'),

('s002', '15/02/2025', '150', '200g', '0', 'Chilli Powder', 'Authentic Kashmiri chilli powder'),

('s003', '17/02/2025', '200', '100g', '0', 'Garam Masala', 'Pure garam masala to elevate the taste'),

('s004', '19/02/2025', '100', '100g', '0', 'Turmeric Powder', 'Pure turmeric powder'),

('s005', '21/02/2025', '60', '1 kg', '0', 'Sugar', 'Granulated sugar'),

('s006', '23/02/2025', '120', '1 kg', '0', 'Rice', 'Basmati rice'),

('s007', '25/02/2025', '70', '1 kg', '0', 'Wheat Flour', 'Whole wheat flour'),

('s008', '27/02/2025', '80', '1 kg', '0', 'Pulses', 'Split chickpeas'),

('s009', '01/03/2025', '150', '250g', '0', 'Tea Powder', 'Strong tea powder'),

('s010', '03/03/2025', '180', '1 liter', '0', 'Cooking Oil', 'Sunflower cooking oil')

]

#SALES PRODUCTS

sales\_products=[('s006', '23/02/2025', '120', '1 kg', '0', 'Rice', 'Basmati rice'),('f009', '23/02/2025', '350', '1 kg', '0', 'Pomegranate', 'Juicy pomegranates'),('v004', '26/01/2025', '120', '1 bunch', '0', 'Spinach', 'Fresh spinach leaves'),('d005', '24/01/2025', '80', '500g', '0', 'Yogurt', 'Fresh, natural yogurt'),('f010', '25/02/2025', '250', '250g', '0', 'Strawberries', 'Fresh strawberries')]

# Insert products into tables

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.executemany('INSERT OR REPLACE INTO dairy (product\_id, exp\_date, price, quantity, discount, product\_name, description) VALUES (?, ?, ?, ?, ?, ?, ?)', dairy\_products)

cursor.executemany('INSERT OR REPLACE INTO veges (product\_id, exp\_date, price, quantity, discount, product\_name, description) VALUES (?, ?, ?, ?, ?, ?, ?)', veges\_products)

cursor.executemany('INSERT OR IGNORE INTO fruits (product\_id, exp\_date, price, quantity, discount, product\_name, description) VALUES (?, ?, ?, ?, ?, ?, ?)', fruits\_products)

cursor.executemany('INSERT OR IGNORE INTO supplies (product\_id, exp\_date, price, quantity, discount, product\_name, description) VALUES (?, ?, ?, ?, ?, ?, ?)', supplies\_products)

cursor.executemany('INSERT OR IGNORE INTO sales (product\_id, exp\_date, price, quantity, discount, product\_name, description) VALUES (?, ?, ?, ?, ?, ?, ?)', sales\_products)

#SIGN UP PAGE

window=ctk.CTk()

window.configure(fg\_color='#093e44')

window.title("Groceria")

window.geometry('1120x630')

cat\_click\_count=0

my\_font=ctk.CTkFont(family='Vivaldi Italic',size=50)

fnt=ctk.CTkFont(family='Vivaldi Italics',size=15)

fnt1=ctk.CTkFont(family='Vivaldi Italics',size=20)

fnt2=ctk.CTkFont(family='Vivaldi Italics',size=40)

logo\_image=ctk.CTkImage(Image.open(r"logo1.png"),size=(80,80))

logo\_image\_one=ctk.CTkImage(Image.open(r"logo1.png"),size=(65,65))

background\_image1=ctk.CTkImage(Image.open(r"groceria.png"),size=(450,450))

background\_image=ctk.CTkImage(Image.open(r"groceria.png"),size=(800,300))

#Sign Up Mainframe

signup\_mainframe=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

signup\_mainframe.place(x=30,y=30)

signup\_backimage=ctk.CTkLabel(signup\_mainframe,text='',image=background\_image1)

signup\_backimage.place(x=575,y=30)

#ALL FRAMES

#LOGIN AND SIGNUP

forgot\_password\_mainframe=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

login\_mainframe=ctk.CTkFrame(window,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_width=3,border\_color='white')

#SHOP

shop\_mainframe=ctk.CTkFrame(window,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_width=3,border\_color='white')

tabs\_frame = ctk.CTkFrame(shop\_mainframe,height=530, width=250,fg\_color='#294d61',corner\_radius=40,bg\_color='#05161a',border\_width=3,border\_color="#0f969c")

shop\_scrollframe1=ctk.CTkFrame(master=shop\_mainframe,height=440,width=745,corner\_radius=30,fg\_color='#294d61',bg\_color='#05161a',border\_width=3,border\_color="#0f969c")

shop\_widgets=ctk.CTkFrame(master=shop\_mainframe,height=70,width=740,fg\_color='#05161a',bg\_color='#05161a')

shop\_widgets\_menu=ctk.CTkFrame(master=shop\_mainframe,height=70,width=740,fg\_color='#05161a',bg\_color='#05161a')

shop\_widgets\_menu.propagate(0)

#TAB FRAMES

supplies\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

fruits\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

dairy\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

vegetable\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

search\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

profile\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

orders\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

settings\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

supports\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

notification\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

cart\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

product\_frame=ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060)

pro\_frame = ctk.CTkFrame(product\_frame, fg\_color="white",width=998,height=433,corner\_radius=30)

transit\_frame = ctk.CTkFrame(window, width=1060, height=580,corner\_radius=60, fg\_color='#05161a')

payment\_frame = ctk.CTkFrame(window,border\_width=3,fg\_color='#05161a',corner\_radius=60,height=580,width=1060,border\_color='white')

details\_frame\_pay = ctk.CTkFrame(payment\_frame, width=400, height=450, corner\_radius=20, fg\_color="white")

logo=ctk.CTkLabel(shop\_mainframe,text='',fg\_color='#05161a',image=logo\_image)

logo.place(x=160,y=20)

frame\_list=[product\_frame,tabs\_frame,signup\_mainframe,shop\_scrollframe1,shop\_widgets,shop\_widgets\_menu,supplies\_frame,fruits\_frame,dairy\_frame,vegetable\_frame,search\_frame,profile\_frame,orders\_frame,settings\_frame,supports\_frame,notification\_frame,cart\_frame,pro\_frame,transit\_frame,payment\_frame,login\_mainframe,details\_frame\_pay]

#Images

open=ctk.CTkImage(Image.open(r"open1.png"),size=(30,30))

prof=ctk.CTkImage(Image.open(r"user-interface.png"),size=(50,50))

carti=ctk.CTkImage(Image.open(r"user-interface.png"),size=(50,50))

logoutimg=ctk.CTkImage(Image.open(r"logout.png"),size=(50,50))

menu=ctk.CTkImage(Image.open(r"menu.png"),size=(60,60))

cartt=ctk.CTkImage(Image.open(r"shopcart.png"),size=(50,50))

notify=ctk.CTkImage(Image.open(r"notify.png"),size=(50,50))

seac=ctk.CTkImage(Image.open(r"search.png"),size=(40,40))

# category icons for main shop window

fru=ctk.CTkImage(Image.open(r"fruits.png"),size=(50,50))

essen=ctk.CTkImage(Image.open(r"essentials.png"),size=(50,50))

vegtab=ctk.CTkImage(Image.open(r"vegetables.png"),size=(50,50))

dair=ctk.CTkImage(Image.open(r"dairy.png"),size=(50,50))

#advertisement images for main shop window

juices = ctk.CTkImage(Image.open("juices.png"), size=(940, 160))

rice = ctk.CTkImage(Image.open("rice.png"), size=(940, 160))

cookies = ctk.CTkImage(Image.open("cookies.png"), size=(940, 160))

snacks = ctk.CTkImage(Image.open("snacks.png"), size=(940, 160))

#advertisement images for the shop window with menu button

juices\_menu= ctk.CTkImage(Image.open("juices.png"), size=(690, 160))

rice\_menu = ctk.CTkImage(Image.open("rice.png"), size=(690, 160))

cookies\_menu= ctk.CTkImage(Image.open("cookies.png"), size=(690, 160))

snacks\_menu = ctk.CTkImage(Image.open("snacks.png"), size=(690, 160))

#category icons for shop window with menu

fru2=ctk.CTkImage(Image.open(r"fruits.png"),size=(50,50))

essen2=ctk.CTkImage(Image.open(r"essentials.png"),size=(50,50))

vegtab2=ctk.CTkImage(Image.open(r"vegetables.png"),size=(50,50))

dair2=ctk.CTkImage(Image.open(r"dairy.png"),size=(50,50))

bg1=ctk.CTkImage(Image.open(r"dairy.png"),size=(500,500))

#login images

instagram=ctk.CTkImage(Image.open(r"instagram (1).png"),size=(50,50))

google=ctk.CTkImage(Image.open(r"google.png"),size=(50,50))

phon=ctk.CTkImage(Image.open(r"phone.png"),size=(50,50))

backbutn=ctk.CTkImage(Image.open(r"back-button.png"),size=(50,50))

backpic=ctk.CTkImage(Image.open(r"back-button.png"),size=(60,60))

close=ctk.CTkImage(Image.open(r"close.png"),size=(60,60))

#product images

pro1=ctk.CTkImage(Image.open(r'beetroot.png'),size=(100,100))

pro2=ctk.CTkImage(Image.open(r'carrot.png'),size=(100,100))

pro3=ctk.CTkImage(Image.open(r'pear.png'),size=(100,100))

pro4=ctk.CTkImage(Image.open(r'cheese.png'),size=(100,100))

pro5=ctk.CTkImage(Image.open(r'milk.png'),size=(100,100))

produce1=ctk.CTkImage(Image.open(r'beetroot.png'),size=(80,80))

produce2=ctk.CTkImage(Image.open(r'carrot.png'),size=(80,80))

produce3=ctk.CTkImage(Image.open(r'pear.png'),size=(80,80))

produce4=ctk.CTkImage(Image.open(r'cheese.png'),size=(80,80))

def animation():

def animate\_gif(counter=0):

gif\_label.configure(image=frames[counter])

counter +=1

if counter == len(frames):

counter = 0

loading.after(40, animate\_gif, counter)

def long\_running\_process():

time.sleep(10)

loading.place\_forget()

loading = ctk.CTkFrame(window, fg\_color='#05161a', corner\_radius=30, width=1060, height=580,border\_width=3,border\_color='white')

loading.propagate(0)

loading.place(x=30, y=30)

# Load GIF

gif = Image.open("animation.gif")

frames = [ctk.CTkImage(frame.copy(),size=(200,200)) for frame in ImageSequence.Iterator(gif)]

# Label to display GIF frames

gif\_label = ctk.CTkLabel(loading,fg\_color='#05161a',text='',height=562,width=1000,corner\_radius=30)

gif\_label.pack(pady=10)

animate\_gif()

# Start the long-running process in a separate thread

loading\_thread = Thread(target=long\_running\_process)

loading\_thread.start()

animation()

def load\_animation():

def animate\_gif(counter=0):

gif\_label.configure(image=frames[counter])

counter +=1

if counter == len(frames):

counter = 0

loading.after(40, animate\_gif, counter)

def long\_running\_process():

time.sleep(7)

loading.place\_forget()

loading = ctk.CTkFrame(window, fg\_color='#05161a', corner\_radius=30, width=1060, height=580,border\_width=3,border\_color='white')

loading.propagate(0)

loading.place(x=30, y=30)

# Load GIF

gif = Image.open("loading\_animation.gif")

frames = [ctk.CTkImage(frame.copy(),size=(200,200)) for frame in ImageSequence.Iterator(gif)] # Label to display GIF frames

gif\_label=ctk.CTkLabel(loading,fg\_color='#05161a',text='',height=562,width=1000,corner\_radius=30)

gif\_label.pack(pady=10)

animate\_gif()

# Start the long-running process in a separate thread

loading\_thread = Thread(target=long\_running\_process)

loading\_thread.start()

def back():

checked()

def sanitize\_email(email):

return email.replace('@', '\_at\_').replace('.', '\_dot\_')

def create\_user\_cart(email):

sanitized\_table\_cart = sanitize\_email(email)

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute(f'''

CREATE TABLE IF NOT EXISTS {sanitized\_table\_cart}\_cart (

product\_id VARCHAR NOT NULL,

product\_name VARCHAR NOT NULL,

price REAL NOT NULL

)

''')

except mysql.Error as e:

print(f"An error occurred: {e}")

def create\_user\_table(email):

sanitized\_table\_name = sanitize\_email(email)

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute(f'''

CREATE TABLE IF NOT EXISTS {sanitized\_table\_name} (

order\_id INTEGER PRIMARY KEY AUTOINCREMENT,

product\_id VARCHAR NOT NULL,

product\_name VARCHAR NOT NULL,

quantity INTEGER NOT NULL,

price REAL NOT NULL,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

order\_status VARCHAR NOT NULL,

payment\_status VARCHAR NOT NULL

)

''')

#messagebox.showerror('Error',f"Table {sanitized\_table\_name} created successfully.")

except mysql.Error as e:

print(f"An error occurred: {e}")

# 1. Send Welcome Email After Registration

def send\_registration\_email(email, first\_name):

subject = "Welcome to Groceria!"

body = f"Hi {first\_name},\n\nThank you for registering with Groceria! We are excited to have you.\n\nHappy Shopping!\nGroceria Team"

send\_email(email, subject, body)

# 2. Send Order Details Email After Order is Placed

def send\_order\_email(email, order\_details):

subject = "Your Groceria Order Details"

body = f"Hi,\n\nYour order has been successfully placed!\n\nOrder Details:\n{order\_details}\n\nThank you for shopping with us!\nGroceria Team"

send\_email(email, subject, body)

# 3. Send Forgot Password Email with New Password

def send\_forgot\_password\_email(email):

new\_password = ''.join(random.choices("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789", k=8))

hashed\_password = sha256(new\_password.encode()).hexdigest()

# Update password in the database

with mysql.connect("groceria.db") as conn:

cursor = conn.cursor()

cursor.execute("UPDATE users SET pass\_word = ? WHERE email\_id = ?", (hashed\_password, email))

conn.commit()

subject = "Reset Your Groceria Password"

body = f"Hi,\n\nYour password has been reset. Use the new password below to log in:\n\nNew Password: {new\_password}\n\nPlease change your password after logging in.\nGroceria Team"

send\_email(email, subject, body)

def signup\_user():

email = email\_address.get().strip()

first\_name = fullname\_entry.get().strip()

last\_name = lastname\_entry.get().strip()

password\_value = password.get().strip()

confirm\_password\_value = confirm\_password.get().strip()

if not all([email, first\_name, last\_name, password\_value, confirm\_password\_value]):

messagebox.showerror("Error", "All fields are required.")

return

if password\_value != confirm\_password\_value:

messagebox.showerror("Error", "Passwords do not match")

return

email\_regex = r"^[\w\.-]+@[\w\.-]+\.[a-zA-Z]{2,}$"

if not re.match(email\_regex, email):

messagebox.showerror("Error", "Invalid email format")

return

hashed\_password = sha256(password\_value.encode()).hexdigest()

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute("INSERT INTO users (email\_id, first\_name, last\_name, pass\_word) VALUES (?, ?, ?, ?)",

