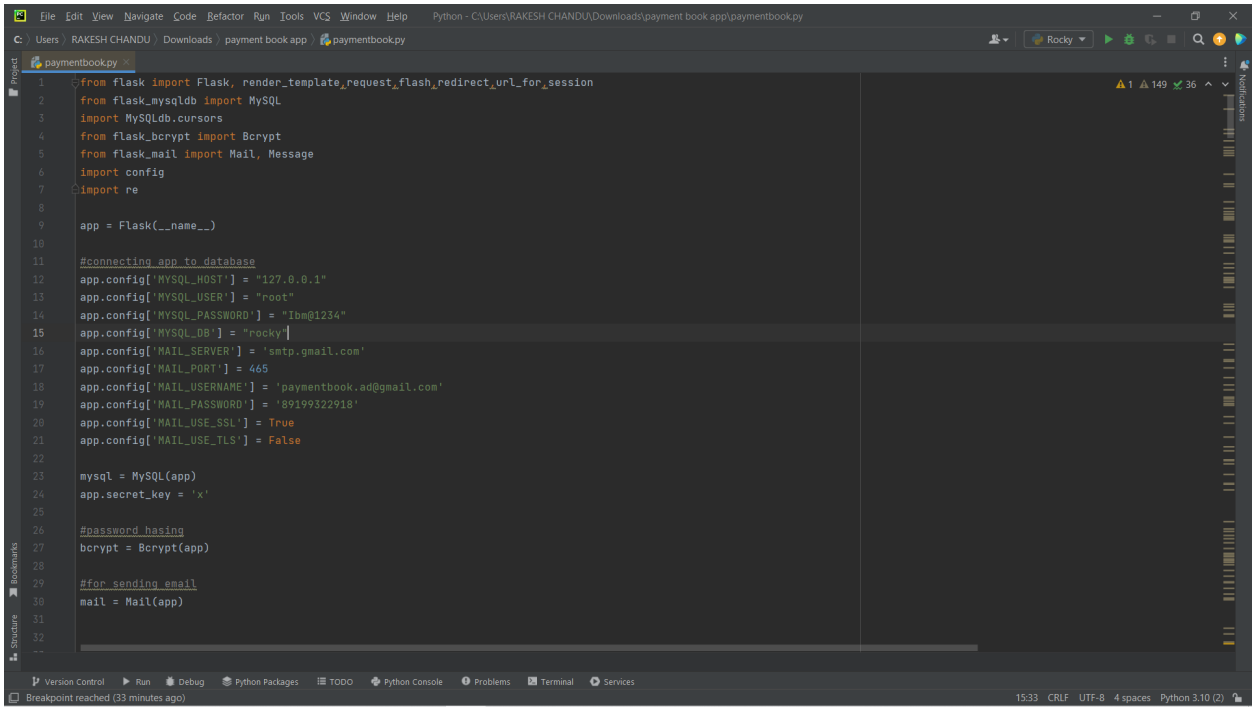
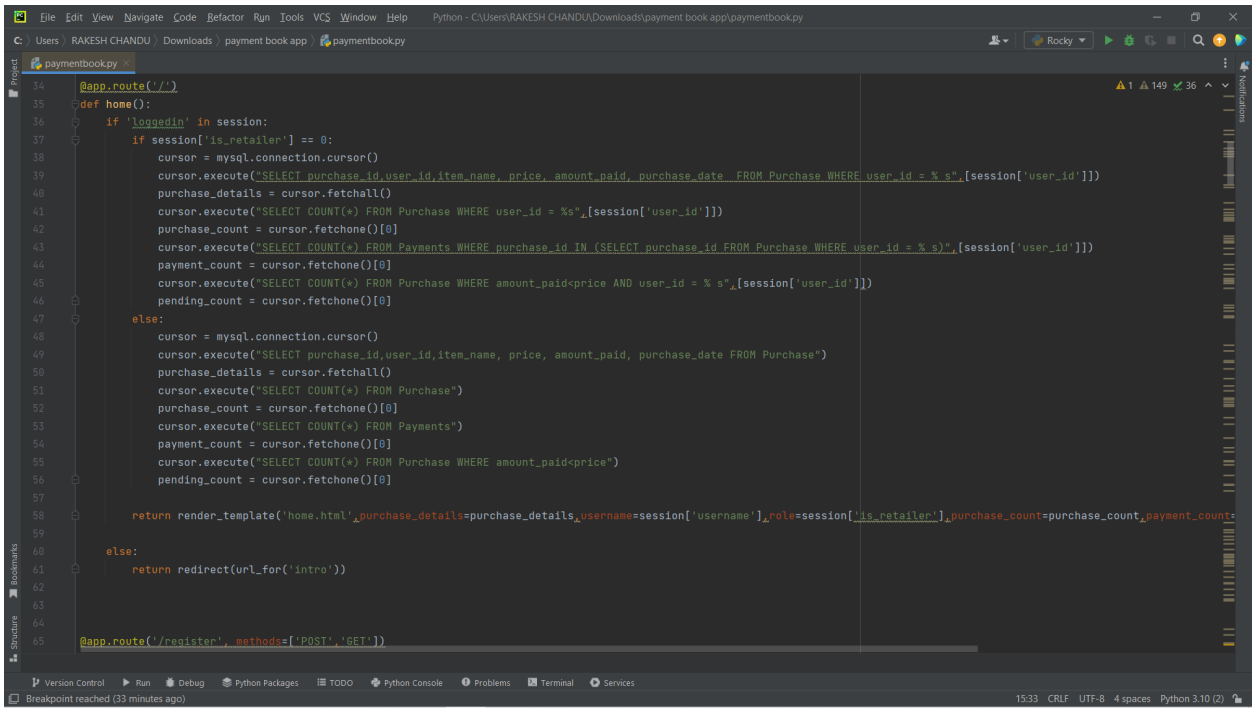
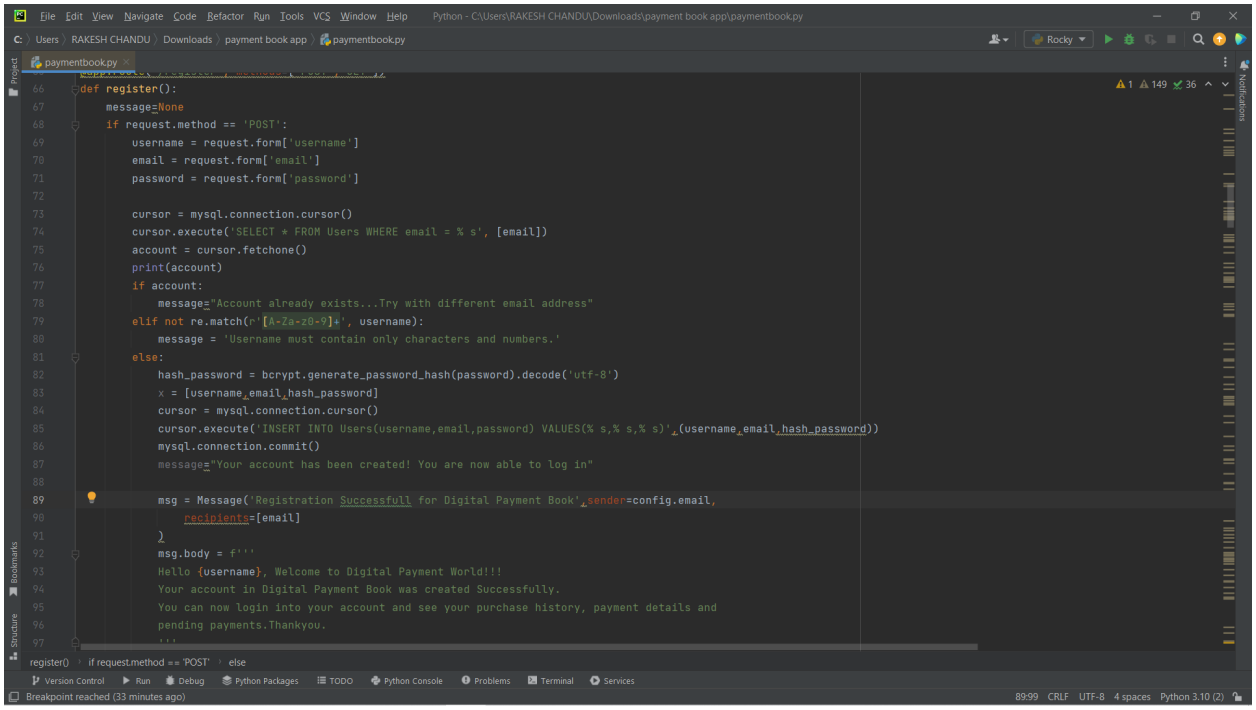
Code:



