

Project Code

//classes in python

```
class User:
```

```
    def __init__(self, user_id, name, email, password):
```

```
        self.user_id = user_id
```

```
        self.name = name
```

```
        self.username_field = Entry()
```

```
        self.password_field = Entry()
```

```
        self.username_label = Label(text="Username")
```

```
        self.password_label = Label(text="Password")
```

```
        self.email = email
```

```
        self.password = password
```

```
        self.signup_button = Button(text="Sign Up")
```

```
        self.login_button = Button(text="Login")
```

```
    def register(self):
```

```
        # Implementation for user registration
```

```
        pass
```

```
    def login(self):
```

```
        # Implementation for user login
```

```
        pass
```

```
def logout(self):  
    # Implementation for user logout  
    pass
```

```
class Customer:  
    def __init__(self, name, address, phone, email):  
        self.name = name  
        self.address = address  
        self.phone = phone  
        self.email = email
```

```
def add_favorite_shop(self, shop):  
    # Implementation to add a favorite shop  
    pass
```

```
def remove_favorite_shop(self, shop):  
    # Implementation to remove a favorite shop  
    pass
```

```
def make_booking(self, booking):  
    # Implementation to make a booking  
    pass
```

```
class Shop:  
    def __init__(self, shop_id, name, address, phone, email):
```

```
self.shop_id = shop_id

self.name = name

self.address = address

self.phone = phone

self.email = email

self.show_product_button = Button(text="Show Products")

self.add_product_button = Button(text="Add Product")

self.calendar_update_button = Button(text="Update Calendar")

self.make_booking_button = Button(text="Make Booking")

self.add_favorite_button = Button(text="Add to Favorites")
```

```
def add_employee(self, employee):
```

```
    # Implementation to add an employee
```

```
    pass
```

```
def remove_employee(self, employee_id):
```

```
    # Implementation to remove an employee
```

```
    pass
```

```
def add_product(self, product):
```

```
    # Implementation to add a product
```

```
    pass
```

```
def show_product_page(self):
```

```
    # Implementation to show product page
```

```
    pass
```

```
def calendar_update(self, calendar):  
    # Implementation to update calendar  
    pass
```

```
def make_booking(self, booking):  
    # Implementation to make a booking  
    pass
```

```
class Employee:  
    def __init__(self, employee_id, name, mail, phone_number, role):  
        self.employee_id = employee_id  
        self.name = name  
        self.mail = mail  
        self.phone_number = phone_number  
        self.role = role
```

```
def update_availability(self, status):  
    # Implementation to update availability  
    pass
```

```
class Profile:  
    def __init__(self, profile_id, user_id, bio, profile_picture, email):  
        self.profile_id = profile_id  
        self.user_id = user_id
```

```
self.bio = bio

self.profile_picture = profile_picture

self.email = email

self.edit_profile_button = Button(text="Edit Profile")
```

```
def edit_profile(self):

    # Implementation to edit profile

    pass
```

```
class CustomerMainMenuPage:
```

```
    def __init__(self):

        self.profile_button = Button(text="Profile")

        self.reservation_button = Button(text="Reservation")

        self.shop_button = Button(text="Shop")

        self.favorite_shop_button = Button(text="Favorite Shop")

        self.calendar_button = Button(text="Calendar")

        self.scroll_pane = Scrollbar()
```

```
    def show_profile_page(self):

        # Implementation to show profile page

        pass
```

```
    def show_reservation_page(self):

        # Implementation to show reservation page

        pass
```

```
def show_shop_page(self):  
    # Implementation to show shop page  
    pass
```

```
def show_favorite_shop_page(self):  
    # Implementation to show favorite shop page  
    pass
```

```
def show_calendar_page(self):  
    # Implementation to show calendar page  
    pass
```

```
def show_subscription_page(self):  
    # Implementation to show subscription page  
    pass
```

```
class ShopMainMenuPage:
```

```
    def __init__(self):  
        self.profile_button = Button(text="Profile")  
        self.employee_button = Button(text="Employee")  
        self.product_button = Button(text="Product")  
        self.calendar_button = Button(text="Calendar")  
        self.subscription_button = Button(text="Subscription")
```

```
self.scroll_pane = Scrollbar()
```

```
def show_profile_page(self):
```

```
    # Implementation to show profile page
```

```
    pass
```

```
def show_employee_page(self):
```

```
    # Implementation to show employee page
```

```
    pass
```

```
def show_product_page(self):
```

```
    # Implementation to show product page
```

```
    pass
```

```
def show_calendar_page(self):
```

```
    # Implementation to show calendar page
```

```
    pass
```

```
def show_subscription_page(self):
```

```
    # Implementation to show subscription page
```

```
    pass
```

```
from tkinter import *
```

```
from datetime import date, time
```

```
class Reservation:
```

```
def __init__(self, reservation_id, date, time, status):  
    self.reservation_id = reservation_id  
    self.date = date  
    self.time = time  
    self.status = status  
    self.confirm_button = Button(text="Confirm")  
    self.cancel_button = Button(text="Cancel")  
    self.activate_reminder_button = Button(text="Activate Reminder")
```

```
def confirm_booking(self):  
    # Implementation to confirm booking  
    pass
```

```
def cancel_booking(self):  
    # Implementation to cancel booking  
    pass
```

```
def activate_reminder(self, reminder):  
    # Implementation to activate reminder  
    pass
```

```
def make_booking(self):  
    # Implementation to make booking  
    pass
```

```
from tkinter import *
```



```
from datetime import date, time
```

```
class Reminder:
```

```
    def __init__(self, reminder_id, calendar_id, message, date, time):
```

```
        self.reminder_id = reminder_id
```

```
        self.calendar_id = calendar_id
```

```
        self.message = message
```

```
        self.date = date
```

```
        self.time = time
```

```
        self.create_reminder_button = Button(text="Create Reminder")
```

```
    def create_reminder(self):
```

```
        # Implementation to create reminder
```

```
        pass
```

```
from tkinter import *
```

```
class Product:
```

```
    def __init__(self, product_id, name, description, price):
```

```
        self.product_id = product_id
```

```
        self.name = name
```

```
        self.description = description
```

```
        self.price = price
```

```
        self.add_to_booking_button = Button(text="Add to Booking")
```

```
from tkinter import *
```

```
class Subscription:

    def __init__(self, subscription_id, plan, price):

        self.subscription_id = subscription_id

        self.plan = plan

        self.price = price

        self.subscribe_button = Button(text="Subscribe")
```

```
def subscribe(self):

    # Implementation to subscribe

    pass
```

```
from tkinter import *

from datetime import date
```

```
class Review:

    def __init__(self, review_id, rating, comment, date):

        self.review_id = review_id

        self.rating = rating

        self.comment = comment

        self.date = date

        self.submit_review_button = Button(text="Submit Review")
```

```
def submit_review(self):

    # Implementation to submit review

    pass
```

```
class FavoriteShop:

    def __init__(self, favorite_shop_id, user_id, shop_id):

        self.favorite_shop_id = favorite_shop_id

        self.user_id = user_id

        self.shop_id = shop_id
```

```
class Calendar:

    def __init__(self, calendar_id, user_id, events):

        self.calendar_id = calendar_id

        self.user_id = user_id

        self.events = events
```

```
def add_event(self, event):

    # Implementation to add event to calendar

    pass
```