(email, first\_name, last\_name, hashed\_password))

messagebox.showinfo('Success', 'Successfully registered.')

create\_user\_table(email)

create\_user\_cart(email)

login()

send\_registration\_email(email, first\_name)

except mysql.IntegrityError:

messagebox.showerror('Error', 'Email ID is already registered.')

def login\_user():

email = email\_address\_login.get().strip()

password = password\_entry.get().strip()

if not all([email, password]):

messagebox.showerror("Error", "All fields are required.")

return

email\_regex = r"^[\w\.-]+@[\w\.-]+\.[a-zA-Z]{2,}$"

if not re.match(email\_regex, email):

messagebox.showerror("Error", "Invalid email format")

return

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute("SELECT \* FROM users WHERE email\_id = ?", (email,))

user = cursor.fetchone()

if user and sha256(password.encode()).hexdigest() == user[3]:

messagebox.showinfo("Success", "Login successful!")

checked()

create\_user\_table(email)

create\_user\_cart(email)

else:

messagebox.showerror("Error", "Invalid email or password")

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

#ALL DEFINED FUCTIONS FOR TAB

def fgpass():

load\_animation()

for i in frame\_list:

i.place\_forget()

forgot\_password\_mainframe.place(x=30,y=30)

#FORGOT PASSWORD TA forgottab\_backimage=ctk.CTkLabel(forgot\_password\_mainframe,text='',image=background\_image1)

forgottab\_backimage.place(x=575,y=30)

forgottab\_frame=ctk.CTkFrame(forgot\_password\_mainframe,height=520,width=480,bg\_color='#05161a',fg\_color='#072e33',corner\_radius=30,border\_width=3,border\_color='white')

forgottab\_frame.place(x=30,y=30)

#Widgets

email\_address=ctk.CTkEntry(forgottab\_frame,height=80,width=440,corner\_radius=30,border\_width=3, border\_color='#0f969c',fg\_color='#0c7075')

email=email\_address.get()

password=ctk.CTkButton(forgottab\_frame,text=" Reset ",height=60,width=200,corner\_radius=30,border\_width=3, border\_color='#0f969c',fg\_color='#0c7075',command=send\_forgot\_password\_email(email),font=('vivaldi italics', 20))

#labels

email\_address\_label=ctk.CTkLabel(email\_address,text='Email',text\_color='white',font=('vivaldi italics',10))

email\_address\_label.place(x=20,y=4)

#placing widgets

email\_address.place(x=20,y=140)

password.place(x=130,y=260)

#email validation

valid\_image = ctk.CTkImage(Image.open(r"valid.png"),size=(40,40))

invalid\_image = ctk.CTkImage(Image.open(r"invalid.png"),size=(40,40))

feedback\_label = ctk.CTkLabel(email\_address, text="", font=("Helvetica", 16),bg\_color='#0c7075',fg\_color='#0c7075', width=30,height=30)

feedback\_label.place(x=380,y=20)

def validate\_email(event):

email = email\_address.get()

# Regular expression to check the validity of the email including domain

email\_regex = r"^[\w\.-]+@[\w\.-]+\.[a-zA-Z]{2,}$"

valid\_length = 5 <= len(email) <= 254

if re.match(email\_regex, email) and valid\_length:

feedback\_label.configure(image=valid\_image, compound='left', fg\_color="#0c7075")

else:

feedback\_label.configure(image=invalid\_image, compound='left', fg\_color="#0c7075")

# Bind the validate\_email function to the KeyRelease event

email\_address.bind("<KeyRelease>", validate\_email)

def login():

global email\_address\_login, password\_entry # Declare as global to access in other functions

# Hide other frames and show login frame

for i in frame\_list:

i.place\_forget()

login\_mainframe.place(x=30, y=30)

# Background image for login frame

login\_backimage = ctk.CTkLabel(login\_mainframe, text='', image=background\_image1)

login\_backimage.place(x=575, y=30)

# Return to signup page function

def signup\_return():

login\_mainframe.place\_forget()

signup\_mainframe.place(x=30, y=30)

# Login frame setup

loginframe = ctk.CTkFrame(login\_mainframe, border\_width=3, border\_color='#093e44', height=520, width=480, bg\_color='#05161a', fg\_color='#072e33', corner\_radius=30)

loginframe.place(x=30, y=30)

# Buttons and labels

login\_label = ctk.CTkLabel(loginframe, text='Log In', font=('vivaldi italics', 30), text\_color='white')

login\_button = ctk.CTkButton(loginframe, text='Login', font=('vivaldi italics', 20), height=60, width=220, corner\_radius=30, fg\_color='#0c7075', hover\_color='#0c7075', text\_color='white', border\_width=3, border\_color='#0f969c', command=login\_user)

google\_button = ctk.CTkButton(loginframe, text="", image=google, fg\_color='#072e33', hover\_color='white', width=50)

phon\_button = ctk.CTkButton(loginframe, text="", image=phon, width=50, fg\_color='#072e33', hover\_color='white')

instagram\_button = ctk.CTkButton(loginframe, text="", image=instagram, width=50, fg\_color='#072e33', hover\_color='white')

forgot\_password\_btn = ctk.CTkButton(loginframe, text='Forgot password?', command=fgpass, fg\_color='#072e33', text\_color='white', hover\_color='#072e33',height=30,width=60)

signup\_return = ctk.CTkButton(loginframe, text='Create Account', fg\_color='#072e33', text\_color='white', hover\_color='#072e33', command=signup\_return)

# Entries

email\_address\_login= ctk.CTkEntry(loginframe, height=80, width=440, corner\_radius=30, border\_width=3, fg\_color='#0c7075', border\_color='#0f969c')

email\_address\_label = ctk.CTkLabel(email\_address\_login, text='Email', font=('vivaldi italics', 10), text\_color='white')

email\_address\_label.place(x=20, y=4)

password\_entry = ctk.CTkEntry(loginframe, height=80, width=210, corner\_radius=30, border\_width=3, fg\_color='#0c7075', border\_color='#0f969c')

password\_label = ctk.CTkLabel(password\_entry, text='Password', font=('vivaldi italics', 10), text\_color='white')

password\_label.place(x=20, y=4)

# Email Validation

feedback\_label\_login = ctk.CTkLabel(email\_address\_login, text="", font=("Helvetica", 16),bg\_color='#0c7075',fg\_color='#0c7075',width=30,height=30)

feedback\_label\_login.place(x=380,y=20)

def validate\_email(event):

global email\_address\_login

email = email\_address\_login.get()

# Regular expression to check the validity of the email including domain

email\_regex = r"^[\w\.-]+@[\w\.-]+\.[a-zA-Z]{2,}$"

valid\_length = 5 <= len(email) <= 254

if re.match(email\_regex, email) and valid\_length:

feedback\_label\_login.configure(image=valid\_image, compound='left', fg\_color="#0c7075")

else:

feedback\_label\_login.configure(image=invalid\_image, compound='left', fg\_color="#0c7075")

#Binding valid email event

email\_address\_login.bind("<KeyRelease>", validate\_email)

# Placing the widgets

login\_label.place(x=80, y=20)

email\_address\_login.place(x=20, y=80)

password\_entry.place(x=20, y=180)

forgot\_password\_btn.place(x=240, y=200)

login\_button.place(x=120, y=290)

google\_button.place(x=120, y=400)

instagram\_button.place(x=200, y=400)

phon\_button.place(x=280, y=400)

signup\_return.place(x=160, y=350)

def checked():

load\_animation()

for i in frame\_list:

i.place\_forget()

shop\_mainframe.place(x=30, y=30)

global searchbox,searchbox\_menu

tabs\_frame.propagate(0)

shop\_scrollframe=ctk.CTkScrollableFrame(shop\_mainframe,height=380,width=960,corner\_radius=30,fg\_color='#294d61',bg\_color='#05161a',border\_width=0,border\_color="#0f969c")

shop\_scrollframe.propagate(0)

shop\_scrollframe1.propagate(0)

shop\_scrollframe.place(x=25,y=110)

### PRODUCT DISPLAY FOR SALES ###

def show\_product\_details\_sales(product\_id, product\_name, price, discount, description, image\_path,quantity):

for frame in frame\_list:

frame.place\_forget()

product\_frame.place(x=30, y=30)

pro\_frame.place(x=30,y=110)

def back():

for i in frame\_list:

i.place\_forget()

shop\_mainframe.place(x=30,y=30)

back\_button = ctk.CTkButton(product\_frame, text="", image=backpic, command=back,fg\_color='#05161a', hover\_color='#05161a', height=70, width=70, corner\_radius=5)

back\_button.place(x=60, y=20)

product\_name\_label = ctk.CTkLabel(pro\_frame, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

try:

img = Image.open(image\_path)

product\_image = ctk.CTkImage(img, size=(250, 250))

image\_label = ctk.CTkButton(pro\_frame, text='', image=product\_image, height=280, width=280, fg\_color="white", state=DISABLED,hover\_color='white',corner\_radius=30,border\_width=3)

image\_label.place(x=50, y=75)

except FileNotFoundError:

image\_label = ctk.CTkLabel(pro\_frame, text="Image Not Available", height=220, width=220, fg\_color="white")

image\_label.place(x=50, y=50)

price\_label = ctk.CTkLabel(pro\_frame, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black")

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(pro\_frame, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black")

q\_label.place(x=500, y=170)

discount\_label = ctk.CTkLabel(pro\_frame, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red")

discount\_label.place(x=500, y=200)

ltdeal=ctk.CTkLabel(pro\_frame,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white")

ltdeal.place(x=500,y=90)

description\_label = ctk.CTkLabel(pro\_frame, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black")

description\_label.place(x=500, y=230)

add\_to\_cart\_button = ctk.CTkButton(pro\_frame, text="Add to Cart", command=lambda: add\_to\_cart(product\_id, product\_name, price), fg\_color='#0c7075', font=('Helvetica', 28),corner\_radius=20, height=50, width=200,text\_color='white')

add\_to\_cart\_button.place(x=500, y=300)

### PRODUCT DISPLAY MINI ##

def show\_product\_details\_sales\_mini(product\_id, product\_name, price, discount, description, image\_path,quantity):

for frame in frame\_list:

frame.place\_forget()

product\_frame.place(x=30, y=30)

pro\_frame.place(x=30,y=110)

def back():

for i in frame\_list:

i.place\_forget()

shop\_mainframe.place(x=30,y=30)

back\_button = ctk.CTkButton(product\_frame, text="", image=backpic, command=back,fg\_color='#05161a', hover\_color='#05161a', height=70, width=70, corner\_radius=5)

back\_button.place(x=60, y=20)

product\_name\_label = ctk.CTkLabel(pro\_frame, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

try:

img = Image.open(image\_path)

product\_image = ctk.CTkImage(img, size=(250, 250))

image\_label = ctk.CTkButton(pro\_frame, text='', image=product\_image, height=280, width=280, fg\_color="white", state=DISABLED,hover\_color='white',corner\_radius=30,border\_width=3)

image\_label.place(x=50, y=75)

except FileNotFoundError:

image\_label = ctk.CTkLabel(pro\_frame, text="Image Not Available", height=220, width=220, fg\_color="white")

image\_label.place(x=50, y=50)

price\_label = ctk.CTkLabel(pro\_frame, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black")

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(pro\_frame, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black")

q\_label.place(x=500, y=170)

discount\_label = ctk.CTkLabel(pro\_frame, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red")

discount\_label.place(x=500, y=200)

ltdeal=ctk.CTkLabel(pro\_frame,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white")

ltdeal.place(x=500,y=90)

description\_label = ctk.CTkLabel(pro\_frame, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black")

description\_label.place(x=500, y=230)

add\_to\_cart\_button = ctk.CTkButton(pro\_frame, text="Add to Cart", command=lambda: add\_to\_cart(product\_id, product\_name, price), fg\_color='#0c7075', font=('Helvetica', 28),corner\_radius=20, height=50, width=200,text\_color='white')

add\_to\_cart\_button.place(x=500, y=300)

#PRODUCT PAGE

def product\_page():

global product\_frame

for i in frame\_list:

i.place\_forget()

product\_frame.place(x=30, y=30)

tab\_namelabel = ctk.CTkLabel(product\_frame, text='Product Details', font=('Helvetica', 40), fg\_color='#05161a', text\_color='#FFFFFF')

tab\_namelabel.place(x=220, y=10)

shop\_mainframe.place(x=30, y=30) # Return to main shop frame

shop\_widgets.place(x=270, y=25)

bckbtn = ctk.CTkButton(product\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

bckbtn.place(x=40, y=10)

### PRODUCT DISPLAY FOR SUPPLIES CATEGORY ###

def show\_product\_details\_supplies(product\_id, product\_name, price, discount, description, image\_path,quantity):

# Hide other frames

for frame in frame\_list:

frame.place\_forget()

product\_frame.place(x=30, y=30)

pro\_frame.place(x=30,y=110)

# Add a back button

def back():

for i in frame\_list:

i.place\_forget()

supplies\_frame.place(x=30, y=30)

back\_button = ctk.CTkButton(product\_frame, text="", image=backpic, command=back,fg\_color='#05161a', hover\_color='#05161a', height=70, width=70, corner\_radius=5)

back\_button.place(x=60, y=20)

# Display product name

product\_name\_label = ctk.CTkLabel(pro\_frame, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

# Display product image

try:

img = Image.open(image\_path)

product\_image = ctk.CTkImage(img, size=(250, 250))

image\_label = ctk.CTkButton(pro\_frame, text='', image=product\_image, height=280, width=280, fg\_color="white", state=DISABLED,hover\_color='white',corner\_radius=30,border\_width=3)

image\_label.place(x=50, y=75)

except FileNotFoundError:

image\_label = ctk.CTkLabel(pro\_frame, text="Image Not Available", height=220, width=220, fg\_color="white")

image\_label.place(x=50, y=50)

# Display price and discount

price\_label = ctk.CTkLabel(pro\_frame, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black")

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(pro\_frame, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black")

q\_label.place(x=500, y=170)

discount\_label = ctk.CTkLabel(pro\_frame, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red")

discount\_label.place(x=500, y=200)

ltdeal=ctk.CTkLabel(pro\_frame,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white")

ltdeal.place(x=500,y=90)

# Display description

description\_label = ctk.CTkLabel(pro\_frame, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black")

description\_label.place(x=500, y=230)

# Add-to-cart button

add\_to\_cart\_button = ctk.CTkButton(pro\_frame, text="Add to Cart", command=lambda: add\_to\_cart(product\_id, product\_name, price), fg\_color='#0c7075', font=('Helvetica', 28),corner\_radius=20, height=50, width=200,text\_color='white')

add\_to\_cart\_button.place(x=500, y=300)

### PRODUCT DISPLAY FOR FRUITS CATEGORY ###

def show\_product\_details\_fruits(product\_id, product\_name, price, discount, description, image\_path,quantity):

# Hide other frames

for frame in frame\_list:

frame.place\_forget()

product\_frame.place(x=30, y=30)

pro\_frame.place(x=30,y=110)

# Add a back button

def back():

for i in frame\_list:

i.place\_forget()

fruits\_frame.place(x=30, y=30)

back\_button = ctk.CTkButton(product\_frame, text="", image=backpic, command=back,fg\_color='#05161a', hover\_color='#05161a', height=70, width=70, corner\_radius=5)

back\_button.place(x=60, y=20)