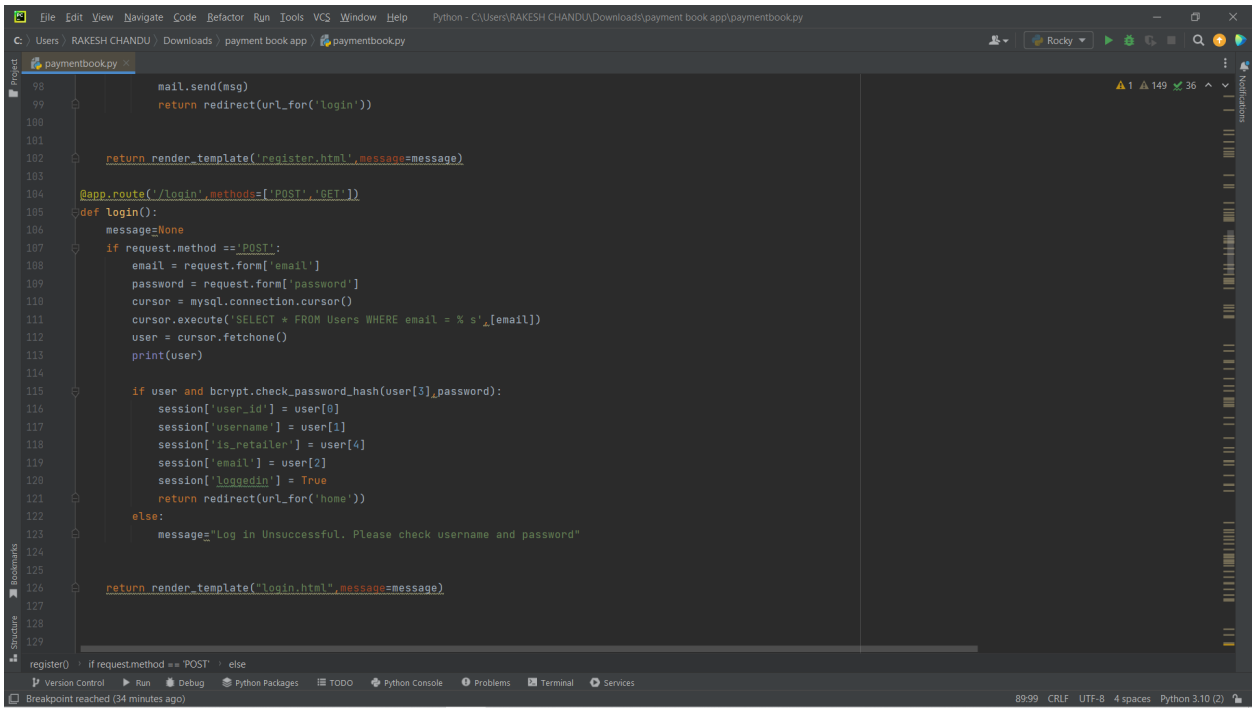
```python
from flask import Flask, render_template,request,flash,redirect,url_for,session
from flask_mysqldb import MySQL
import MySQLdb.cursors
from flask_bcrypt import Bcrypt
from flask_mail import Mail, Message
import config
import re

app = Flask(__name__)

#connecting app to database
app.config['MYSQL_HOST'] = "127.0.0.1"
app.config['MYSQL_USER'] = "root"
app.config['MYSQL_PASSWORD'] = "Ibm@1234"
app.config['MYSQL_DB'] = "rocky"
app.config['MAIL_SERVER'] = 'smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'paymentbook.ad@gmail.com'
app.config['MAIL_PASSWORD'] = '89199322918'
app.config['MAIL_USE_SSL'] = True
app.config['MAIL_USE_TLS'] = False

mysql = MySQL(app)
app.secret_key = 'x'

#password hasing
bcrypt = Bcrypt(app)

#for sending email
mail = Mail(app)
```



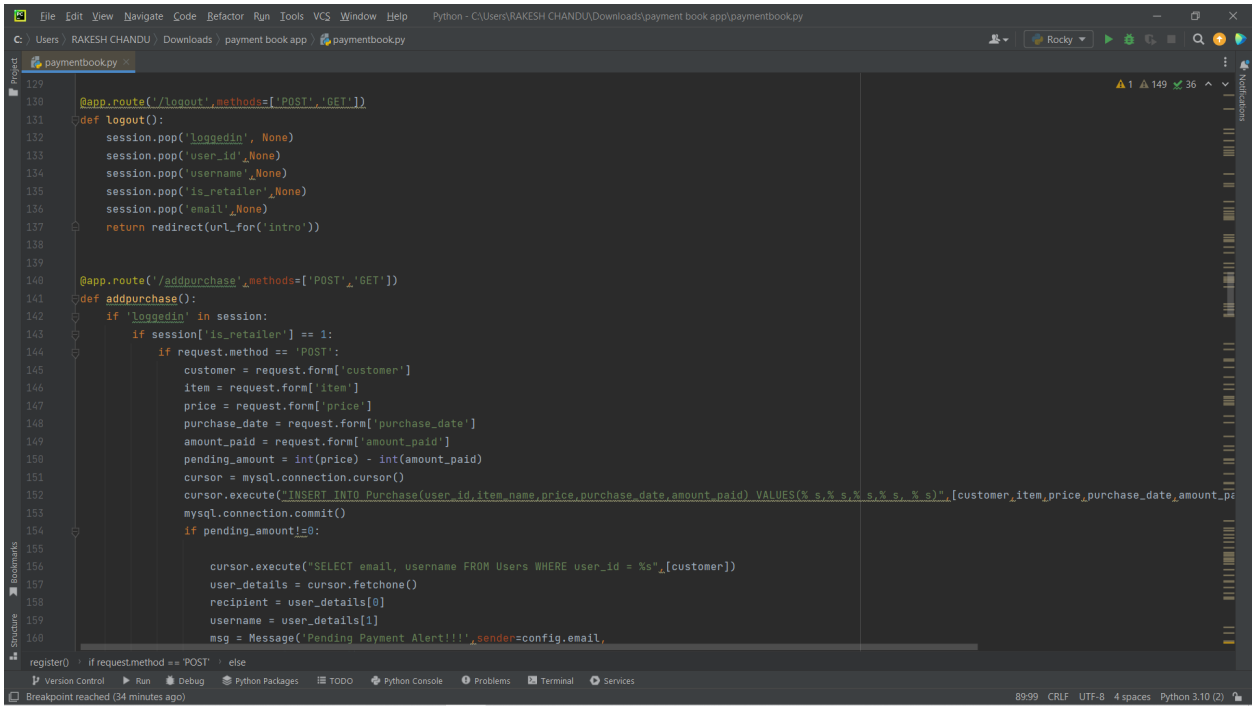
```python
@app.route('/')
def home():
    if 'loggedin' in session:
        if session['is_retailer'] == 0:
            cursor = mysql.connection.cursor()
            cursor.execute("SELECT purchase_id,user_id,item_name, price, amount_paid, purchase_date  FROM Purchase WHERE user_id = % s",[session['user_id']])
            purchase_details = cursor.fetchall()
            cursor.execute("SELECT COUNT(*) FROM Purchase WHERE user_id = %s",[session['user_id']])
            purchase_count = cursor.fetchone()[0]
            cursor.execute("SELECT COUNT(*) FROM Payments WHERE purchase_id IN (SELECT purchase_id FROM Purchase WHERE user_id = % s)",[session['user_id']])
            payment_count = cursor.fetchone()[0]
            cursor.execute("SELECT COUNT(*) FROM Purchase WHERE amount_paid<price AND user_id = % s",[session['user_id']])
            pending_count = cursor.fetchone()[0]
        else:
            cursor = mysql.connection.cursor()
            cursor.execute("SELECT purchase_id,user_id,item_name, price, amount_paid, purchase_date FROM Purchase")
            purchase_details = cursor.fetchall()
            cursor.execute("SELECT COUNT(*) FROM Purchase")
            purchase_count = cursor.fetchone()[0]
            cursor.execute("SELECT COUNT(*) FROM Payments")
            payment_count = cursor.fetchone()[0]
            cursor.execute("SELECT COUNT(*) FROM Purchase WHERE amount_paid<price")
            pending_count = cursor.fetchone()[0]

        return render_template('home.html',purchase_details=purchase_details,username=session['username'],role=session['is_retailer'],purchase_count=purchase_count,payment_count=

    else:
        return redirect(url_for('intro'))



@app.route('/register', methods=['POST','GET'])
```