UPDATE with methods

Methods

```
class Customer:

    def __init__(self, name, address, phone, email):

        self.name = name

        self.address = address

        self.phone = phone

        self.email = email

        self.favorite_shops = [] # Initialize an empty list to store favorite shops


    def add_favorite_shop(self, shop):

        if shop not in self.favorite_shops:

            self.favorite_shops.append(shop)

            print(f"{shop.name} has been added to your favorite shops.")

        else:

            print(f"{shop.name} is already in your favorite shops.")


    def remove_favorite_shop(self, shop):

        if shop in self.favorite_shops:

            self.favorite_shops.remove(shop)

            print(f"{shop.name} has been removed from your favorite shops.")

        else:

            print(f"{shop.name} is not in your favorite shops.")


    def make_booking(self, shop, date, time):

        booking = Booking(self, shop, date, time)

        shop.make_booking(booking)
```

```
class Reservation:
```

```
    def __init__(self, reservation_id, date, time, status):
```

```
        self.reservation_id = reservation_id
```

```
        self.date = date
```

```
        self.time = time
```

```
        self.status = status
```

```
        self.confirm_button = Button(text="Confirm")
```

```
        self.cancel_button = Button(text="Cancel")
```

```
        self.activate_reminder_button = Button(text="Activate Reminder")
```

```
    def confirm_booking(self):
```

```
        self.status = "Confirmed"
```

```
        print("Booking confirmed.")
```

```
    def cancel_booking(self):
```

```
        self.status = "Cancelled"
```

```
        print("Booking cancelled.")
```

```
    def activate_reminder(self, reminder):
```

```
        # Implementation to activate reminder
```

```
        pass
```

```
class Booking:
```

```
    def __init__(self, customer, shop, date, time):
```

```
        self.customer = customer
```

```
        self.shop = shop
```

```
self.date = date
```

```
self.time = time
```

```
def make_booking(self):
```

```
    # Implementation to make booking
```

```
    pass
```

```
class Shop:
```

```
    def __init__(self, shop_id, name, address, phone, email):
```

```
        self.shop_id = shop_id
```

```
        self.name = name
```

```
        self.address = address
```

```
        self.phone = phone
```

```
        self.email = email
```

```
        self.show_product_button = Button(text="Show Products")
```

```
        self.add_product_button = Button(text="Add Product")
```

```
        self.calendar_update_button = Button(text="Update Calendar")
```

```
        self.make_booking_button = Button(text="Make Booking")
```

```
        self.add_favorite_button = Button(text="Add to Favorites")
```

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

```
        self.name = name
```

```
        self.reservations = []
```

```
    def add_employee(self, employee):
```

```
self.employees.append(employee)

print(f"Employee {employee.name} added to {self.name}")


def remove_employee(self, employee_id):
    for employee in self.employees:
        if employee.employee_id == employee_id:
            self.employees.remove(employee)

            print(f"Employee {employee.name} removed from {self.name}")

            return

    print(f"Employee with ID {employee_id} not found in {self.name}")


def add_product(self, product):
    self.products.append(product)

    print(f"Product '{product.name}' added to {self.name}")


def show_product_page(self):
    print(f"Displaying product page for shop: {self.name}")

    pass


def calendar_update(self, event):
    self.calendar.append(event)

    print(f"Calendar updated with event: {event}")


def receive_reservation(self, reservation):
    self.reservations.append(reservation)

    print("Reservation received.")
```

```
class Employee:

    def __init__(self, employee_id, name, mail, phone_number, role):

        self.employee_id = employee_id

        self.name = name

        self.mail = mail

        self.phone_number = phone_number

        self.role = role


    def update_availability(self, status):

        if status.lower() == "available":

            self.available = True

        elif status.lower() == "unavailable":

            self.available = False

        else:

            print("Invalid availability status. Please specify 'available' or 'unavailable!'")
```