# Display product name

product\_name\_label = ctk.CTkLabel(pro\_frame, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

# Display product image

try:

img = Image.open(image\_path)

product\_image = ctk.CTkImage(img, size=(250, 250))

image\_label = ctk.CTkButton(pro\_frame, text='', image=product\_image, height=280, width=280, fg\_color="white", state=DISABLED,hover\_color='white',corner\_radius=30,border\_width=3)

image\_label.place(x=50, y=75)

except FileNotFoundError:

image\_label = ctk.CTkLabel(pro\_frame, text="Image Not Available", height=220, width=220, fg\_color="white")

image\_label.place(x=50, y=50)

# Display price and discount

price\_label = ctk.CTkLabel(pro\_frame, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black")

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(pro\_frame, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black")

q\_label.place(x=500, y=170)

discount\_label = ctk.CTkLabel(pro\_frame, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red")

discount\_label.place(x=500, y=200)

ltdeal=ctk.CTkLabel(pro\_frame,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white")

ltdeal.place(x=500,y=90)

# Display description

description\_label = ctk.CTkLabel(pro\_frame, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black")

description\_label.place(x=500, y=230)

# Add-to-cart button

add\_to\_cart\_button = ctk.CTkButton(pro\_frame, text="Add to Cart", command=lambda: add\_to\_cart(product\_id, product\_name, price), fg\_color='#0c7075', font=('Helvetica', 28),corner\_radius=20, height=50, width=200,text\_color='white')

add\_to\_cart\_button.place(x=500, y=300)

### PRODUCT DISPLAY FOR VEGGIES CATEGORY ###

def show\_product\_details\_veggies(product\_id, product\_name, price, discount, description, image\_path,quantity):

# Hide other frames

for frame in frame\_list:

frame.place\_forget()

product\_frame.place(x=30, y=30)

pro\_frame.place(x=30,y=110)

# Add a back button

def back():

for i in frame\_list:

i.place\_forget()

vegetable\_frame.place(x=30, y=30)

back\_button = ctk.CTkButton(product\_frame, text="", image=backpic, command=back,fg\_color='#05161a', hover\_color='#05161a', height=70, width=70, corner\_radius=5)

back\_button.place(x=60, y=20)

# Display product name

product\_name\_label = ctk.CTkLabel(pro\_frame, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

# Display product image

try:

img = Image.open(image\_path)

product\_image = ctk.CTkImage(img, size=(250, 250))

image\_label = ctk.CTkButton(pro\_frame, text='', image=product\_image, height=280, width=280, fg\_color="white", state=DISABLED,hover\_color='white',corner\_radius=30,border\_width=3)

image\_label.place(x=50, y=75)

except FileNotFoundError:

image\_label = ctk.CTkLabel(pro\_frame, text="Image Not Available", height=220, width=220, fg\_color="white")

image\_label.place(x=50, y=50)

# Display price and discount

price\_label = ctk.CTkLabel(pro\_frame, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black")

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(pro\_frame, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black")

q\_label.place(x=500, y=170)

discount\_label = ctk.CTkLabel(pro\_frame, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red")

discount\_label.place(x=500, y=200)

ltdeal=ctk.CTkLabel(pro\_frame,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white")

ltdeal.place(x=500,y=90)

# Display description

description\_label = ctk.CTkLabel(pro\_frame, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black")

description\_label.place(x=500, y=230)

# Add-to-cart button

add\_to\_cart\_button = ctk.CTkButton(pro\_frame, text="Add to Cart", command=lambda: add\_to\_cart(product\_id, product\_name, price), fg\_color='#0c7075', font=('Helvetica', 28),corner\_radius=20, height=50, width=200,text\_color='white')

add\_to\_cart\_button.place(x=500, y=300)

### PRODUCT DISPLAY FOR DAIRY CATEGORY ###

def show\_product\_details\_dairy(product\_id, product\_name, price, discount, description, image\_path,quantity):

# Hide other frames

for frame in frame\_list:

frame.place\_forget()

product\_frame.place(x=30, y=30)

pro\_frame.place(x=30,y=110)

# Add a back button

def back():

for i in frame\_list:

i.place\_forget()

dairy\_frame.place(x=30, y=30)

back\_button = ctk.CTkButton(product\_frame, text="", image=backpic, command=back,fg\_color='#05161a', hover\_color='#05161a', height=70, width=70, corner\_radius=5)

back\_button.place(x=60, y=20)

# Display product name

product\_name\_label = ctk.CTkLabel(pro\_frame, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

# Display product image

try:

img = Image.open(image\_path)

product\_image = ctk.CTkImage(img, size=(250, 250))

image\_label = ctk.CTkButton(pro\_frame, text='', image=product\_image, height=280, width=280, fg\_color="white", state=DISABLED,hover\_color='white',corner\_radius=30,border\_width=3)

image\_label.place(x=50, y=75)

except FileNotFoundError:

image\_label = ctk.CTkLabel(pro\_frame, text="Image Not Available", height=220, width=220, fg\_color="white")

image\_label.place(x=50, y=50)

# Display price and discount

price\_label = ctk.CTkLabel(pro\_frame, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black")

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(pro\_frame, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black")

q\_label.place(x=500, y=170)

discount\_label = ctk.CTkLabel(pro\_frame, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red")

discount\_label.place(x=500, y=200)

ltdeal=ctk.CTkLabel(pro\_frame,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white")

ltdeal.place(x=500,y=90)

# Display description

description\_label = ctk.CTkLabel(pro\_frame, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black")

description\_label.place(x=500, y=230)

# Add-to-cart button

add\_to\_cart\_button = ctk.CTkButton(pro\_frame, text="Add to Cart", command=lambda: add\_to\_cart(product\_id, product\_name, price), fg\_color='#0c7075', font=('Helvetica', 28),corner\_radius=20, height=50, width=200,text\_color='white')

add\_to\_cart\_button.place(x=500, y=300)

def supplie\_products():

# Hide other frames

for frame in frame\_list:

frame.place\_forget()

# Display the supplies\_frame for the Supplies tab

supplies\_frame.place(x=30, y=30)

tab\_namelabel = ctk.CTkLabel(supplies\_frame, text='Supplies', font=('Helvetica', 40), fg\_color='#05161a', text\_color='#FFFFFF')

tab\_namelabel.place(x=220, y=10)

# Back button setup

def back():

for frame in frame\_list:

frame.place\_forget()

shop\_mainframe.place(x=30, y=30)

shop\_widgets.place(x=270, y=25)

bckbtn = ctk.CTkButton(supplies\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

logo = ctk.CTkButton(supplies\_frame, image=logo\_image\_one, text='', fg\_color='#05161a', bg\_color='#05161a', hover\_color='#05161a', command=checked, height=60, width=70)

logo.place(x=120, y=3)

bckbtn.place(x=40, y=10)

# Scrollable frame for displaying products with dynamically adjusted height

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute("SELECT product\_id, product\_name, price, discount,description,quantity FROM supplies")

products = cursor.fetchall()

positions = [(10, 10), (480, 10), (10, 210), (480, 210), (10, 410), (480, 410),(10, 610), (480, 610), (10, 810), (480, 810)][:len(products)]

product\_height = 180 # Height of each product frame

scroll\_frame\_height = ((product\_height )\*2)+40 # Adjust height with padding

scroll\_frame = ctk.CTkScrollableFrame(supplies\_frame, width=960, height=scroll\_frame\_height, fg\_color='#05161a', corner\_radius=40, border\_width=0)

scroll\_frame.place(x=20, y=70)

backlbl= ctk.CTkLabel(scroll\_frame, text='', font=('Helvetica', 40), width=900, height=1000, fg\_color='#05161a', corner\_radius=20)

backlbl.pack()

for i, product in enumerate(products):

product\_id, product\_name, price, discount,description,quantity = product

# Product display frame

product\_label = ctk.CTkLabel(scroll\_frame, text='', font=('Helvetica', 40), width=440, height=180, fg\_color='white', corner\_radius=20)

product\_label.place(x=positions[i][0], y=positions[i][1])

try:

image\_path = f"images/{product\_id}.png" # Assuming each image is named by product\_id

img = Image.open(image\_path)

ctk\_image = ctk.CTkImage(img, size=(100, 100))

image\_button = ctk.CTkButton(product\_label, text='', image=ctk\_image, height=140, width=140, corner\_radius=20, fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white',command=lambda p\_id=product\_id, p\_name=product\_name, p\_price=price, p\_discount=discount, p\_desc=description,p\_quant=quantity: show\_product\_details\_supplies(p\_id, p\_name, p\_price, p\_discount, p\_desc, f"images/{p\_id}.png",p\_quant))

image\_button.place(x=20, y=20)

except FileNotFoundError:

print("Image file not found:", image\_path)

image\_button = ctk.CTkButton(product\_label, text='No Image', height=140, width=140, corner\_radius=20, fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white')

image\_button.place(x=20, y=20)

info\_label = ctk.CTkFrame(product\_label, height=140, width=240, corner\_radius=20, fg\_color='#ffffff', border\_width=3)

info\_label.place(x=180, y=20)

product\_name\_label = ctk.CTkLabel(info\_label, text=f"{product\_name}", font=('Helvetica', 22))

product\_name\_label.place(x=10, y=10)

price\_label = ctk.CTkLabel(info\_label, text=f"₹{price}", font=('Helvetica', 22))

price\_label.place(x=20, y=40)

discount\_label = ctk.CTkLabel(info\_label, text=f"{discount}% Off", font=('Helvetica', 22),text\_color='red')

discount\_label.place(x=20, y=70)

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

def fruit\_products():

# Hide other frames

for frame in frame\_list:

frame.place\_forget()

fruits\_frame.place(x=30, y=30)

tab\_namelabel = ctk.CTkLabel(fruits\_frame, text='Fruits', font=('Helvetica', 40), fg\_color='#05161a', text\_color='#FFFFFF')

tab\_namelabel.place(x=220, y=10)

# Back button setup

def back():

for frame in frame\_list:

frame.place\_forget()

shop\_mainframe.place(x=30, y=30)

shop\_widgets.place(x=270, y=25)

bckbtn = ctk.CTkButton(fruits\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

logo = ctk.CTkButton(fruits\_frame, image=logo\_image\_one, text='', fg\_color='#05161a', bg\_color='#05161a', hover\_color='#05161a', command=checked, height=60, width=70)

logo.place(x=120, y=3)

bckbtn.place(x=40, y=10)

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute("SELECT product\_id, product\_name, price, discount,description,quantity FROM fruits")

products = cursor.fetchall()

positions = [(10, 10), (480, 10), (10, 210), (480, 210), (10, 410), (480, 410),(10, 610), (480, 610), (10, 810), (480, 810)][:len(products)]

product\_height = 180 # Height of each product frame

scroll\_frame\_height = ((product\_height )\*2)+40 # Adjust height with padding

scroll\_frame = ctk.CTkScrollableFrame(fruits\_frame, width=960, height=scroll\_frame\_height, fg\_color='#05161a', corner\_radius=40, border\_width=0)

scroll\_frame.place(x=20, y=70)

backlbl= ctk.CTkLabel(scroll\_frame, text='', font=('Helvetica', 40), width=900, height=1000, fg\_color='#05161a', corner\_radius=20)

backlbl.pack()

for i, product in enumerate(products):

product\_id, product\_name, price, discount,description,quantity = product

# Product display frame

product\_label = ctk.CTkLabel(scroll\_frame, text='', font=('Helvetica', 40), width=440, height=180, fg\_color='white', corner\_radius=20)

product\_label.place(x=positions[i][0], y=positions[i][1])

try:

image\_path = f"images/{product\_id}.png" # Assuming each image is named by product\_id

img = Image.open(image\_path)

ctk\_image = ctk.CTkImage(img, size=(100, 100))

image\_button = ctk.CTkButton(product\_label, text='', image=ctk\_image, height=140, width=140, corner\_radius=20, fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white',command=lambda p\_id=product\_id, p\_name=product\_name, p\_price=price, p\_discount=discount, p\_desc=description,p\_quant=quantity: show\_product\_details\_fruits(p\_id, p\_name, p\_price, p\_discount, p\_desc, f"images/{p\_id}.png",p\_quant))

image\_button.place(x=20, y=20)

except FileNotFoundError:

print("Image file not found:", image\_path)

image\_button = ctk.CTkButton(product\_label, text='No Image', height=140, width=140, corner\_radius=20, fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white')

image\_button.place(x=20, y=20)

info\_label = ctk.CTkFrame(product\_label, height=140, width=240, corner\_radius=20, fg\_color='#ffffff', border\_width=3)

info\_label.place(x=180, y=20)

product\_name\_label = ctk.CTkLabel(info\_label, text=f"{product\_name}", font=('Helvetica', 22))

product\_name\_label.place(x=10, y=10)

price\_label = ctk.CTkLabel(info\_label, text=f"₹{price}", font=('Helvetica', 22))

price\_label.place(x=20, y=40)

discount\_label = ctk.CTkLabel(info\_label, text=f"{discount}% Off", font=('Helvetica', 22),text\_color='red')

discount\_label.place(x=20, y=70)

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

def dairy\_products():

# Hide other frames

for frame in frame\_list:

frame.place\_forget()

# Display the supplies\_frame for the Supplies tab

dairy\_frame.place(x=30, y=30)

tab\_namelabel = ctk.CTkLabel(dairy\_frame, text='Dairy', font=('Helvetica', 40), fg\_color='#05161a', text\_color='#FFFFFF')

tab\_namelabel.place(x=220, y=10)

# Back button setup

def back():

for frame in frame\_list:

frame.place\_forget()

shop\_mainframe.place(x=30, y=30)

shop\_widgets.place(x=270, y=25)

bckbtn = ctk.CTkButton(dairy\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

logo = ctk.CTkButton(dairy\_frame, image=logo\_image\_one, text='', fg\_color='#05161a', bg\_color='#05161a', hover\_color='#05161a', command=checked, height=60, width=70)

logo.place(x=120, y=3)

bckbtn.place(x=40, y=10)