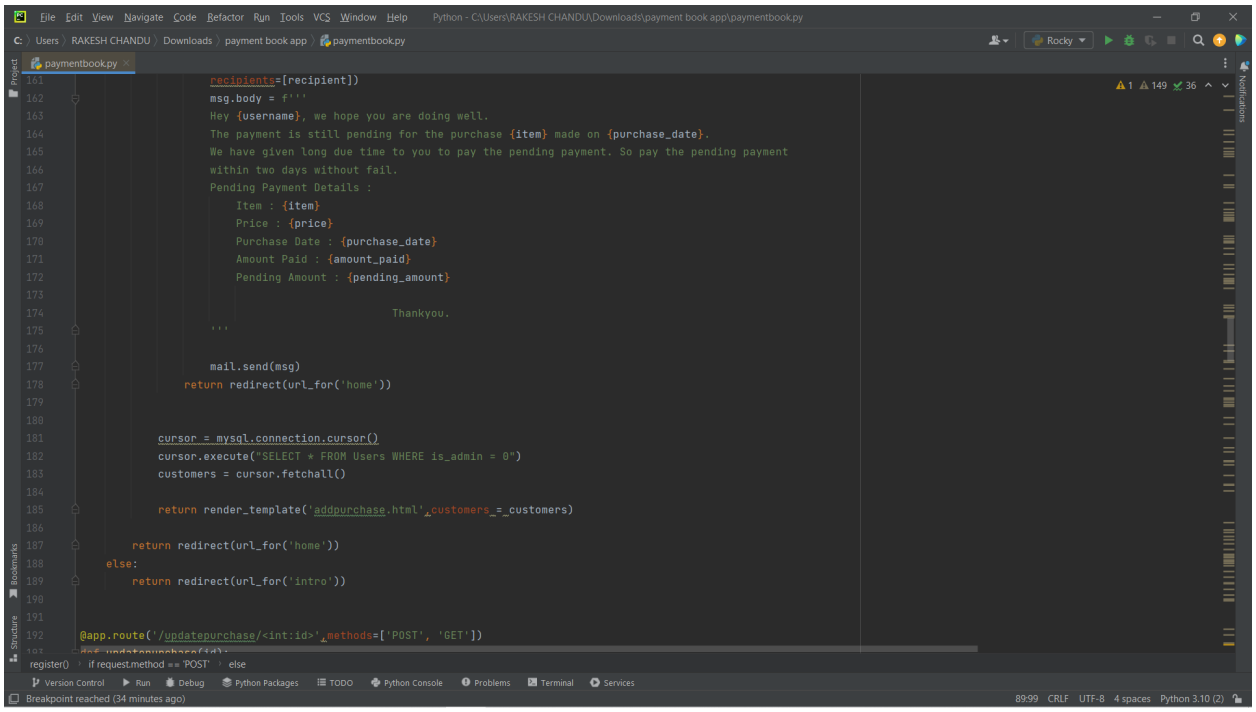
```
def register():
    message=None
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']

        cursor = mysql.connection.cursor()
        cursor.execute('SELECT * FROM Users WHERE email = % s', [email])
        account = cursor.fetchone()
        print(account)
        if account:
            message="Account already exists...Try with different email address"
        elif not re.match(r'[A-Za-z0-9]+', username):
            message = 'Username must contain only characters and numbers.'
        else:
            hash_password = bcrypt.generate_password_hash(password).decode('utf-8')
            x = [username,email,hash_password]
            cursor = mysql.connection.cursor()
            cursor.execute('INSERT INTO Users(username,email,password) VALUES(% s,% s,% s)',(username,email,hash_password))
            mysql.connection.commit()
            message="Your account has been created! You are now able to log in"

            msg = Message('Registration Successfull for Digital Payment Book',sender=config.email,
                recipients=[email]
            )
            msg.body = f'''
            Hello {username}, Welcome to Digital Payment World!!!
            Your account in Digital Payment Book was created Successfully.
            You can now login into your account and see your purchase history, payment details and
            pending payments.Thankyou.
            '''
```



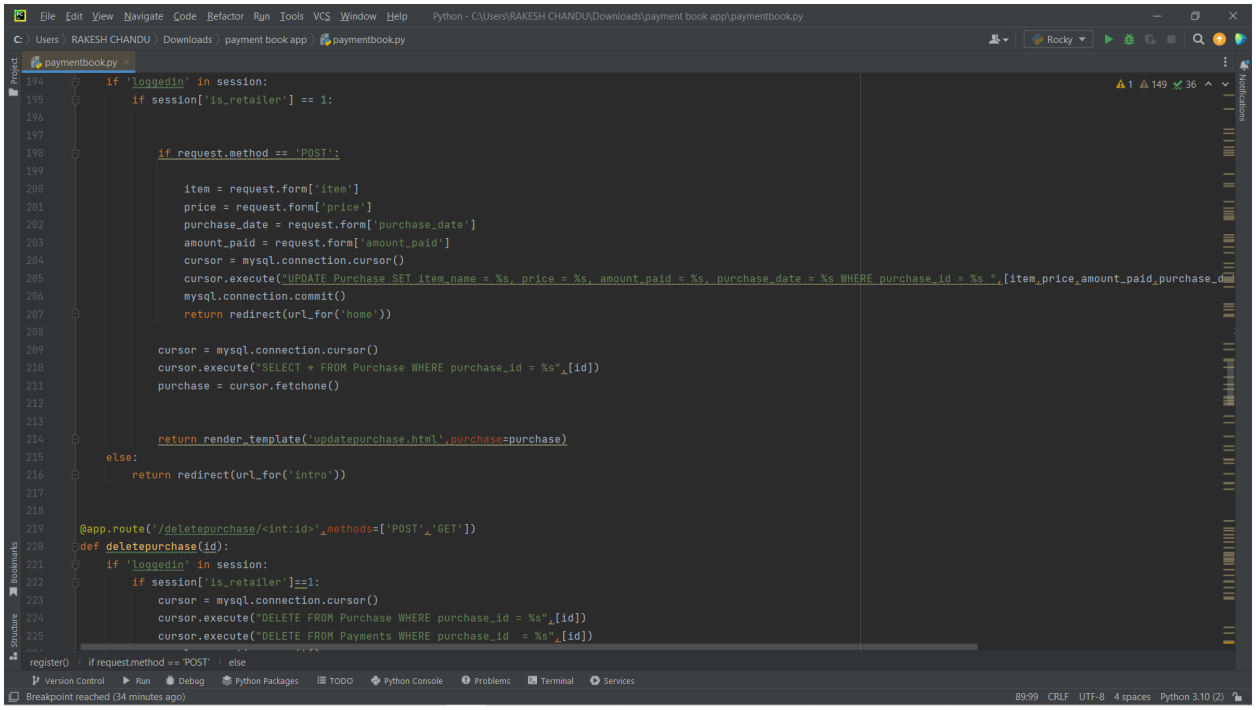
```
            mail.send(msg)
            return redirect(url_for('login'))


    return render_template('register.html',message=message)

@app.route('/login',methods=['POST','GET'])
def login():
    message=None
    if request.method =='POST':
        email = request.form['email']
        password = request.form['password']
        cursor = mysql.connection.cursor()
        cursor.execute('SELECT * FROM Users WHERE email = % s',[email])
        user = cursor.fetchone()
        print(user)

        if user and bcrypt.check_password_hash(user[3],password):
            session['user_id'] = user[0]
            session['username'] = user[1]
            session['is_retailer'] = user[4]
            session['email'] = user[2]
            session['loggedin'] = True
            return redirect(url_for('home'))
        else:
            message="Log in Unsuccessful. Please check username and password"


    return render_template("login.html",message=message)
```