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute("SELECT product\_id, product\_name, price, discount,description,quantity FROM dairy")

products = cursor.fetchall()

positions = [(10, 10), (480, 10), (10, 210), (480, 210), (10, 410), (480, 410),(10, 610), (480, 610), (10, 810), (480, 810)][:len(products)]

product\_height = 180 # Height of each product frame

scroll\_frame\_height = ((product\_height )\*2)+40 # Adjust height with padding

scroll\_frame = ctk.CTkScrollableFrame(dairy\_frame, width=960, height=scroll\_frame\_height, fg\_color='#05161a', corner\_radius=40, border\_width=0)

scroll\_frame.place(x=20, y=70)

backlbl= ctk.CTkLabel(scroll\_frame, text='', font=('Helvetica', 40), width=900, height=1000, fg\_color='#05161a', corner\_radius=20)

backlbl.pack()

for i, product in enumerate(products):

product\_id, product\_name, price, discount,description,quantity = product

# Product display frame

product\_label = ctk.CTkLabel(scroll\_frame, text='', font=('Helvetica', 40), width=440, height=180, fg\_color='white', corner\_radius=20)

product\_label.place(x=positions[i][0], y=positions[i][1])

info\_label = ctk.CTkFrame(product\_label, height=140, width=240, corner\_radius=20, fg\_color='#ffffff', border\_width=3)

info\_label.place(x=180, y=20)

try:

image\_path = f"images/{product\_id}.png" # Assuming each image is named by product\_id

img = Image.open(image\_path)

ctk\_image = ctk.CTkImage(img, size=(100, 100))

image\_button = ctk.CTkButton(product\_label, text='', image=ctk\_image, height=140, width=140, corner\_radius=20, fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white',command=lambda p\_id=product\_id, p\_name=product\_name, p\_price=price, p\_discount=discount, p\_desc=description,p\_quant=quantity: show\_product\_details\_dairy(p\_id, p\_name, p\_price, p\_discount, p\_desc, f"images/{p\_id}.png",p\_quant))

image\_button.place(x=20, y=20)

except FileNotFoundError:

print("Image file not found:", image\_path)

image\_button = ctk.CTkButton(product\_label, text='No Image', height=140, width=140, corner\_radius=20, fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white')

image\_button.place(x=20, y=20)

product\_name\_label = ctk.CTkLabel(info\_label, text=f"{product\_name}", font=('Helvetica', 22))

product\_name\_label.place(x=10, y=10)

price\_label = ctk.CTkLabel(info\_label, text=f"₹{price}", font=('Helvetica', 22))

price\_label.place(x=20, y=40)

discount\_label = ctk.CTkLabel(info\_label, text=f"{discount}% Off", font=('Helvetica', 22),text\_color='red')

discount\_label.place(x=20, y=70)

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

def vegetable\_products():

# Hide other frames

for frame in frame\_list:

frame.place\_forget()

# Display the supplies\_frame for the Supplies tab

vegetable\_frame.place(x=30, y=30)

tab\_namelabel = ctk.CTkLabel(vegetable\_frame, text='Vegetables', font=('Helvetica', 40), fg\_color='#05161a', text\_color='#FFFFFF')

tab\_namelabel.place(x=220, y=10)

# Back button setup

def back():

for frame in frame\_list:

frame.place\_forget()

shop\_mainframe.place(x=30, y=30)

shop\_widgets.place(x=270, y=25)

bckbtn = ctk.CTkButton(vegetable\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

logo = ctk.CTkButton(vegetable\_frame, image=logo\_image\_one, text='', fg\_color='#05161a', bg\_color='#05161a', hover\_color='#05161a', command=checked, height=60, width=70)

logo.place(x=120, y=3)

bckbtn.place(x=40, y=10)

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

cursor.execute("SELECT product\_id, product\_name, price, discount,description,quantity FROM veges")

products = cursor.fetchall()

positions = [(10, 10), (480, 10), (10, 210), (480, 210), (10, 410), (480, 410),(10, 610), (480, 610), (10, 810), (480, 810)][:len(products)]

product\_height = 180 # Height of each product frame

scroll\_frame\_height = ((product\_height )\*2)+40 # Adjust height with padding

scroll\_frame = ctk.CTkScrollableFrame(vegetable\_frame, width=960, height=scroll\_frame\_height, fg\_color='#05161a', corner\_radius=40, border\_width=0)

scroll\_frame.place(x=20, y=70)

backlbl= ctk.CTkLabel(scroll\_frame, text='', font=('Helvetica', 40), width=900, height=1000, fg\_color='#05161a', corner\_radius=20)

backlbl.pack()

for i, product in enumerate(products):

product\_id, product\_name, price, discount,description,quantity = product

# Product display frame

product\_label = ctk.CTkLabel(scroll\_frame, text='', font=('Helvetica', 40), width=440, height=180, fg\_color='white', corner\_radius=20)

product\_label.place(x=positions[i][0], y=positions[i][1])

try:

image\_path = f"images/{product\_id}.png" # Assuming each image is named by product\_id

img = Image.open(image\_path)

ctk\_image = ctk.CTkImage(img, size=(100, 100))

image\_button = ctk.CTkButton(product\_label, text='', image=ctk\_image, height=140, width=140, corner\_radius=20, fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white',command=lambda p\_id=product\_id, p\_name=product\_name, p\_price=price, p\_discount=discount, p\_desc=description,p\_quant=quantity: show\_product\_details\_veggies(p\_id, p\_name, p\_price, p\_discount, p\_desc, f"images/{p\_id}.png",p\_quant))

image\_button.place(x=20, y=20)

except FileNotFoundError:

print("Image file not found:", image\_path)

image\_button = ctk.CTkButton(product\_label, text='No Image', height=140, width=140, corner\_radius=20, fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white')

image\_button.place(x=20, y=20)

info\_label = ctk.CTkFrame(product\_label, height=140, width=240, corner\_radius=20, fg\_color='#ffffff', border\_width=3)

info\_label.place(x=180, y=20)

product\_name\_label = ctk.CTkLabel(info\_label, text=f"{product\_name}", font=('Helvetica', 22))

product\_name\_label.place(x=10, y=10)

price\_label = ctk.CTkLabel(info\_label, text=f"₹{price}", font=('Helvetica', 22))

price\_label.place(x=20, y=40)

discount\_label = ctk.CTkLabel(info\_label, text=f"{discount}% Off", font=('Helvetica', 22),text\_color='red')

discount\_label.place(x=20, y=70)

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

#widgets in scrollable frame

# Advertisement Frame

show = ctk.CTkLabel(shop\_scrollframe, width=960, height=700, fg\_color='#294d61',text='')

show.grid(column=0,row=0)

advertisement = ctk.CTkLabel(shop\_scrollframe, width=960, height=200, fg\_color='#294d61',text='')

advertisement.place(x=-10,y=0)

#CATEGORY TABS

category\_frame=ctk.CTkFrame(shop\_scrollframe,height=160,width=850,corner\_radius=30,fg\_color='#05161a',border\_width=3,border\_color='#0f969c')

products\_frame=ctk.CTkFrame(shop\_scrollframe,height=340,width=920,corner\_radius=30,fg\_color='#05161a',border\_width=3,border\_color='#0f969c')

category\_label= ctk.CTkLabel(category\_frame,text\_color='white', width=100, height=60, fg\_color='#05161a',text='Categories:',font=fnt1)

sale\_label= ctk.CTkLabel(shop\_scrollframe,text\_color='white', width=100, height=60, fg\_color='#05161a',text='Grab the Trending Deals now !!',font=fnt1)

vegetables=ctk.CTkButton(category\_frame,text='Veges',image=vegtab,height=70,width=200,fg\_color='#0c7075',corner\_radius=20,font=fnt1,border\_width=3,border\_color='#0f969c',command=vegetable\_products)

fruits=ctk.CTkButton(category\_frame,text='Fruits',image=fru,height=70,width=200,fg\_color='#0c7075',corner\_radius=20,font=fnt1,border\_width=3,border\_color='#0f969c',command=fruit\_products)

dairy=ctk.CTkButton(category\_frame,text='Dairy',height=70,image=dair,width=200,fg\_color='#0c7075',corner\_radius=20,font=fnt1,border\_width=3,border\_color='#0f969c',command=dairy\_products)

essentials=ctk.CTkButton(category\_frame,text='Supplies',image=essen,height=70,width=200,fg\_color='#0c7075',corner\_radius=20,font=fnt1,border\_width=3,border\_color='#0f969c',command=supplie\_products)

product\_labels = []

x\_positions = [25, 203, 381, 559, 737]

# Create product labels

for x\_pos in x\_positions:

product\_label = ctk.CTkLabel(products\_frame, text='', height=240, width=158, fg\_color='white', corner\_radius=20)

product\_label.place(x=x\_pos, y=80)

product\_labels.append(product\_label)

# Fetch product data from the database

try:

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute("SELECT product\_id, product\_name, price, discount, description, quantity FROM sales LIMIT 5")

products = cursor.fetchall()

for index, (product\_label, product) in enumerate(zip(product\_labels, products)):

product\_id, product\_name, price, discount, description, quantity = product

# Load product image

try:

image\_path = f"images/{product\_id}.png"

img = Image.open(image\_path)

ctk\_image = ctk.CTkImage(img, size=(100, 100))

image\_button = ctk.CTkButton(product\_label,text='',height=120,width=120,corner\_radius=20,fg\_color='white',font=('Helvetica', 50),border\_width=3,image=ctk\_image,command=lambda p\_id=product\_id, p\_name=product\_name, p\_price=price, p\_discount=discount, p\_desc=description, p\_quant=quantity: show\_product\_details\_sales(p\_id, p\_name, p\_price, p\_discount, p\_desc, image\_path, p\_quant))

image\_button.place(x=10, y=10) # Positioning inside product label

except FileNotFoundError:

print(f"Image not found: {image\_path}")

image\_button = ctk.CTkButton(product\_label,text='No Image',height=120,width=120,corner\_radius=20,fg\_color='white',border\_width=3)

image\_button.place(x=10, y=10)

# Create info label for price and discount

info\_label = ctk.CTkFrame(product\_label, height=90, width=140, corner\_radius=20, fg\_color='#ffffff', border\_width=3)

info\_label.place(x=10, y=140) # Positioning inside product label

# Add price and discount text

price\_label = ctk.CTkLabel(info\_label, text=f"₹{price}", font=('Helvetica', 16))

price\_label.pack(padx=10, pady=5)

if discount:

discount\_label = ctk.CTkLabel(info\_label, text=f"{discount}% Off", font=('Helvetica', 14), text\_color='red')

discount\_label.pack(padx=10, pady=5)

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

#placing of tabs

category\_frame.place(x=50,y=180)

sale\_label.place(x=50,y=375)

products\_frame.place(x=10,y=360)

category\_label.place(x=30,y=10)

dairy.place(x=430,y=70)

fruits.place(x=220,y=70)

vegetables.place(x=10,y=70)

essentials.place(x=640,y=70)

#scrollable frame with menu button on screen

advertisement\_menu= ctk.CTkLabel(shop\_scrollframe1, width=700, height=300, fg\_color='#294d61',text='')

advertisement\_menu.place(x=20,y=20)

category\_frame\_menu=ctk.CTkFrame(tabs\_frame,height=150,width=210,corner\_radius=30,fg\_color='#05161a',border\_width=3,border\_color='#0f969c')

products\_frame\_menu=ctk.CTkFrame(shop\_scrollframe1,height=210,width=675,corner\_radius=30,fg\_color='#05161a',border\_width=3,border\_color='#0f969c')

#CATEGORY TABS

vegetables=ctk.CTkButton(category\_frame\_menu,text='',image=vegtab2,height=50,width=50,fg\_color='#0c7075',corner\_radius=20,font=fnt,border\_width=3,border\_color='#0f969c',command=vegetable\_products)

fruits=ctk.CTkButton(category\_frame\_menu,text='',image=fru2,height=50,width=50,fg\_color='#0c7075',corner\_radius=20,font=fnt,border\_width=3,border\_color='#0f969c',command=fruit\_products)

dairy=ctk.CTkButton(category\_frame\_menu,text='',height=50,image=dair2,width=50,fg\_color='#0c7075',corner\_radius=20,font=fnt,border\_width=3,border\_color='#0f969c',command=dairy\_products)

essentials=ctk.CTkButton(category\_frame\_menu,text='',image=essen2,height=50,width=50,fg\_color='#0c7075',corner\_radius=20,font=fnt,border\_width=3,border\_color='#0f969c',command=supplie\_products)

product\_labels = []

x\_positions = [25, 185, 345, 505]

for x\_pos in x\_positions:

product\_label = ctk.CTkLabel(products\_frame\_menu, text='', height=190, width=140, fg\_color='white', corner\_radius=20)

product\_label.place(x=x\_pos, y=10)

product\_labels.append(product\_label)

# Fetch product data from the database

try:

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute("SELECT product\_id, product\_name, price, discount, description, quantity FROM sales LIMIT 5")

products = cursor.fetchall()

for index, (product\_label, product) in enumerate(zip(product\_labels, products)):

product\_id, product\_name, price, discount, description, quantity = product

# Load product image

try:

image\_path = f"images/{product\_id}.png"

img = Image.open(image\_path)

ctk\_image = ctk.CTkImage(img, size=(80, 80))

image\_button = ctk.CTkButton(product\_label,text='',height=90,width=90,corner\_radius=20,fg\_color='white',font=('Helvetica', 50),border\_width=3,image=ctk\_image,command=lambda p\_id=product\_id, p\_name=product\_name, p\_price=price, p\_discount=discount, p\_desc=description, p\_quant=quantity: show\_product\_details\_sales\_mini(p\_id, p\_name, p\_price, p\_discount, p\_desc, image\_path, p\_quant))

image\_button.place(x=10, y=10) # Positioning inside product label

except FileNotFoundError:

print(f"Image not found: {image\_path}")

image\_button = ctk.CTkButton(product\_label,text='No Image',height=120,width=120,corner\_radius=20,fg\_color='white',border\_width=3)

image\_button.place(x=10, y=10)

# Create info label for price and discount

info\_label = ctk.CTkFrame(product\_label, height=90, width=140, corner\_radius=20, fg\_color='#ffffff', border\_width=3)

info\_label.place(x=10, y=105) # Positioning inside product label

# Add price and discount text

price\_label = ctk.CTkLabel(info\_label, text=f"₹{price}", font=('Helvetica', 16))

price\_label.pack(padx=10, pady=5)

if discount:

discount\_label = ctk.CTkLabel(info\_label, text=f"{discount}% Off", font=('Helvetica', 14), text\_color='red')

discount\_label.pack(padx=10, pady=5)

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

#placing tabs and widgets

# category\_frame\_menu.place(x=20,y=280)

fruits.place(x=110,y=10)

dairy.place(x=10,y=80)

vegetables.place(x=10,y=10)

essentials.place(x=110,y=80)

def confirm\_order(order\_details):

try:

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

s\_mail = email\_address\_login.get()

sanitized\_table\_name = sanitize\_email(s\_mail) + "\_orders"

order\_id = random.randint(10000, 99999)

cursor.execute(f'''

CREATE TABLE IF NOT EXISTS {sanitized\_table\_name} (

order\_id INTEGER PRIMARY KEY AUTOINCREMENT,

product\_id VARCHAR NOT NULL,

product\_name VARCHAR NOT NULL,

quantity INTEGER NOT NULL,

price REAL NOT NULL,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

order\_status VARCHAR NOT NULL,

payment\_status VARCHAR NOT NULL)''')

cursor.execute(f'''

INSERT INTO {sanitized\_table\_name} (order\_id, product\_id, product\_name, quantity, price, order\_status, payment\_status)

VALUES (?, ?, ?, ?, ?, ?, ?)