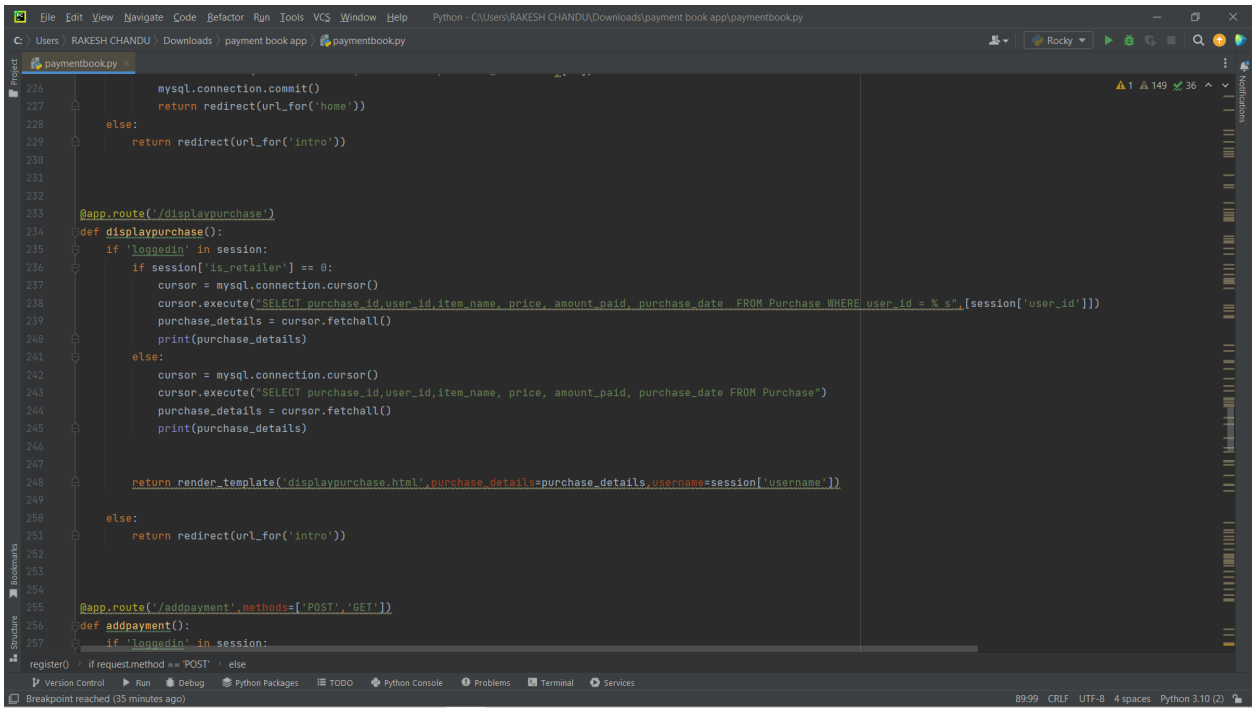
```
@app.route('/logout',methods=['POST','GET'])
def logout():
    session.pop('loggedin', None)
    session.pop('user_id',None)
    session.pop('username',None)
    session.pop('is_retailer',None)
    session.pop('email',None)
    return redirect(url_for('intro'))


@app.route('/addpurchase',methods=['POST','GET'])
def addpurchase():
    if 'loggedin' in session:
        if session['is_retailer'] == 1:
            if request.method == 'POST':
                customer = request.form['customer']
                item = request.form['item']
                price = request.form['price']
                purchase_date = request.form['purchase_date']
                amount_paid = request.form['amount_paid']
                pending_amount = int(price) - int(amount_paid)
                cursor = mysql.connection.cursor()
                cursor.execute("INSERT INTO Purchase(user_id,item_name,price,purchase_date,amount_paid) VALUES(%s,%s,%s,%s,%s,%s)",[customer,item,price,purchase_date,amount_pa
                mysql.connection.commit()
                if pending_amount!=0:

                    cursor.execute("SELECT email, username FROM Users WHERE user_id = %s",[customer])
                    user_details = cursor.fetchone()
                    recipient = user_details[0]
                    username = user_details[1]
                    msg = Message('Pending Payment Alert!!!!',sender=config.email,
```



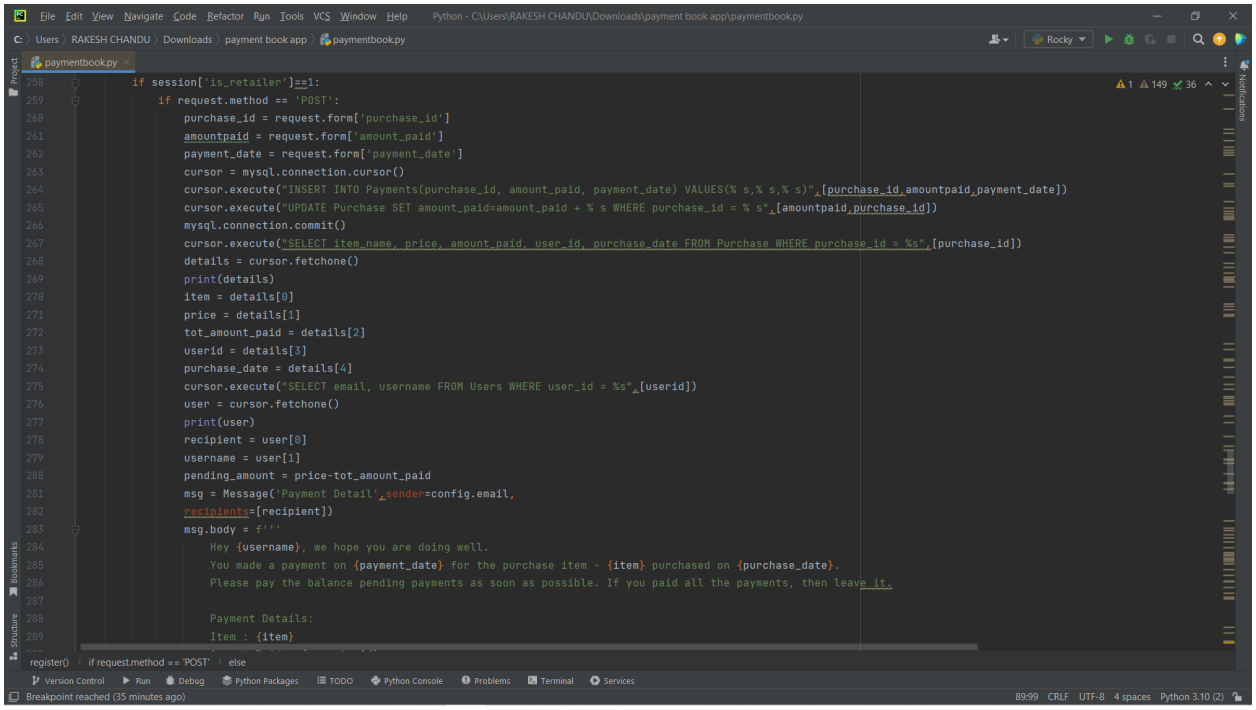
```
                    recipients=[recipient])
                    msg.body = f'''
                    Hey {username}, we hope you are doing well.
                    The payment is still pending for the purchase {item} made on {purchase_date}.
                    We have given long due time to you to pay the pending payment. So pay the pending payment
                    within two days without fail.
                    Pending Payment Details :
                        Item : {item}
                        Price : {price}
                        Purchase Date : {purchase_date}
                        Amount Paid : {amount_paid}
                        Pending Amount : {pending_amount}

                                        Thankyou.
                    '''

                    mail.send(msg)
                return redirect(url_for('home'))


            cursor = mysql.connection.cursor()
            cursor.execute("SELECT * FROM Users WHERE is_admin = 0")
            customers = cursor.fetchall()

            return render_template('addpurchase.html',customers=_customers)

        return redirect(url_for('home'))
    else:
        return redirect(url_for('intro'))


@app.route('/updatepurchase/<int:id>',methods=['POST', 'GET'])
```



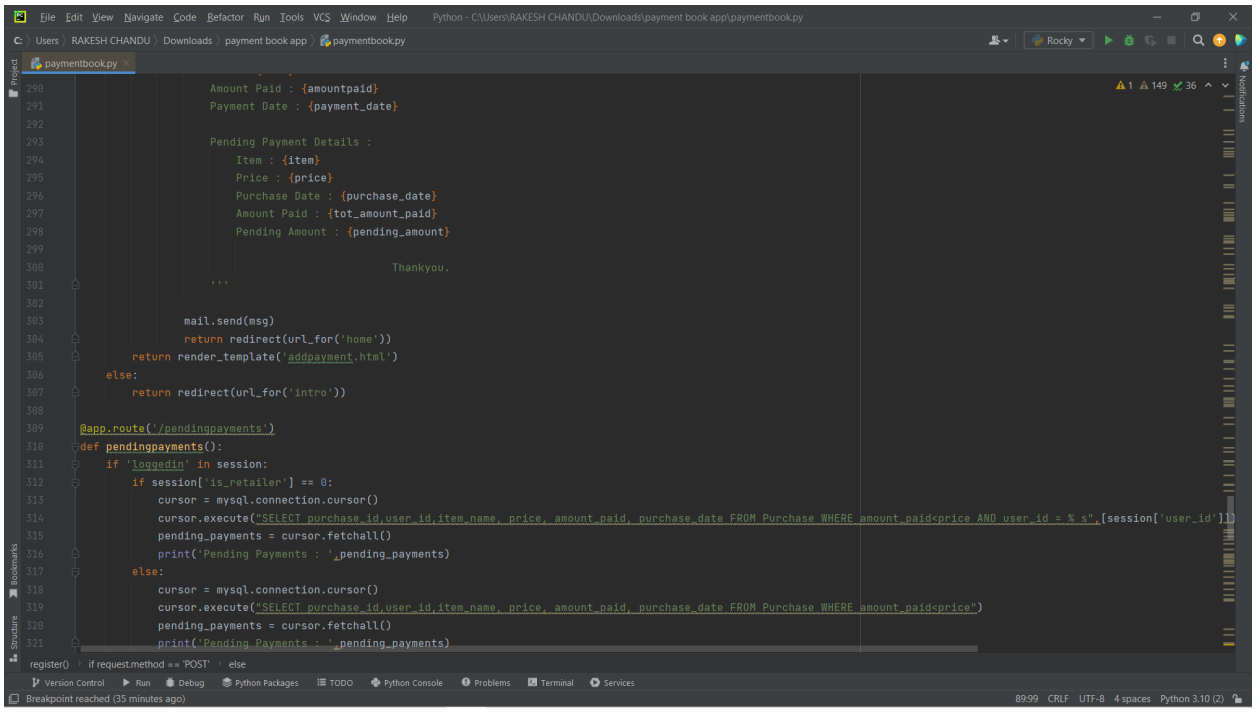
```
    if 'loggedin' in session:
        if session['is_retailer'] == 1:


            if request.method == 'POST':

                item = request.form['item']
                price = request.form['price']
                purchase_date = request.form['purchase_date']
                amount_paid = request.form['amount_paid']
                cursor = mysql.connection.cursor()
                cursor.execute("UPDATE Purchase SET item_name = %s, price = %s, amount_paid = %s, purchase_date = %s WHERE purchase_id = %s ",[item,price,amount_paid,purchase_d
                mysql.connection.commit()
                return redirect(url_for('home'))

            cursor = mysql.connection.cursor()
            cursor.execute("SELECT * FROM Purchase WHERE purchase_id = %s",[id])
            purchase = cursor.fetchone()


            return render_template('updatepurchase.html',purchase=purchase)
    else:
        return redirect(url_for('intro'))


@app.route('/deletepurchase/<int:id>',methods=['POST','GET'])
def deletepurchase(id):
    if 'loggedin' in session:
        if session['is_retailer']==1:
            cursor = mysql.connection.cursor()
            cursor.execute("DELETE FROM Purchase WHERE purchase_id = %s",[id])
            cursor.execute("DELETE FROM Payments WHERE purchase_id  = %s",[id])
```