''', (

order\_id,

order\_details['product\_id'],

order\_details['product\_name'],

order\_details['quantity'],

order\_details['total'], # Total price as `price`

"To be Shipped",

"Due"

))

conn.commit()

messagebox.showinfo(

"Order Confirmation",

f"Your order for {order\_details['quantity']} units of {order\_details['product\_name']} "

f"has been placed successfully!\n\nOrder Status: To be Shipped soon")

cart\_tab()

send\_order\_email(email, order\_details)

except mysql.Error as e:

messagebox.showerror("Database Error", str(e))

messagebox.showerroe("Debug",sanitized\_email\_name)

def payments\_page(order\_details):

for i in frame\_list:

i.place\_forget() # Hide all frames

payment\_frame.place(x=30, y=30)

heading\_label = ctk.CTkLabel(payment\_frame,text="Payment",font=("Helvetica", 40),fg\_color='#05161a',text\_color="#FFFFFF")

heading\_label.place(x=440, y=10)

bill=ctk.CTkLabel(details\_frame\_pay,text="Bill",font=('Helvetica',28),fg\_color='white')

bill.place(x=180,y=20)

div=ctk.CTkFrame(details\_frame\_pay,width=340,height=2,border\_color='black',border\_width=2,fg\_color='black',bg\_color='black')

div.place(x=30,y=60)

divl=ctk.CTkFrame(details\_frame\_pay,width=340,height=2,border\_color='black',border\_width=2,fg\_color='black',bg\_color='black')

divl.place(x=30,y=360)

back\_button = ctk.CTkButton(payment\_frame,text="",command=lambda: transit\_tab(order\_details), image=backbutn,fg\_color='#05161a',hover\_color='#05161a',height=55,width=55,corner\_radius=5)

back\_button.place(x=40, y=10)

logo = ctk.CTkButton(payment\_frame,image=logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

qr=ctk.CTkImage(Image.open(r"qr.png"),size=(280,280))

qr\_label=ctk.CTkLabel(payment\_frame,text="",image=qr,corner\_radius=20,fg\_color='white')

qr\_label.place(x=650,y=150)

logo.place(x=120, y=3)

details\_frame\_pay.place(x=80, y=80)

product\_name = order\_details["product\_name"]

price = order\_details["price"]

quantity = order\_details["quantity"]

total= order\_details["total"]

price\_label = ctk.CTkLabel(details\_frame\_pay, text=f"{quantity} × {product\_name} ₹{price}", font=('Helvetica', 20), text\_color="black",fg\_color="white")

price\_label.place(x=100, y=80)

gst = round((total \* 12) / 100, 2) # Using round for better precision

net = total + gst

gst\_label = ctk.CTkLabel(details\_frame\_pay, text=f"Gst - ₹{gst}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black",fg\_color="white")

gst\_label.place(x=100, y=120)

net\_label = ctk.CTkLabel(details\_frame\_pay, text=f"Total - {net}", font=('Helvetica', 24), wraplength=500, justify="left", text\_color="black",fg\_color="white")

net\_label.place(x=100, y=370)

print(f"pid:{product\_id}")

order\_detail= {"product\_id":product\_id,"product\_name": product\_name,"price": price,"quantity": quantity,"total": net}

confirm\_button = ctk.CTkButton(payment\_frame,text="Confirm Order",font=("Helvetica", 22),corner\_radius=20,fg\_color="#0c7075",text\_color="#FFFFFF",command=lambda: confirm\_order(order\_detail),height=50,width=80)

confirm\_button.place(x=720, y=450)

def transit\_tab(product):

for i in frame\_list:

i.place\_forget() # Hide all frames

transit\_frame.place(x=30, y=30)

print(ui)

product\_frame\_exe=ctk.CTkFrame(transit\_frame,border\_width=3,fg\_color='white',corner\_radius=60,height=450,width=1000)

product\_frame\_exe.place(x=32, y=90)

tab\_namelabel = ctk.CTkLabel(transit\_frame,text='Order Summary',font=('Helvetica', 40),fg\_color='#05161a',text\_color='#FFFFFF')

tab\_namelabel.place(x=300, y=10)

bckbtn = ctk.CTkButton(transit\_frame,text="",command=cart\_tab,image=backbutn,fg\_color='#05161a',hover\_color='#05161a',height=55,width=55,corner\_radius=5)

bckbtn.place(x=40, y=10)

logo = ctk.CTkButton(transit\_frame,image=logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

logo.place(x=120, y=3)

try:

image\_path = f"images/{product\_id}.png"

img = Image.open(image\_path)

product\_image = ctk.CTkImage(img, size=(100, 100))

image\_label = ctk.CTkButton(product\_frame\_exe, image=product\_image, text="",corner\_radius=30,border\_width=3,state=DISABLED,height=280,width=280,hover\_color='white',fg\_color='white')

image\_label.image = product\_image

image\_label.place(x=30, y=50)

print(f"ni{product\_id}")

except FileNotFoundError:

image\_button = ctk.CTkButton(product\_label,text='No Image',height=140,width=140,corner\_radius=20,fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white')

image\_button.place(x=20, y=20)

product\_name = product["product\_name"]

price = product["price"]

available\_quantity = product["quantity"]

description = product["description"]

discount = product.get("discount", None)

product\_name\_label = ctk.CTkLabel(product\_frame\_exe, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

price\_label = ctk.CTkLabel(product\_frame\_exe, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black",fg\_color="white")

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(product\_frame\_exe, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black",fg\_color="white")

q\_label.place(x=500, y=170)

ltdeal=ctk.CTkLabel(product\_frame\_exe,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white")

ltdeal.place(x=500,y=90)

description\_label = ctk.CTkLabel(product\_frame\_exe, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black",fg\_color="white")

description\_label.place(x=500, y=230)

if discount:

discount\_label = ctk.CTkLabel(product\_frame\_exe, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red",fg\_color="white")

discount\_label.place(x=500, y=200)

quantity\_input = ctk.CTkEntry(product\_frame\_exe, placeholder\_text="Enter Quantity", width=120,height=50,corner\_radius=20,border\_width=3)

quantity\_input.place(x=500, y=300)

def proceed\_to\_payment():

try:

quantity = quantity\_input.get().strip() # Get and strip any extra spaces

if not quantity.isdigit():

messagebox.showerror("Error" ,"Please enter a valid numeric quantity.")

quantity = int(quantity\_input.get())

order\_details= {"product\_id":product\_id,"product\_name": product\_name,"price": price,"quantity": quantity,"total": price \* quantity}

payments\_page(order\_details)

except ValueError as e:

messagebox.showerror("Input Error", str(e))

order\_now\_button = ctk.CTkButton(product\_frame\_exe,text="Order Now", command=proceed\_to\_payment,font=("Helvetica", 28),corner\_radius=20,fg\_color="#0c7075",text\_color="#FFFFFF",height=50)

order\_now\_button.place(x=640, y=300)

#DEFINING THE TABS

## NESTED SEARCH BOX ##

def search\_box(nested\_searchbox, search\_frame, email\_id):

search\_term = nested\_searchbox.get().strip()

for i in frame\_list:

i.place\_forget()

search\_frame.place(x=30,y=30)

# Clear previous results

tables = ["veges", "fruits", "dairy", "supplies"]

found\_products = []

# Connect to the database

connection = mysql.connect("groceria.db")

cursor = connection.cursor()

try:

for table in tables:

cursor.execute(f"SELECT \* FROM {table} WHERE product\_name LIKE ?", ('%' + search\_term + '%',))

products = cursor.fetchall()

found\_products.extend(products)

if found\_products:

for product in found\_products:

product\_id, exp\_date, price, quantity, discount, product\_name, description = product

# Display product information

product\_frame = ctk.CTkFrame(search\_frame, fg\_color="white", width=980, height=430,corner\_radius=30)

product\_frame.place(x=40,y=90)

logo=ctk.CTkButton(search\_frame,image= logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

logo.place(x=120,y=3)

# Display image if available

try:

image\_path = f"images/{product\_id}.png"

img = Image.open(image\_path)

product\_image =ctk.CTkImage(img, size=(250, 250))

image\_label = ctk.CTkButton(product\_frame, image=product\_image, text="",corner\_radius=30,border\_width=3,state=DISABLED,height=280,width=280,hover\_color='white',fg\_color='white')

image\_label.image = product\_image

image\_label.place(x=30, y=50)

except FileNotFoundError:

print(f"Image for {product\_name} not found at {image\_path}")

# Display details

product\_name\_label = ctk.CTkLabel(product\_frame, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

price\_label = ctk.CTkLabel(product\_frame, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black",fg\_color="white")

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(product\_frame, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black",fg\_color="white")

q\_label.place(x=500, y=170)

discount\_label = ctk.CTkLabel(product\_frame, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red",fg\_color="white")

discount\_label.place(x=500, y=200)

ltdeal=ctk.CTkLabel(product\_frame,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white")

ltdeal.place(x=500,y=90)

description\_label = ctk.CTkLabel(product\_frame, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black",fg\_color="white")

description\_label.place(x=500, y=230)

add\_to\_cart\_button = ctk.CTkButton(product\_frame, text="Add to Cart", command=lambda: add\_to\_cart(product\_id, product\_name, price), fg\_color='#0c7075', font=('Helvetica', 28),corner\_radius=20, height=50, width=200,text\_color='white')

add\_to\_cart\_button.place(x=500, y=300)

else:

no\_product\_label = ctk.CTkLabel(search\_frame, text="Product not found", font=("Helvetica", 16))

no\_product\_label.pack(pady=10)

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

def add\_to\_cart( product\_id,product\_name,price):

try:

conn= mysql.connect("groceria.db")

cursor = conn.cursor()

s\_mail=email\_address\_login.get()

user\_id=sanitize\_email(s\_mail)

table\_name = f"{user\_id}\_cart"

cursor.execute(f"""

CREATE TABLE IF NOT EXISTS {table\_name} (

product\_id INTEGER PRIMARY KEY,

product\_name VARCHAR,

price REAL

)

""")

try:

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

cursor.execute(f'''

INSERT INTO {table\_name} (product\_id, product\_name, price)

VALUES (?, ?, ?)

''', (product\_id, product\_name, price))

messagebox.showinfo("Cart", f"{product\_name} added to cart!")

except mysql.Error as e:

messagebox.showerror("Database Error", f"Could not add to cart: {e}")

except mysql.Error as e:

messagebox.showerroe("Error",f"Database Error: {e}")

return f"An error occurred: {e}"

#SEARCH TAB

def search\_tab(searchbox, search\_frame, email\_id):

global nested\_searchbox

search\_term = searchbox.get().strip()

for i in frame\_list:

i.place\_forget()

search\_frame.place(x=30,y=30)

def back():

for frame in frame\_list:

search\_frame.place\_forget()

shop\_mainframe.place(x=30, y=30)

shop\_widgets.place(x=270, y=25)

# Clear previous results

tables = ["veges", "fruits", "dairy", "supplies"]

found\_products = []

# Connect to the database

connection = mysql.connect("groceria.db")

cursor = connection.cursor()

try:

for table in tables:

cursor.execute(f"SELECT \* FROM {table} WHERE product\_name LIKE ?", ('%' + search\_term + '%',))

products = cursor.fetchall()

found\_products.extend(products)

if found\_products:

for product in found\_products:

product\_id, exp\_date, price, quantity, discount, product\_name, description = product

# Display product information

product\_frame = ctk.CTkFrame(search\_frame, fg\_color="white", width=980, height=430,corner\_radius=30)

product\_frame.place(x=40,y=90)

logo=ctk.CTkButton(search\_frame,image= logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

logo.place(x=120,y=8)

bckbtn = ctk.CTkButton(search\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

bckbtn.place(x=40,y=15)

nested\_searchbox=ctk.CTkEntry(search\_frame,fg\_color='white',width=500,height=60,corner\_radius=20,border\_width=3,bg\_color='#05161a',border\_color='#0f969c')

nested\_searchbox.place(x=240,y=10)

search\_button=ctk.CTkButton(nested\_searchbox,text='',image=seac,fg\_color='#05161a',width=50,height=30,corner\_radius=20,border\_width=3,bg\_color='#ffffff',border\_color='#0f969c',command=lambda: search\_box(nested\_searchbox, search\_frame, email\_id))

search\_button.place(x=415,y=5)

# Display image if available

try:

image\_path = f"images/{product\_id}.png"

img = Image.open(image\_path)

product\_image =ctk.CTkImage(img, size=(250, 250))

image\_label = ctk.CTkButton(product\_frame, image=product\_image, text="",corner\_radius=30,border\_width=3,state=DISABLED,height=280,width=280,hover\_color='white',fg\_color='white')

image\_label.image = product\_image

image\_label.place(x=30, y=50)

except FileNotFoundError:

print(f"Image for {product\_name} not found at {image\_path}")

# Display details

product\_name\_label = ctk.CTkLabel(product\_frame, text=product\_name, font=('Helvetica', 45), text\_color="black", fg\_color="white")

product\_name\_label.place(x=520, y=30)

price\_label = ctk.CTkLabel(product\_frame, text=f" ₹{price}", font=('Helvetica', 38), text\_color="black",bg\_color='white',fg\_color='white')

price\_label.place(x=500, y=120)

q\_label = ctk.CTkLabel(product\_frame, text=f"Pack Of {quantity}", font=('Helvetica', 20), text\_color="black",bg\_color='white',fg\_color='white')

q\_label.place(x=500, y=170)

discount\_label = ctk.CTkLabel(product\_frame, text=f" -{discount}% M.R.P", font=('Helvetica', 22), text\_color="red",bg\_color='white',fg\_color='white')

discount\_label.place(x=500, y=200)

ltdeal=ctk.CTkLabel(product\_frame,text="Limited time deal",font=('Helvetica',20),fg\_color='red',corner\_radius=10,height=20,width=80,text\_color="white",bg\_color='white')

ltdeal.place(x=500,y=90)

description\_label = ctk.CTkLabel(product\_frame, text=f"Product details:\n {description}", font=('Helvetica', 20), wraplength=500, justify="left", text\_color="black",bg\_color='white',fg\_color='white')

description\_label.place(x=500, y=230)

add\_to\_cart\_button = ctk.CTkButton(product\_frame, text="Add to Cart", command=lambda: add\_to\_cart(product\_id, product\_name, price), fg\_color='#0c7075', font=('Helvetica', 28),corner\_radius=20, height=50, width=200,text\_color='white',bg\_color='white')

add\_to\_cart\_button.place(x=500, y=300)

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

def order(product, email\_id):

product\_id, exp\_date, price, quantity, discount, product\_name, description = product

connection = mysql.connect("groceria.db")

cursor = connection.cursor()

sanitized\_table\_name = email.id.replace('@', '\_at\_').replace('.', '\_dot\_')

try:

cursor.execute(f"INSERT INTO {sanitized\_table\_name} (order\_id, product\_name, quantity, price) VALUES (?, ?, ?, ?)",

(order\_id, product\_name, quantity, price))

connection.commit()

messagebox.showinfo("Order Placed Successfully")

except mysql.Error as e:

messagebox.showerror("Database Error", f"An error occurred: {e}")

finally:

connection.close()

#PROFILE TAB

def profile\_tab():

for i in frame\_list:

i.place\_forget()

profile\_frame.place(x=30,y=30)

try:

connection = mysql.connect("groceria.db")

cursor = connection.cursor()

user\_id = email\_address\_login.get() # Example: Replace with the logged-in user# SQL query to fetch user de

query = """SELECT first\_name, last\_name, email\_id, address, pincode

FROM users

WHERE email\_id = ?"""

cursor.execute(query, (user\_id,))

user\_data = cursor.fetchone()

if user\_data:

first\_name, last\_name, email, address, pincode = user\_data

except mysql.Error as e:

print(f"An error occurred: {e}")

#widgets in profile tab

logo=ctk.CTkButton(profile\_frame,image= logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

logo.place(x=120,y=3)

tab\_namelabel=ctk.CTkLabel(profile\_frame,text='Profile',font=('Helvetica',40),fg\_color='#05161a',text\_color='#FFFFFF')

profile\_tab\_one=ctk.CTkFrame(profile\_frame,fg\_color='#294d61',height=480,width=600,corner\_radius=50,border\_width=3,border\_color="#0f969c")

profile\_tab\_two=ctk.CTkFrame(profile\_frame,fg\_color='#294d61',height=480,width=380,corner\_radius=50,border\_width=3,border\_color="#0f969c")

prof\_image=ctk.CTkButton(profile\_tab\_one,text='',corner\_radius=50,image=prof,fg\_color='#ffffff',border\_width=3,width=200,height=200)

edit\_profile=ctk.CTkButton(profile\_tab\_one,text='Edit Profile',corner\_radius=50,fg\_color='#0c7075',border\_width=3,width=400,height=60,font=fnt1)

delete\_button=ctk.CTkButton(profile\_tab\_two,text='Delete Acoount',font=fnt1,border\_width=4,border\_color='#811331',fg\_color='#0c7075',height=60,width=280,corner\_radius=30)

terms\_button=ctk.CTkButton(profile\_tab\_two,text='Terms & Conditions ',font=fnt1,border\_width=3,fg\_color='#0c7075',height=60,width=280,corner\_radius=30)

about\_button=ctk.CTkButton(profile\_tab\_two,text='About Us',font=fnt1,border\_width=3,fg\_color='#0c7075',height=60,width=280,corner\_radius=30)

payments=ctk.CTkButton(profile\_tab\_two,text='Payments',font=fnt1,border\_width=3,fg\_color='#0c7075',height=60,width=280,corner\_radius=30)

payments.place(x=20,y=180)

#Labels

fullname\_label=ctk.CTkLabel(profile\_tab\_one,text=first\_name,text\_color='white',font=('vivaldi italics',20))

lastname\_label=ctk.CTkLabel(profile\_tab\_one,text=last\_name,text\_color='white',font=('vivaldi italics',20))

email\_address\_label=ctk.CTkLabel(profile\_tab\_one,text=email,text\_color='white',font=('vivaldi italics',20))

address=ctk.CTkLabel(profile\_tab\_one,text=address,text\_color='white',font=('vivaldi italics',20))

pincode=ctk.CTkLabel(profile\_tab\_one,text=pincode,text\_color='white',font=('vivaldi italics',20))

features=ctk.CTkLabel(profile\_tab\_two,text="Options",font=("Helvetica",30))

features.place(x=60,y=30)

divider=ctk.CTkFrame(profile\_tab\_two,height=1,width=300,border\_width=4)

divider.place(x=40,y=80)

bckbtn = ctk.CTkButton(profile\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

#placing widgets

profile\_tab\_one.place(x=30,y=70)

profile\_tab\_two.place(x=650,y=70)

tab\_namelabel.place(x=220, y=10)

delete\_button.place(x=20,y=360)

terms\_button.place(x=20,y=100)

about\_button.place(x=20,y=260)

prof\_image.place(x=40,y=40)

edit\_profile.place(x=100,y=400)

bckbtn.place(x=40, y=10)

lastname\_label.place(x=450,y=40)

email\_address\_label.place(x=300,y=120)

fullname\_label.place(x=300,y=40)

address.place(x=300,y=200)

pincode.place(x=300,y=280)

# ORDERS TAB

def orders\_tab():

for i in frame\_list:

i.place\_forget()

orders\_frame.place(x=30,y=30)

tab\_namelabel=ctk.CTkLabel(orders\_frame,text='Orders',font=('Helvetica',40),fg\_color='#05161a',text\_color='#FFFFFF')

tab\_namelabel.place(x=220, y=10)

bckbtn = ctk.CTkButton(orders\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

logo=ctk.CTkButton(orders\_frame,image= logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

logo.place(x=120,y=3)

bckbtn.place(x=40, y=10)

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

s\_mail = email\_address\_login.get()

user\_id = sanitize\_email(s\_mail)

table\_name = f"{user\_id}"+"\_orders"

query = f"SELECT \* FROM {table\_name}"

cursor.execute(query)

orders = cursor.fetchall()

for i, order in enumerate(orders):

order\_id, product\_id,product\_name, quantity, price, order\_date, order\_status,payment\_status = order

print(f"Order ID: {order\_id}, Product: {product\_id}, Price: {price}, Order Date: {order\_date}")

if not products:

ctk.CTkLabel(cart\_frame,text="Your cart is empty!",font=('Helvetica', 22),text\_color='#FFFFFF').place(x=50, y=100)

return

product\_height = 180 # Height of each product frame

scroll\_frame\_height = ((product\_height) \* 2) + 40 # Adjust height with padd

scroll\_frame = ctk.CTkScrollableFrame(orders\_frame,width=960,height=scroll\_frame\_height,fg\_color='#05161a',corner\_radius=40,border\_width=3)

scroll\_frame.place(x=20, y=70)

positions = [(10, 10), (10, 210), (10, 410),(10, 610), (10, 810), (10, 1010), (10, 1210), (10, 1410), (10, 1610), (10, 1810), (10, 2010), (10, 2210), (10, 2410), (10, 2610), (10, 2810), (10, 3010), (10, 3210), (10, 3410), (10, 3610), (10, 3810)][:len(products)]

backlbl = ctk.CTkLabel(scroll\_frame,text='',font=('Helvetica', 40),width=900,height=4000,fg\_color='#05161a',corner\_radius=20)

backlbl.pack()

product\_label = ctk.CTkLabel(scroll\_frame,text='',font=('Helvetica', 40),width=900,height=180,fg\_color='white',corner\_radius=20)

product\_label.place(x=positions[i][0], y=positions[i][1])

image\_path = f"images/{product\_id}.png"

img = Image.open(image\_path)

ctk\_image = ctk.CTkImage(img, size=(100, 100))

image\_button = ctk.CTkButton(product\_label,text='',image=ctk\_image,height=140,width=140,corner\_radius=20,fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white')

image\_button.place(x=20, y=20)

info\_label = ctk.CTkFrame(product\_label,height=140,width=620,corner\_radius=20,fg\_color='#ffffff',border\_width=3)

info\_label.place(x=180, y=20)

product\_name\_label = ctk.CTkLabel(info\_label,text=f"Name: {product\_name}",font=('Helvetica', 22))

product\_name\_label.place(x=10, y=10)

price\_label = ctk.CTkLabel(info\_label,text=f"Price: ₹{price}",font=('Helvetica', 22))

price\_label.place(x=10, y=40)

quantity\_label = ctk.CTkLabel(info\_label,text=f"Quantity: {quantity}",font=('Helvetica', 22))

quantity\_label.place(x=10, y=70)

description\_label = ctk.CTkLabel(info\_label,text=f"Ordered On: {order\_date}",font=('Helvetica', 16),wraplength=550,justify='left')

description\_label.place(x=10, y=100)

if discount:

discount\_label = ctk.CTkLabel(info\_label,text=f"Discount: {discount}%",font=('Helvetica', 22),text\_color='red')

discount\_label.place(x=400, y=10)

except mysql.Error as e:

messagebox.showerror("Database Error", str(e))

#CATEGORIES BUTTON

def cat\_function\_one():

category\_frame\_menu.place(x=20,y=280)

def cat\_function\_two():

category\_frame\_menu.place\_forget()

def category\_button():

global cat\_click\_count

cat\_click\_count += 1

if cat\_click\_count == 1:

cat\_function\_one()

elif cat\_click\_count == 2:

cat\_function\_two()

cat\_click\_count = 0

#SUPPORTS TAB

def supports\_tab():

for i in frame\_list:

i.place\_forget()

supports\_frame.place(x=30,y=30)

tab\_namelabel=ctk.CTkLabel(supports\_frame,text='Contact Us',font=('Helvetica',40),fg\_color='#05161a',text\_color='#FFFFFF')

tab\_namelabel.place(x=430, y=20)

bckbtn = ctk.CTkButton(supports\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

logo=ctk.CTkButton(supports\_frame,image= logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

help\_namelabel=ctk.CTkLabel(supports\_frame,text="We're here to help.",font=('Helvetica',35),fg\_color='#05161a',text\_color='grey')

help\_namelabel.place(x=390, y=100)

query\_namelabel=ctk.CTkLabel(supports\_frame,text="Have any queries ?",font=('Helvetica',20),fg\_color='#05161a',text\_color='grey')

query\_namelabel.place(x=450, y=80)

divider=ctk.CTkFrame(supports\_frame,width=700,height=3,border\_width=0,border\_color='grey',fg\_color='grey',corner\_radius=20)

divider.place(x=215,y=145)

help\_sales=ctk.CTkLabel(supports\_frame,text='Sales',fg\_color='#0c7075',bg\_color='#05161a',height=150,width=240,corner\_radius=20,text\_color='white',font=fnt1)

help\_returns=ctk.CTkLabel(supports\_frame,text='Returns',fg\_color='#0c7075',bg\_color='#05161a',height=150,width=240,corner\_radius=20,text\_color='white',font=fnt1)

help\_complaints=ctk.CTkLabel(supports\_frame,text='Complaints',fg\_color='#0c7075',bg\_color='#05161a',height=150,width=240,corner\_radius=20,text\_color='white',font=fnt1)

help\_marketing=ctk.CTkLabel(supports\_frame,text='Marketing',fg\_color='#0c7075',bg\_color='#05161a',height=150,width=240,corner\_radius=20,text\_color='white',font=fnt1)

help\_sales.place(x=35,y=180)

help\_returns.place(x=285,y=180)

help\_complaints.place(x=535,y=180)

help\_marketing.place(x=785,y=180)

logo.place(x=120,y=3)

bckbtn.place(x=40, y=10)

#NOTIFICATION TAB

def notification\_tab():

for i in frame\_list:

i.place\_forget()

notification\_frame.place(x=30,y=30)

tab\_namelabel=ctk.CTkLabel(notification\_frame,text='Notifications',font=('Helvetica',40),fg\_color='#072e33',text\_color='#FFFFFF')

tab\_namelabel.place(x=220, y=10)

bckbtn = ctk.CTkButton(notification\_frame, text="", command=back, image=backbutn, fg\_color='#05161a', hover\_color='#05161a', height=55, width=55, corner\_radius=5)

logo=ctk.CTkButton(notification\_frame,image= logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

logo.place(x=120,y=3)

bckbtn.place(x=40, y=10)

scroll\_frame = ctk.CTkScrollableFrame(notification\_frame, height=400, width=960, fg\_color='#05161a', corner\_radius=40, border\_width=3)

scroll\_frame.place(x=20, y=70)

shop\_scrollframe1=ctk.CTkLabel(scroll\_frame,text='',width=100,height=1200)

shop\_scrollframe1.pack()

#CART TAB

def cart\_tab():

global ui

for i in frame\_list:

i.place\_forget()

cart\_frame.place(x=30, y=30)

tab\_namelabel = ctk.CTkLabel(cart\_frame, text='Cart', font=('Helvetica', 40),fg\_color='#05161a',text\_color='#FFFFFF' )

tab\_namelabel.place(x=220, y=10)

bckbtn = ctk.CTkButton(cart\_frame,text="",command=back,image=backbutn,fg\_color='#05161a',hover\_color='#05161a',height=55,width=55,corner\_radius=5)

logo = ctk.CTkButton(cart\_frame,image=logo\_image\_one,text='',fg\_color='#05161a',bg\_color='#05161a',hover\_color='#05161a',command=checked,height=60,width=70)

logo.place(x=120, y=3)

bckbtn.place(x=40, y=10)

with mysql.connect('groceria.db') as conn:

cursor = conn.cursor()

try:

s\_mail = email\_address\_login.get()

user\_id = sanitize\_email(s\_mail)

table\_name = f"{user\_id}\_cart"

query = f"SELECT product\_id, product\_name, price FROM {table\_name}"

cursor.execute(query)

products = cursor.fetchall()

if not products:

ctk.CTkLabel(cart\_frame,text="Your cart is empty!",font=('Helvetica', 22),text\_color='#FFFFFF').place(x=50, y=100)

return

product\_height = 180 # Height of each product frame

scroll\_frame\_height = ((product\_height) \* 2) + 40 # Adjust height with padd

scroll\_frame = ctk.CTkScrollableFrame(cart\_frame,width=960,height=scroll\_frame\_height,fg\_color='#05161a',corner\_radius=40,border\_width=0)

scroll\_frame.place(x=20, y=70)

positions = [(10, 10), (10, 210), (10, 410),(10, 610), (10, 810), (10, 1010), (10, 1210), (10, 1410), (10, 1610), (10, 1810), (10, 2010), (10, 2210), (10, 2410), (10, 2610), (10, 2810), (10, 3010), (10, 3210), (10, 3410), (10, 3610), (10, 3810)][:len(products)]

backlbl = ctk.CTkLabel(scroll\_frame,text='',font=('Helvetica', 40),width=900,height=4000,fg\_color='#05161a',corner\_radius=20)

backlbl.pack()

tables = ["veges", "fruits", "dairy", "supplies"]

for i, product in enumerate(products):

product\_id, product\_name, price = product

product\_details = None

for table in tables:

cursor.execute(f"SELECT description, quantity, discount FROM {table} WHERE product\_id = ?",(product\_id,))

product\_details = cursor.fetchone()

if product\_details:

break

if not product\_details:

continue # Skip products not found in any category table

description, db\_quantity, discount = product\_details

product\_label = ctk.CTkLabel(scroll\_frame,text='',font=('Helvetica', 40),width=900,height=180,fg\_color='white',corner\_radius=20)

product\_label.place(x=positions[i][0], y=positions[i][1])

try:

image\_path = f"images/{product\_id}.png"

img = Image.open(image\_path)

ctk\_image = ctk.CTkImage(img, size=(100, 100))

image\_button = ctk.CTkButton(product\_label,text='',image=ctk\_image,height=140,width=140,corner\_radius=20,fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white',command=lambda p={"product\_id":product\_id,

"product\_name": product\_name,

"price": price,

"quantity": quantity,

"description": description,

"discount": discount

}: transit\_tab(p))

image\_button.place(x=20, y=20)

print(f"ui{product\_id}")

ui=product\_id

except FileNotFoundError:

image\_button = ctk.CTkButton(product\_label,text='No Image',height=140,width=140,corner\_radius=20,fg\_color='white',border\_width=3,bg\_color='white',hover\_color='white')

image\_button.place(x=20, y=20)

info\_label = ctk.CTkFrame(product\_label,height=140,width=620,corner\_radius=20,fg\_color='#ffffff',border\_width=3)

info\_label.place(x=180, y=20)

product\_name\_label = ctk.CTkLabel(info\_label,text=f"Name: {product\_name}",font=('Helvetica', 22))

product\_name\_label.place(x=10, y=10)

price\_label = ctk.CTkLabel(info\_label,text=f"Price: ₹{price}",font=('Helvetica', 22))

price\_label.place(x=10, y=40)

quantity\_label = ctk.CTkLabel(info\_label,text=f"Quantity: {quantity}",font=('Helvetica', 22))

quantity\_label.place(x=10, y=70)

description\_label = ctk.CTkLabel(info\_label,text=f"Description: {description}",font=('Helvetica', 16),wraplength=550,justify='left')

print(f"cid:{product\_id}")

if discount:

discount\_label = ctk.CTkLabel(info\_label,text=f"Discount: {discount}%",font=('Helvetica', 22),text\_color='red')

discount\_label.place(x=10, y=100)

except mysql.Error as e:

messagebox.showerror("Database Error", str(e))