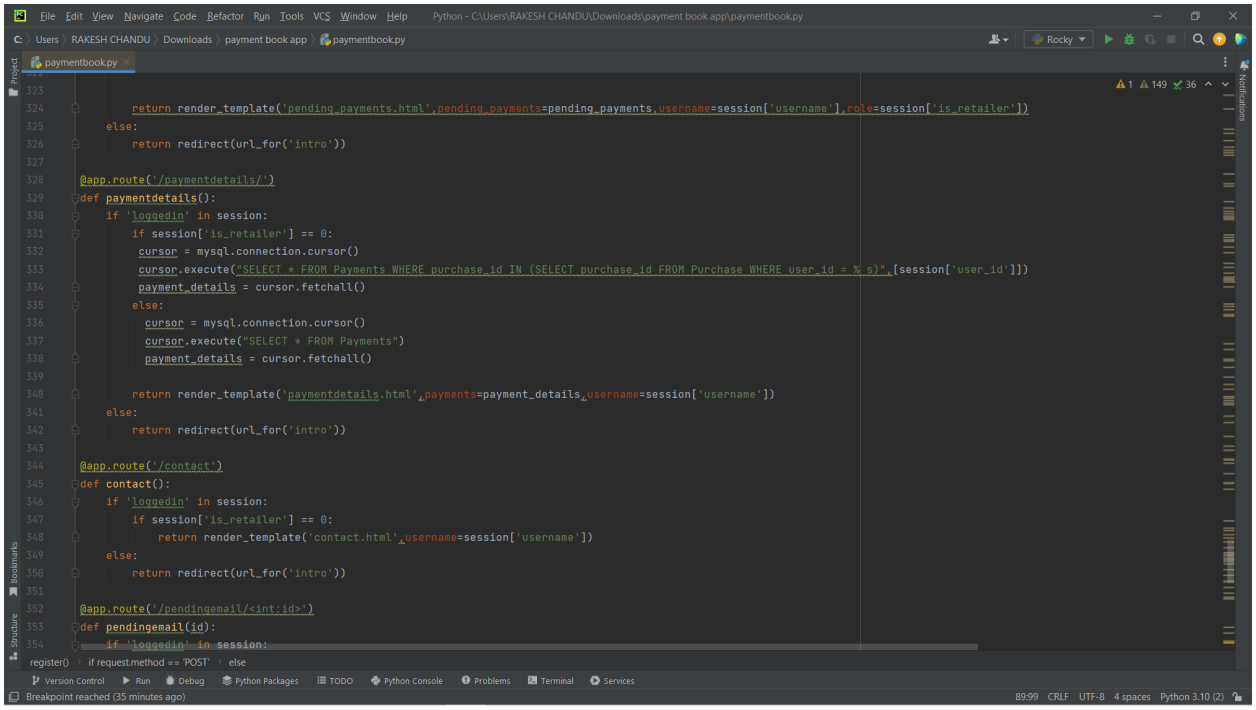
```
            mysql.connection.commit()
            return redirect(url_for('home'))
    else:
        return redirect(url_for('intro'))



@app.route('/displaypurchase')
def displaypurchase():
    if 'loggedin' in session:
        if session['is_retailer'] == 0:
            cursor = mysql.connection.cursor()
            cursor.execute("SELECT purchase_id,user_id,item_name, price, amount_paid, purchase_date  FROM Purchase WHERE user_id = % s",[session['user_id']])
            purchase_details = cursor.fetchall()
            print(purchase_details)
        else:
            cursor = mysql.connection.cursor()
            cursor.execute("SELECT purchase_id,user_id,item_name, price, amount_paid, purchase_date FROM Purchase")
            purchase_details = cursor.fetchall()
            print(purchase_details)


        return render_template('displaypurchase.html',purchase_details=purchase_details,username=session['username'])

    else:
        return redirect(url_for('intro'))



@app.route('/addpayment',methods=['POST','GET'])
def addpayment():
    if 'loggedin' in session:
```



```
        if session['is_retailer']==1:
            if request.method == 'POST':
                purchase_id = request.form['purchase_id']
                amountpaid = request.form['amount_paid']
                payment_date = request.form['payment_date']
                cursor = mysql.connection.cursor()
                cursor.execute("INSERT INTO Payments(purchase_id, amount_paid, payment_date) VALUES(% s,% s,% s)",[purchase_id,amountpaid,payment_date])
                cursor.execute("UPDATE Purchase SET amount_paid=amount_paid + % s WHERE purchase_id = % s",[amountpaid,purchase_id])
                mysql.connection.commit()
                cursor.execute("SELECT item_name, price, amount_paid, user_id, purchase_date FROM Purchase WHERE purchase_id = %s",[purchase_id])
                details = cursor.fetchone()
                print(details)
                item = details[0]
                price = details[1]
                tot_amount_paid = details[2]
                userid = details[3]
                purchase_date = details[4]
                cursor.execute("SELECT email, username FROM Users WHERE user_id = %s",[userid])
                user = cursor.fetchone()
                print(user)
                recipient = user[0]
                username = user[1]
                pending_amount = price-tot_amount_paid
                msg = Message('Payment Detail',sender=config.email,
                recipients=[recipient])
                msg.body = f'''
                    Hey {username}, we hope you are doing well.
                    You made a payment on {payment_date} for the purchase item - {item} purchased on {purchase_date}.
                    Please pay the balance pending payments as soon as possible. If you paid all the payments, then leave it.

                    Payment Details:
                    Item : {item}
```



```
                    Amount Paid : {amountpaid}
                    Payment Date : {payment_date}

                    Pending Payment Details :
                        Item : {item}
                        Price : {price}
                        Purchase Date : {purchase_date}
                        Amount Paid : {tot_amount_paid}
                        Pending Amount : {pending_amount}

                                            Thankyou.
                    '''

                mail.send(msg)
                return redirect(url_for('home'))
        return render_template('addpayment.html')
    else:
        return redirect(url_for('intro'))

@app.route('/pendingpayments')
def pendingpayments():
    if 'loggedin' in session:
        if session['is_retailer'] == 0:
            cursor = mysql.connection.cursor()
            cursor.execute("SELECT purchase_id,user_id,item_name, price, amount_paid, purchase_date FROM Purchase WHERE amount_paid<price AND user_id = % s",[session['user_id']])
            pending_payments = cursor.fetchall()
            print('Pending Payments : ',pending_payments)
        else:
            cursor = mysql.connection.cursor()
            cursor.execute("SELECT purchase_id,user_id,item_name, price, amount_paid, purchase_date FROM Purchase WHERE amount_paid<price")
            pending_payments = cursor.fetchall()
            print('Pending Payments : ',pending_payments)
```