#widgets on shop without menu

searchbox=ctk.CTkEntry(shop\_widgets,fg\_color='white',width=500,height=60,corner\_radius=20,border\_width=3,bg\_color='#05161a',border\_color='#0f969c')

notification\_button=ctk.CTkButton(shop\_widgets,text='',image=notify,fg\_color='#05161a',width=50,height=50,corner\_radius=20,border\_width=0,bg\_color='#05161a',command=notification\_tab)

cart\_button=ctk.CTkButton(shop\_widgets,text='',image=cartt,fg\_color='#0c7075',width=50,height=50,corner\_radius=20,border\_width=3,bg\_color='#05161a',command=cart\_tab)

search\_button=ctk.CTkButton(shop\_widgets,text='',image=seac,fg\_color='#05161a',width=50,height=30,corner\_radius=20,border\_width=3,bg\_color='#ffffff',border\_color='#0f969c',command=lambda: search\_tab(searchbox, search\_frame, email\_id))

#placing the widgets without menu

shop\_widgets.place(x=270,y=25)

notification\_button.place(x=630,y=5)

cart\_button.place(x=520,y=5)

search\_button.place(x=420,y=10)

searchbox.place(x=5,y=5)

#widgets in the menu

orders = ctk.CTkButton(tabs\_frame, text="Orders ",image=open, fg\_color="#05161a", font=fnt1, corner\_radius=20, width=210, height=40,border\_width=3,compound='right',text\_color='white',command=orders\_tab,border\_color="#0f969c")

profile = ctk.CTkButton(tabs\_frame, text=" Hi user ", fg\_color="#ffffff", font=fnt1, corner\_radius=20, width=210, height=100,border\_width=3,compound='left',text\_color='black',image=prof,command=profile\_tab)

hpc = ctk.CTkButton(tabs\_frame, text="Contact Us ", fg\_color="#05161a", font=fnt1, corner\_radius=20, width=210, height=40,image=open,border\_width=3,text\_color='white',compound='right',command=supports\_tab,border\_color="#0f969c")

category= ctk.CTkButton(tabs\_frame, text="Categories ", fg\_color="#05161a", font=fnt1, corner\_radius=20, width=210, height=40,image=open,border\_width=3,text\_color='white',compound='right',command=category\_button,border\_color="#0f969c")

logout= ctk.CTkButton(tabs\_frame, text="", fg\_color="#0f969c", font=fnt1, corner\_radius=20, width=210, height=40, command=login,border\_width=3,image=logoutimg,text\_color='black')

# placing tab buttons

profile.place(x=20, y=20)

orders.place(x=20, y=130)

hpc.place(x=20, y=180)

category.place(x=20, y=230)

logout.place(x=20,y=450)

#widgets with menu

menu\_searchbox=ctk.CTkEntry(shop\_widgets\_menu,fg\_color='white',width=420,height=60,corner\_radius=20,border\_width=3,bg\_color='#05161a',border\_color="#0f969c")

menu\_notification\_button=ctk.CTkButton(shop\_widgets\_menu,text='',image=notify,fg\_color='#05161a',width=50,height=50,corner\_radius=20,border\_width=0,bg\_color='#05161a',command=notification\_tab)

menu\_cart\_button=ctk.CTkButton(shop\_widgets\_menu,text='',image=cartt,fg\_color='#0c7075',width=50,height=50,corner\_radius=20,border\_width=3,bg\_color='#05161a',command=cart\_tab)

menu\_search\_button=ctk.CTkButton(shop\_widgets\_menu,text='',image=seac,fg\_color='#05161a',width=50,height=30,corner\_radius=20,border\_width=3,bg\_color='#ffffff',border\_color='#0f969c',command=search\_tab)

#placing widgets with menu

menu\_notification\_button.place(x=630,y=5)

menu\_cart\_button.place(x=530,y=5)

menu\_search\_button.place(x=435,y=10)

menu\_searchbox.place(x=100,y=5)

#MENU BUTTON FUNCTIONING AND DEFINING

def function\_one():

shop\_scrollframe.place\_forget()

shop\_scrollframe1.place(x=290,y=110)

tabs\_frame.place(x=25,y=25)

products\_frame\_menu.place(x=35,y=205)

#withdrawing the widgets without menu

menubtn.place\_forget()

logo.place\_forget()

shop\_widgets.place\_forget()

#placing the widgets with menu

shop\_widgets\_menu.place(x=285,y=25)

def function\_two():

menubtn.place(x=50,y=30)

tabs\_frame.place\_forget()

shop\_scrollframe1.place\_forget()

shop\_scrollframe.place(x=25,y=110)

#placing the widgets without menu

logo.place(x=160,y=20)

shop\_widgets.place(x=270,y=25)

#withdrawing the widgets without menu

shop\_widgets\_menu.place\_forget()

def click():

global click\_count

click\_count += 1

if click\_count == 1:

function\_one()

elif click\_count == 2:

function\_two()

click\_count = 0

menubtn=ctk.CTkButton(shop\_mainframe,text='',image=menu,fg\_color='#05161a',command=click,corner\_radius=20,height=40, width=40,hover\_color='#05161a',border\_width=0,bg\_color='#05161a')

menubtn.place(x=50,y=30)

close\_menu=ctk.CTkButton(shop\_widgets\_menu,text='',height=40,width=60,command=click,image=close,fg\_color='#05171a')

close\_menu.place(x=20,y=5)

# ADVERTISEMENT FUNCTION FOR SHOP WINDOW WITH MENU

#image list

canvas\_menu = ctk.CTkCanvas(advertisement\_menu, width=720, height=160)

canvas\_menu.configure(bg='#294d61')

canvas\_menu.place(x=0, y=0)

images\_menu = [juices\_menu, rice\_menu, cookies\_menu, snacks\_menu]

commands = [fruit\_products, vegetable\_products, supplie\_products, dairy\_products]

# Create buttons for each image with corresponding commands

buttons\_menu= [ctk.CTkButton(canvas\_menu, image=image, width=720, fg\_color='#294d61',height=150, command=command,text='') for image, command in zip(images\_menu, commands)]

# Function to animate the sliding effect

def slide\_images(canvas\_menu, buttons\_menu, duration=25, steps=30):

button\_iter = itertools.cycle(buttons\_menu) # Create an iterator that cycles through the buttons

current\_button = next(button\_iter)

window\_id = canvas\_menu.create\_window(0, 0, anchor='nw', window=current\_button)

canvas\_menu.update()

def animate():

nonlocal current\_button, window\_id

next\_button = next(button\_iter)

next\_window\_id = canvas\_menu.create\_window(canvas\_menu.winfo\_width(), 0, anchor='nw', window=next\_button)

canvas\_menu.update()

for step in range(steps + 1):

offset = step / steps \* canvas\_menu.winfo\_width()

canvas\_menu.coords(window\_id, -offset, 0)

canvas\_menu.coords(next\_window\_id, canvas\_menu.winfo\_width() - offset, 0)

canvas\_menu.update()

canvas\_menu.after(duration // steps)

canvas\_menu.delete(window\_id)

current\_button = next\_button

window\_id = next\_window\_id

canvas\_menu.after(3000, animate) # Schedule the next animation after 3000 milliseconds (3 seconds)

animate()

slide\_images(canvas\_menu, buttons\_menu)

#ADVERTISEMENT FUCNTION FOR SHOP WINDOW WITHOUT MENU

#Image list

# Create a canvas to hold the images

canvas = ctk.CTkCanvas(advertisement, width=960, height=160)

canvas.configure(bg='#294d61')

canvas.place(x=0, y=0)

images = [juices, rice, cookies, snacks]

commands = [fruit\_products, vegetable\_products, supplie\_products, dairy\_products]

# Create buttons for each image with corresponding commands

buttons = [ctk.CTkButton(canvas, image=image,text='' ,width=960, fg\_color='#294d61',height=150, command=command) for image, command in zip(images, commands)]

# Function to animate the sliding effect

def slide\_images(canvas, buttons, duration=25, steps=30):

button\_iter = itertools.cycle(buttons) # Create an iterator that cycles through the buttons

current\_button = next(button\_iter)

window\_id = canvas.create\_window(0, 0, anchor='nw', window=current\_button)

canvas.update()

def animate():

nonlocal current\_button, window\_id

next\_button = next(button\_iter)

next\_window\_id = canvas.create\_window(canvas.winfo\_width(), 0, anchor='nw', window=next\_button)

canvas.update()

for step in range(steps + 1):

offset = step / steps \* canvas.winfo\_width()

canvas.coords(window\_id, -offset, 0)

canvas.coords(next\_window\_id, canvas.winfo\_width() - offset, 0)

canvas.update()

canvas.after(duration // steps)

canvas.delete(window\_id)

current\_button = next\_button

window\_id = next\_window\_id

canvas.after(3000, animate) # Schedule the next animation after 3000 milliseconds (3 seconds)

animate()

slide\_images(canvas, buttons)

#SIGN UP PAGE SETUP

click\_count=0

widgetframe\_su=ctk.CTkFrame(signup\_mainframe,height=520,width=480,fg\_color='#072e33',bg\_color='#05161a',corner\_radius=30,border\_width=3,border\_color='#093e44')

#Frame for widgets

signup\_label=ctk.CTkLabel(widgetframe\_su,text='Create Account !',font=('vivaldi italics',25),text\_color='white')

#entry boxes

fullname\_entry=ctk.CTkEntry(widgetframe\_su,height=80,width=210,corner\_radius=30,border\_width=3,border\_color='#0f969c',fg\_color='#0c7075')

lastname\_entry=ctk.CTkEntry(widgetframe\_su,height=80,width=210,corner\_radius=30,border\_width=3,border\_color='#0f969c',fg\_color='#0c7075')

confirm\_password=ctk.CTkEntry(widgetframe\_su,height=80,width=210,corner\_radius=30,border\_width=3,border\_color='#0f969c',fg\_color='#0c7075')

email\_address=ctk.CTkEntry(widgetframe\_su,height=80,width=440,corner\_radius=30,border\_width=3,border\_color='#0f969c',fg\_color='#0c7075')

password=ctk.CTkEntry(widgetframe\_su,height=80,width=210,corner\_radius=30,border\_width=3,border\_color='#0f969c',fg\_color='#0c7075')

email\_id = email\_address.get()

email=email\_address.get()

#labels

fullname\_label=ctk.CTkLabel(fullname\_entry,text='First name',text\_color='white',font=('vivaldi italics',10))

fullname\_label.place(x=20,y=4)

lastname\_label=ctk.CTkLabel(lastname\_entry,text='Last name',text\_color='white',font=('vivaldi italics',10))

lastname\_label.place(x=20,y=4)

confirm\_password\_label=ctk.CTkLabel(confirm\_password,text='Confirm Password',text\_color='white',font=('vivaldi italics',10))

confirm\_password\_label.place(x=20,y=4)

email\_address\_label=ctk.CTkLabel(email\_address,text='Email',text\_color='white',font=('vivaldi italics',10))

email\_address\_label.place(x=20,y=4)

password\_label=ctk.CTkLabel(password,text='Password',text\_color='white',font=('vivaldi italics',10))

password\_label.place(x=20,y=4)

existing\_user=ctk.CTkButton(widgetframe\_su,text='Existing User ? Login',font=('vivaldi italics',15),text\_color='white',hover\_color='#072e33',fg\_color='#072e33',command=login)

existing\_user.place(x=150,y=390)

#placing entry boxes

widgetframe\_su.place(x=30,y=30)

lastname\_entry.place(x=250,y=60)

fullname\_entry.place(x=20,y=60)

email\_address.place(x=20,y=150)

confirm\_password.place(x=250,y=240)

password.place(x=20,y=240)

#buttons

signup\_button=ctk.CTkButton(widgetframe\_su,text='Sign Up',font=('vivaldi italics',20),height=60,width=220,corner\_radius=30,fg\_color='#0c7075',hover\_color='white',border\_color='#0f969c',border\_width=3,command=signup\_user)

google\_button=ctk.CTkButton(widgetframe\_su,text="",image=google,fg\_color='#072e33',hover\_color='#072e33',width=60,border\_width=0)

phon\_button=ctk.CTkButton(widgetframe\_su,text="",image=phon,width=60,fg\_color='#072e33',hover\_color='white')

instagram\_button=ctk.CTkButton(widgetframe\_su,text="",image=instagram,width=60,fg\_color='#072e33',hover\_color='white')

#placing the buttons

signup\_label.place(x=50,y=10)

signup\_button.place(x=120,y=330)

google\_button.place(x=100,y=440)

instagram\_button.place(x=200,y=440)

phon\_button.place(x=300,y=440)

#email address validator

valid\_image = ctk.CTkImage(Image.open(r"valid.png"),size=(40,40))

invalid\_image = ctk.CTkImage(Image.open(r"invalid.png"),size=(40,40))

feedback\_label = ctk.CTkLabel(email\_address, text="", font=("Helvetica", 16),bg\_color='#0c7075',fg\_color='#0c7075',width=30,height=30)

feedback\_label.place(x=380,y=20)

def validate\_email(event):

global email\_address

email = email\_address.get()

# Regular expression to check the validity of the email including domain

email\_regex = r"^[\w\.-]+@[\w\.-]+\.[a-zA-Z]{2,}$"

valid\_length = 5 <= len(email) <= 254

if re.match(email\_regex, email) and valid\_length:

feedback\_label.configure(image=valid\_image, compound='left', fg\_color="#0c7075")

else:

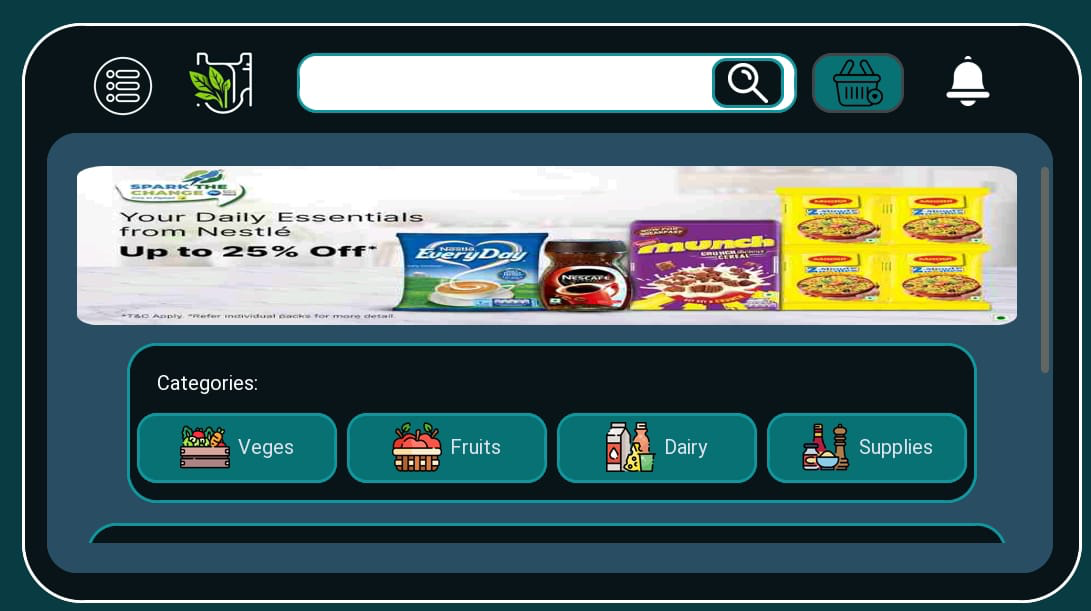
feedback\_label.configure(image=invalid\_image, compound='left', fg\_color="#0c7075")