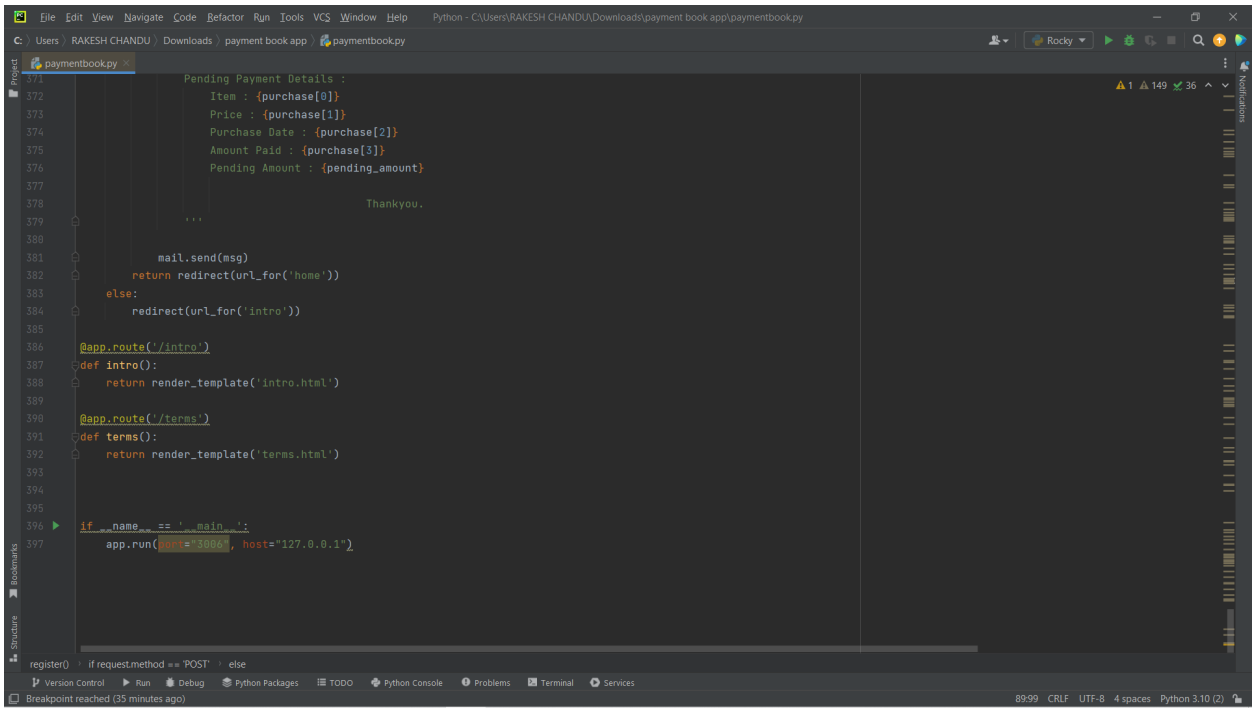


```
        return render_template('pending_payments.html',pending_payments=pending_payments,username=session['username'],role=session['is_retailer'])
    else:
        return redirect(url_for('intro'))

@app.route('/paymentdetails/')
def paymentdetails():
    if 'loggedin' in session:
        if session['is_retailer'] == 0:
         cursor = mysql.connection.cursor()
         cursor.execute("SELECT * FROM Payments WHERE purchase_id IN (SELECT purchase_id FROM Purchase WHERE user_id = %s)",[session['user_id']])
         payment_details = cursor.fetchall()
        else:
          cursor = mysql.connection.cursor()
          cursor.execute("SELECT * FROM Payments")
          payment_details = cursor.fetchall()

        return render_template('paymentdetails.html',payments=payment_details,username=session['username'])
    else:
        return redirect(url_for('intro'))

@app.route('/contact')
def contact():
    if 'loggedin' in session:
        if session['is_retailer'] == 0:
            return render_template('contact.html',username=session['username'])
    else:
        return redirect(url_for('intro'))

@app.route('/pendingemail/<int:id>')
def pendingemail(id):
    if 'loggedin' in session:
```

register() › if request.method == 'POST' › else

Version Control | Run | Debug | Python Packages | TODO | Python Console | Problems | Terminal | Services

Breakpoint reached (35 minutes ago) — 89:99 CRLF UTF-8 4 spaces Python 3.10 (2)



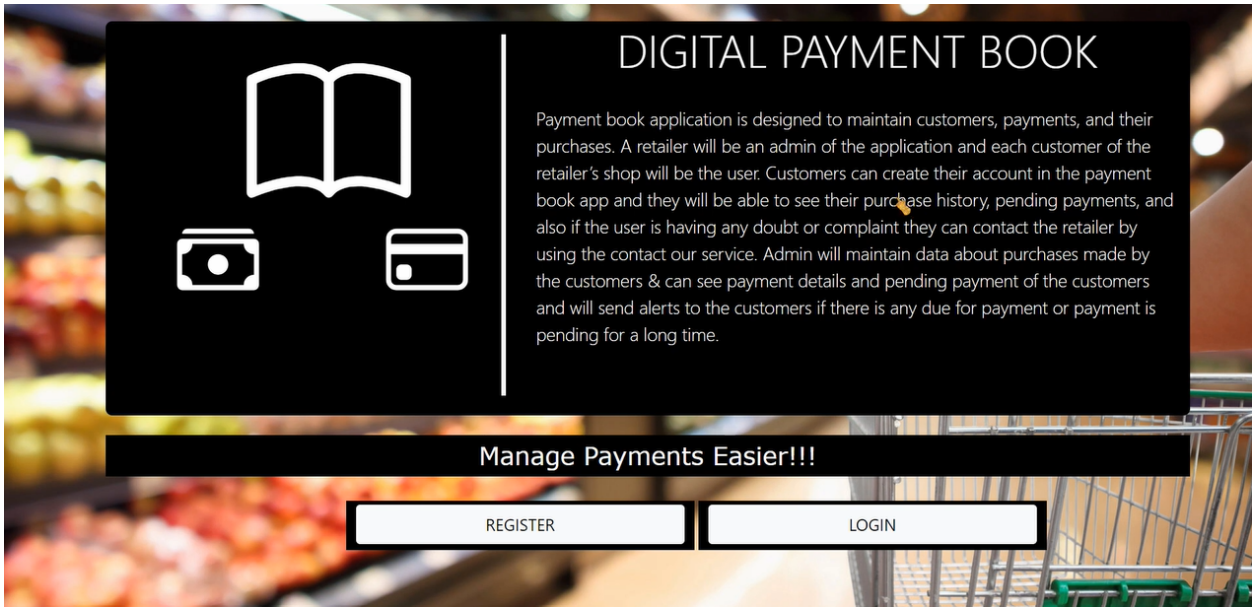
```
                    Pending Payment Details :
                        Item : {purchase[0]}
                        Price : {purchase[1]}
                        Purchase Date : {purchase[2]}
                        Amount Paid : {purchase[3]}
                        Pending Amount : {pending_amount}

                                            Thankyou.
                    '''

            mail.send(msg)
        return redirect(url_for('home'))
    else:
        redirect(url_for('intro'))

@app.route('/intro')
def intro():
    return render_template('intro.html')

@app.route('/terms')
def terms():
    return render_template('terms.html')



if __name__ == '__main__':
    app.run(port="3006", host="127.0.0.1")
```

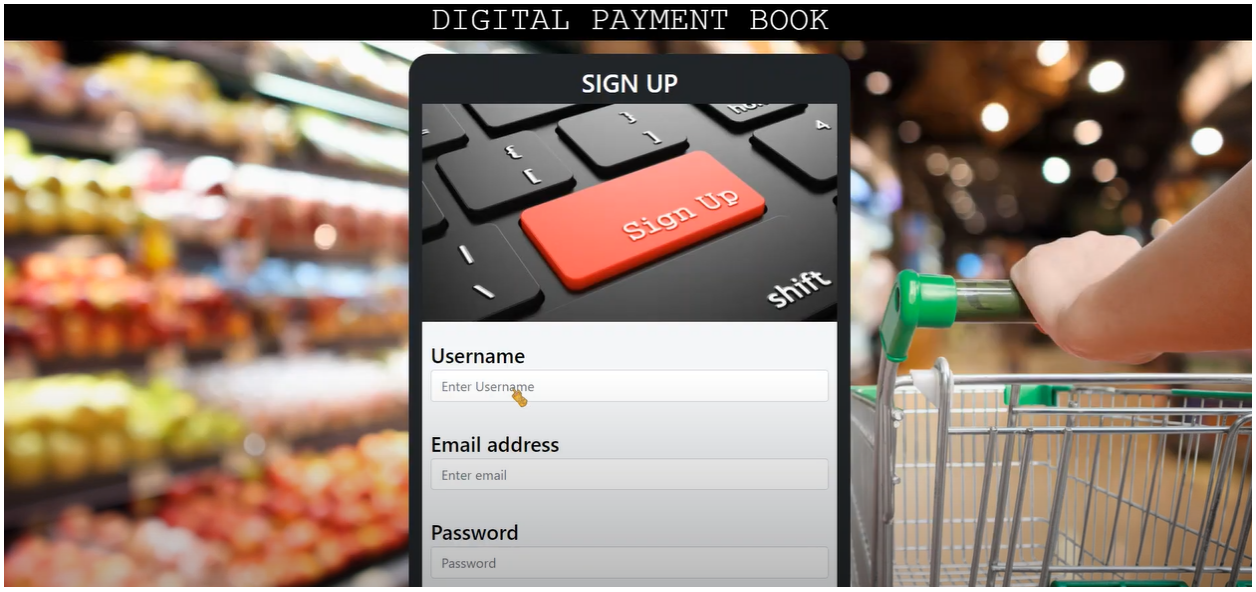
register() › if request.method == 'POST' › else