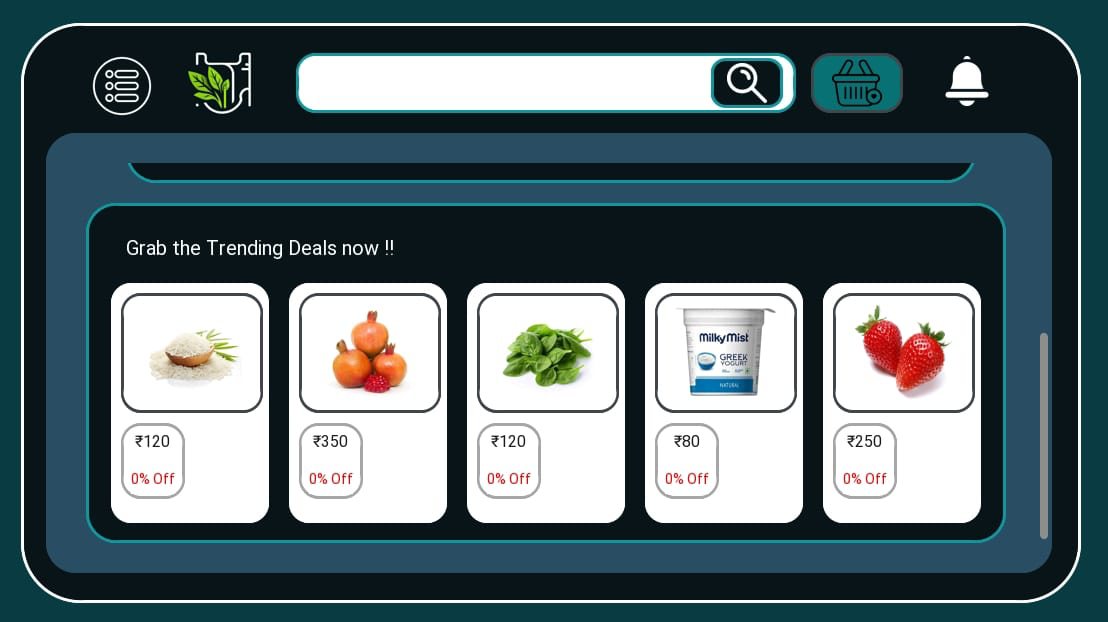
# Bind the validate\_email function to the KeyRelease event

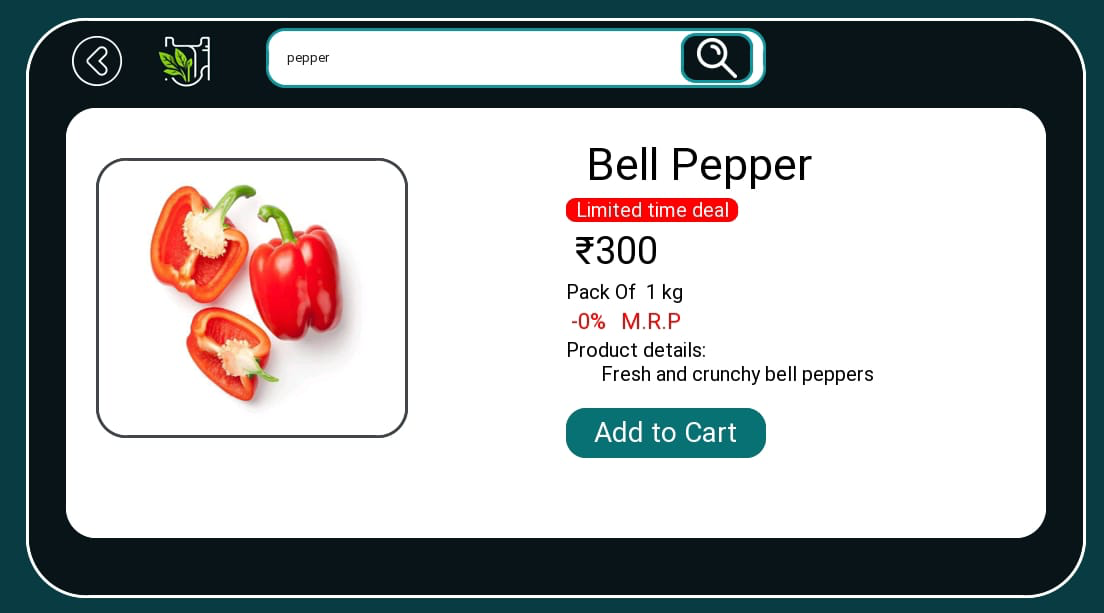
email\_address.bind("<KeyRelease>", validate\_email)

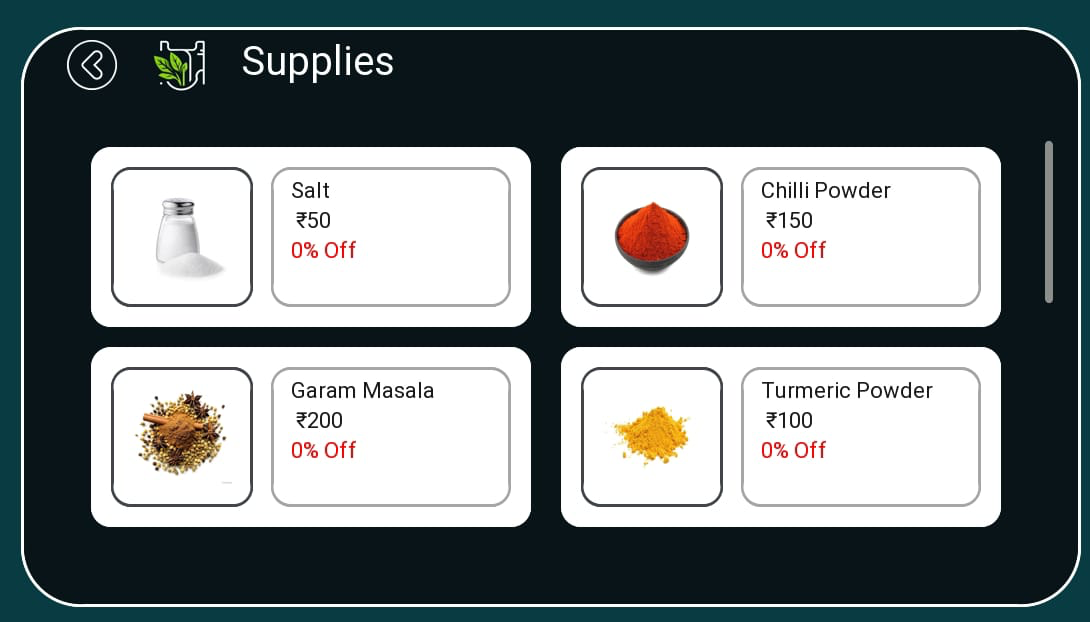
window.mainloop()

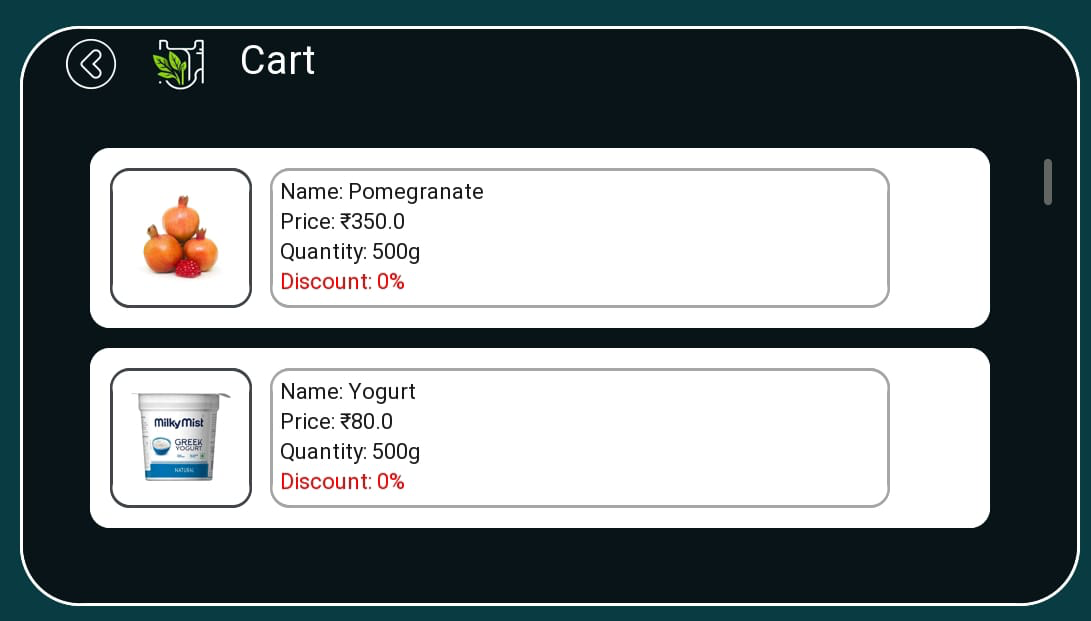
OUTPUT SEGMENT

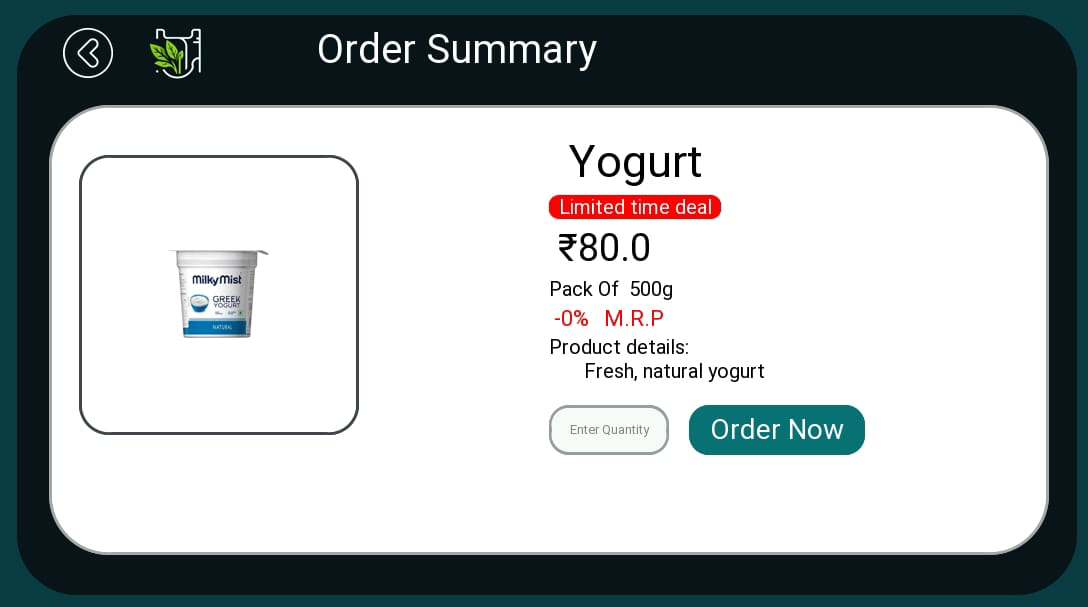


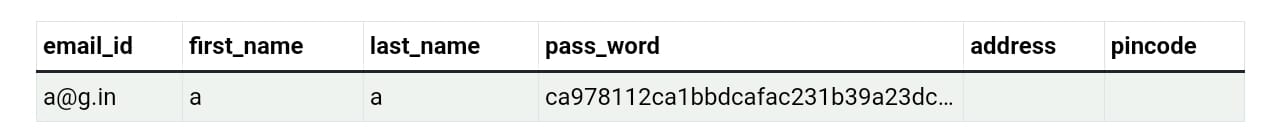












**FUNCTIONS AND FEATURES PROPOSED IN THE SYSTEM**

“GROCERIA” is designed with two features i.e. i) ADMIN and ii) CUSTOMER

**Customer Features:**

* **ACCOUNT CREATION :** Users can create a new account by providing necessary details such as username , real name and their desired password.
* **CART :** Users can wish to add different items from the store which they may decide to buy later.
* **CUSTOMER SUPPORT :** If the user is unable to understand anything , we have an exclusive customer support option which shows the layout of every corner of our app.
* **REMOTE ACCESSIBILITY :** Users can buy groceries right from their home which will be delivered within 30 mins without being externally present at the shop.
* **HISTORY :** We have a database for every user where users can see what products they bought in the recent history.
* **SMTP :** Sends mail over the internet from the app , considering protocols.

**Admin Features:**

* **Database management:** Admins can add , change , or modify products form the inventory database. They are also alloted with duty of replacing expired products.

**FRONT-END AND BACKEND USED**

**FRONTEND DEVELOPMENT:**

* The primary programming language used for frontend development is python.
* The GUI is created using Tkinter module which has a diverse variation of objects to work with.
* A message box is used to notify users for errors during their session.
* “Custom Tkinter” is also used to modify and modernise all accessibilities.
* Multiples images are used to enhance the working of the programs.

**BACKEND DEVELOPMENT:**

* The backend is connected to the fronted using the “mysql.connector” module , which has options for integration of Python to MySQL.
* MySQL is an open-source relational database management system (RDBMS) developed by Oracle Corporation. It is based on Structured Query Language (SQL), a standard language for managing and manipulating databases.
* All user data & transactions are securely stored in MySQL database.

**HARDWARE AND SOFTWARE REQUIREMENT**

RECOMMENDED REQUIREMENTS:

* WINDOWS
* 2nd generation Core i5 (2GHz+),generation core i5 processore, or equivalent
* Windows 7 or later\
* APPLE
* Mackbook Air 2012 or newer
* OS X 10.9 or later
* Chrome OS
* All 2014 models and later.

**ROLE OF STUDENTS**

ADITYA DAS : GROUP LEADER , BACKEND , FRONTEND UI , DOCUMENTATION .

ASHOK KUMAR GHOSH : BACKEND , FRONTEND

SUMAN DAS : PRESENTATION,SCRIPT

ARIYAN GUPTA : IMAGES HANDLING

MAYANK : DOCUMENTATION

ADRIJA CHAKRABORTY : PRESENTATION

**BIBLIOGRAPHY**

I would like to mention some sources which really helped me with my project

1.Geeks for Geeks

2.scribbd

3. Computer Schools

THANK YOU