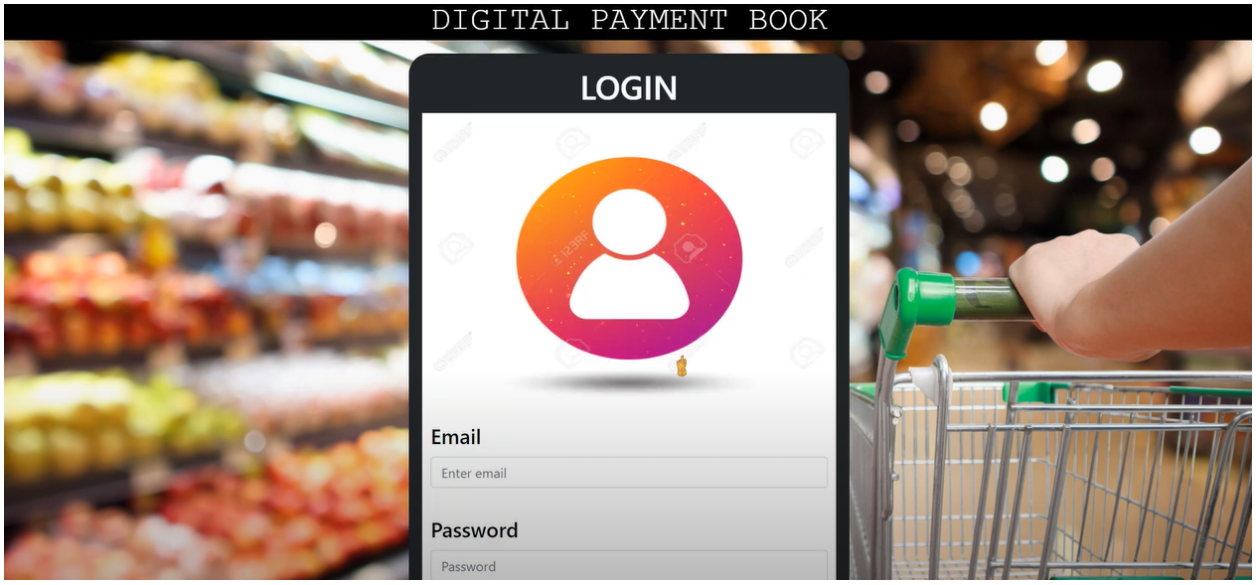
Version Control | Run | Debug | Python Packages | TODO | Python Console | Problems | Terminal | Services

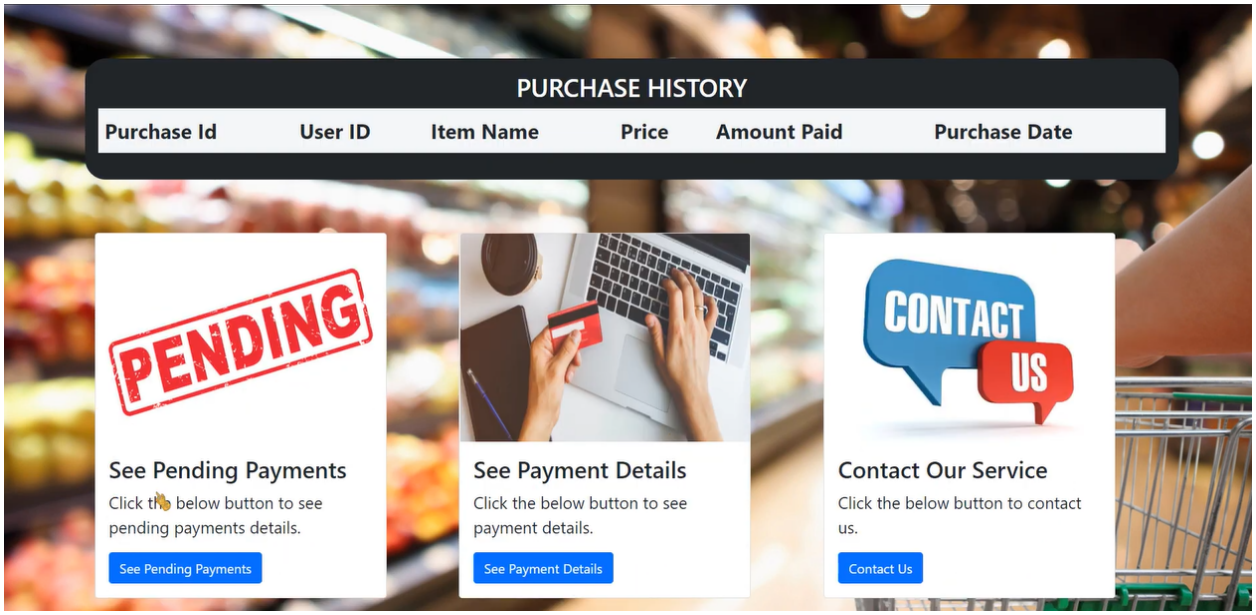
Breakpoint reached (35 minutes ago) — 89:99 CRLF UTF-8 4 spaces Python 3.10 (2)

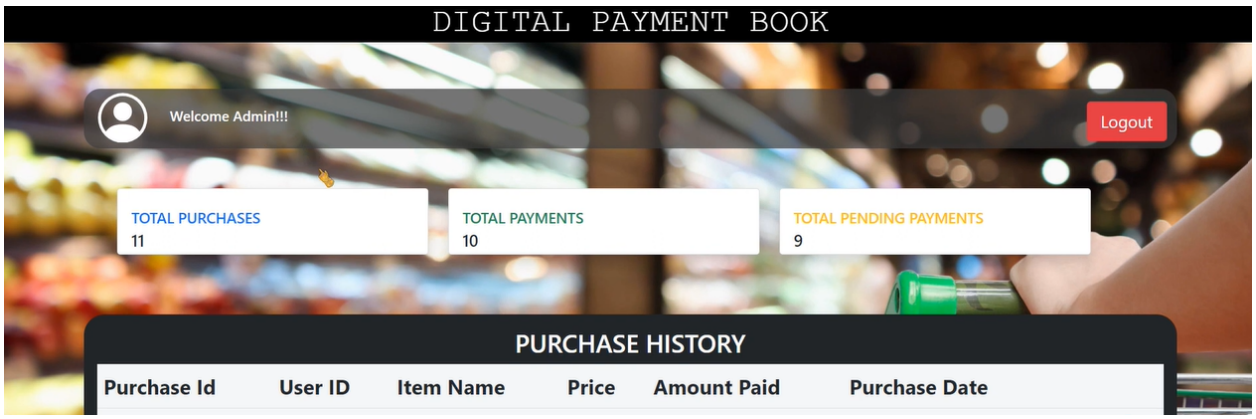
Outputs:

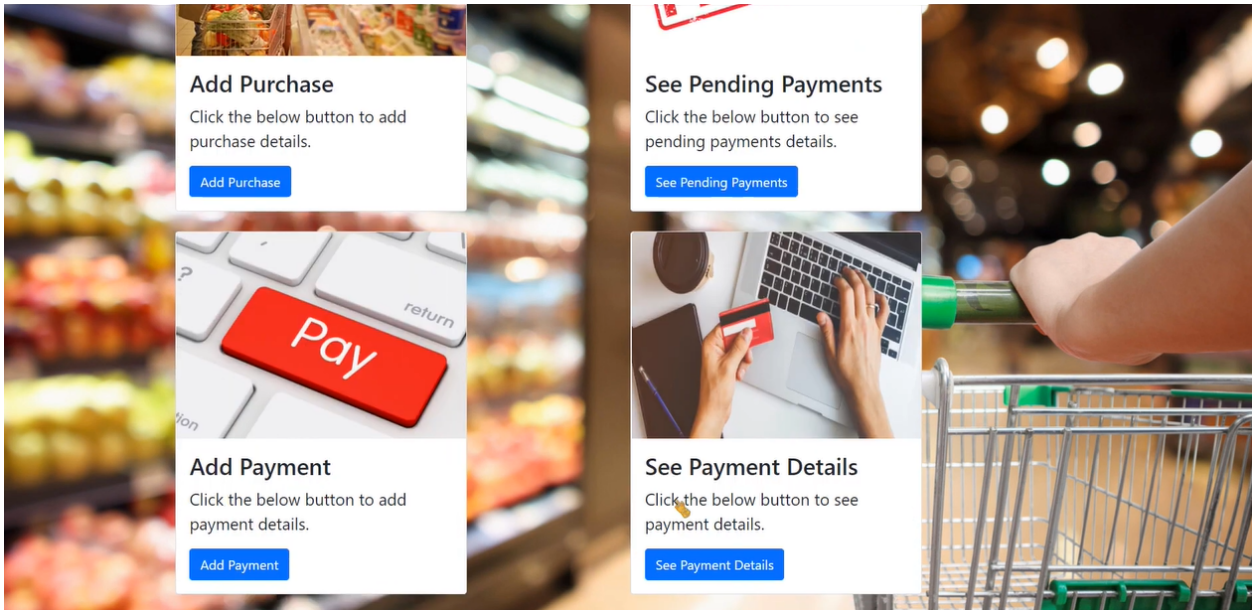

DIGITAL PAYMENT BOOK
Payment book application is designed to maintain customers, payments, and their purchases. A retailer will be an admin of the application and each customer of the retailer's shop will be the user. Customers can create their account in the payment book app and they will be able to see their purchase history, pending payments, and also if the user is having any doubt or complaint they can contact the retailer by using the contact our service. Admin will maintain data about purchases made by the customers & can see payment details and pending payment of the customers and will send alerts to the customers if there is any due for payment or payment is pending for a long time.
Manage Payments Easier!!!
REGISTER
LOGIN

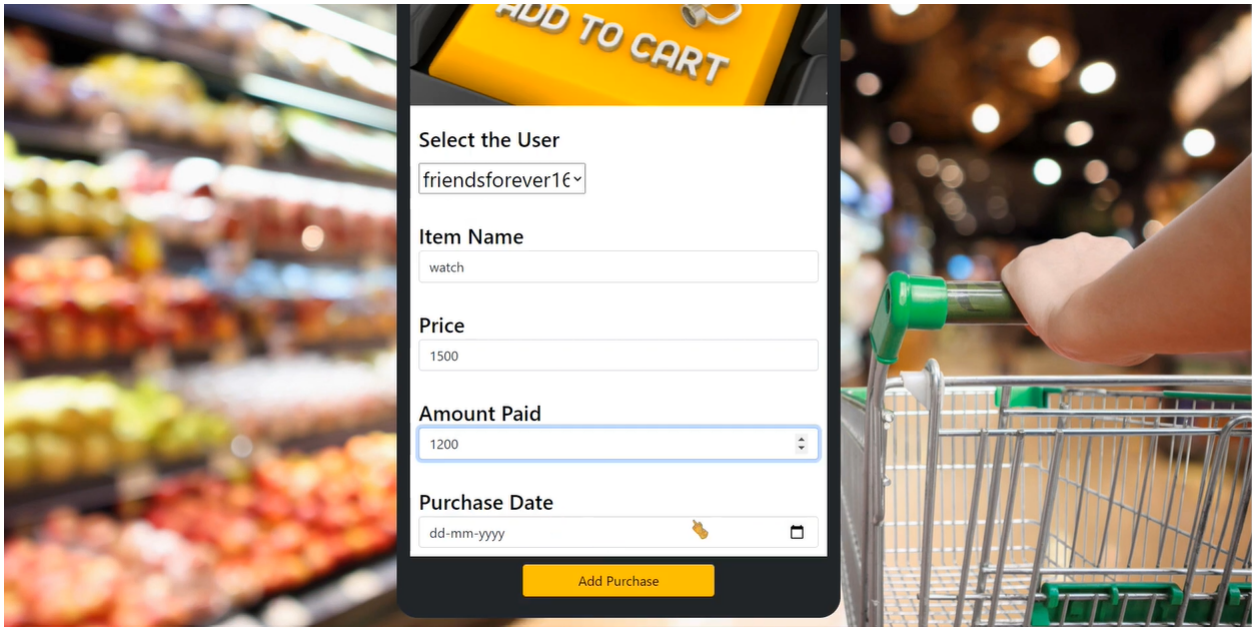

DIGITAL PAYMENT BOOK
SIGN UP
Sign Up
shift
Username
Enter Username
Email address
Enter email
Password
Password

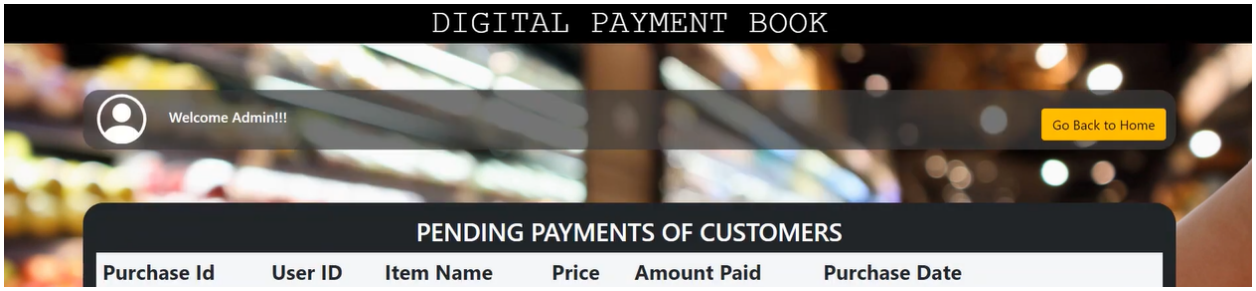

DIGITAL PAYMENT BOOK
LOGIN
Email
Enter email
Password
Password


PURCHASE HISTORY
Purchase Id
User ID
Item Name
Price
Amount Paid
Purchase Date
PENDING
See Pending Payments
Click th below button to see pending payments details.
See Pending Payments
See Payment Details
Click the below button to see payment details.
See Payment Details
CONTACT US
Contact Our Service
Click the below button to contact us.
Contact Us

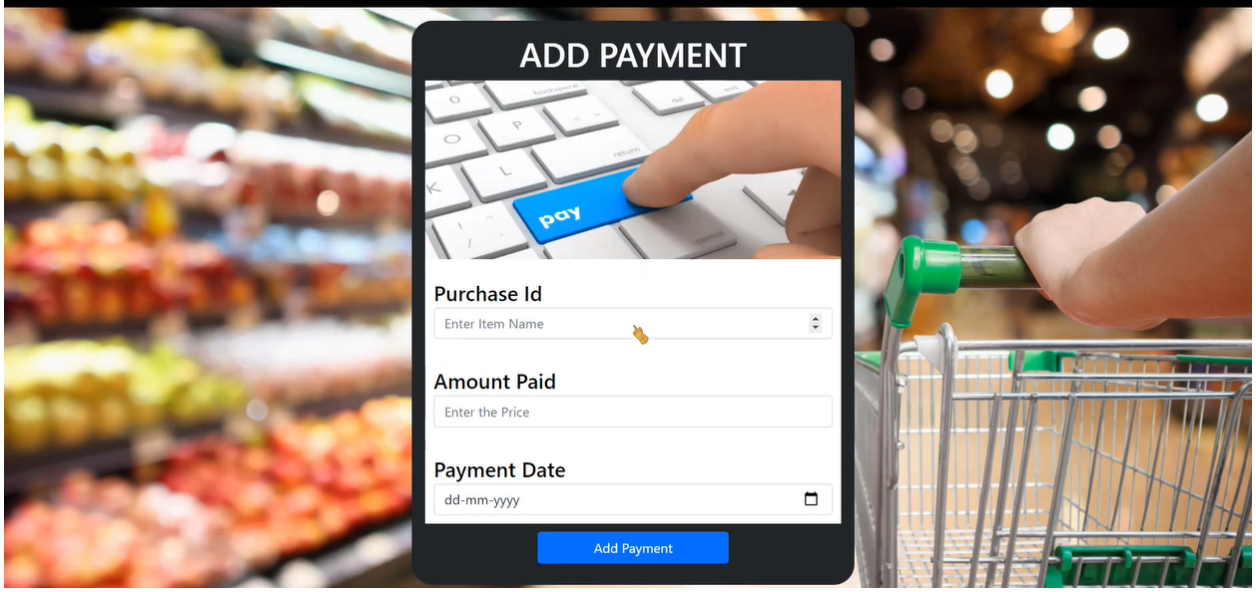

DIGITAL PAYMENT BOOK
Welcome Admin!!!
Logout
TOTAL PURCHASES
11
TOTAL PAYMENTS
10
TOTAL PENDING PAYMENTS
9
PURCHASE HISTORY
Purchase Id
User ID
Item Name
Price
Amount Paid
Purchase Date


Add Purchase
Click the below button to add purchase details.
Add Purchase
See Pending Payments
Click the below button to see pending payments details.
See Pending Payments
Pay
return
Add Payment
Click the below button to add payment details.
Add Payment
See Payment Details
Click the below button to see payment details.
See Payment Details


ADD TO CART
Select the User
friendsforever16
Item Name
watch
Price
1500
Amount Paid
1200
Purchase Date
dd-mm-yyyy
Add Purchase


DIGITAL PAYMENT BOOK
Welcome Admin!!!
Go Back to Home
PENDING PAYMENTS OF CUSTOMERS
Purchase Id
User ID
Item Name
Price
Amount Paid
Purchase Date

DIGITAL PAYMENT BOOK



Welcome {{username}}!!!

Go Back to Home

Contact Our Service

For any queries and information, contact us via email or mobile phone

Email : paymentbook.ad@gmail.com

Phone no : 7674897856