```
from flask import Flask, render_template, request, redirect, session ,url_for
import ibm_db
import re
import sendemail



app = Flask(__name__, template_folder = 'templates')


app.config['SECRET_KEY'] = ''
app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USERNAME'] = 'apikey'
app.config['MAIL_PASSWORD'] = ''
app.config['MAIL_DEFAULT_SENDER'] = 'abitha.thangadurai@gmail.com'


conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=55fbc997-9266-4331-afd3-
888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31929;SECURITY=SSL;SSLS
erverCertificate=DigiCertGlobalRootCA.crt;UID=spq76013;PWD=lcAFi0FUxoaVlpTn",'','')



#HOME--PAGE
@app.route("/home")
def home():
    return render_template("homepage.html")

@app.route("/")
def add():
    return render_template("home.html")
```

```
#SIGN--UP--OR--REGISTER



@app.route("/signup")
def signup():
    return render_template("signup.html")




@app.route('/register', methods =['GET', 'POST'])
def register():
    global user_email
    msg = ''
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        query = "SELECT * FROM register WHERE email=?;"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
```

```
        else:
            query = "INSERT INTO register values(?,?,?);"
            stmt = ibm_db.prepare(conn, query)
            ibm_db.bind_param(stmt, 1, username)
            ibm_db.bind_param(stmt, 2, email)
            ibm_db.bind_param(stmt, 3, password)
            ibm_db.execute(stmt)
            session['loggedin'] = True
            session['id'] = email
            user_email = email
            session['email'] = email
            session['username'] = username

            msg = 'You have successfully registered ! Proceed Login Process'
            return render_template('login.html', msg = msg)
    else:
        msg = 'PLEASE FILL OUT OF THE FORM'
        return render_template('register.html', msg=msg)



 #LOGIN--PAGE

@app.route("/signin")
def signin():
    return render_template('login.html')

@app.route('/login',methods =['GET', 'POST'])
def login():
    global user_email
    msg = ''
```

```
    if request.method == 'POST' :
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM register WHERE email =? AND password=?;"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print (account)

        if account:
            session['loggedin'] = True
            session['id'] = account['EMAIL']
            user_email=  account['EMAIL']
            session['email']=account['EMAIL']
            session['username'] = account['USERNAME']

            return redirect('/home')
        else:
            msg = 'Incorrect username / password !'
    return render_template('login.html', msg = msg)

#CHANGE FORGOT PASSWORD

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

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

```
def forgotpw():
    msg = ''
    if request.method == 'POST' :
        email = request.form['email']
        password = request.form['password']
        query = "SELECT * FROM register WHERE email=?;"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            query = "UPDATE register SET password = ? WHERE email = ?;"
            stmt = ibm_db.prepare(conn, query)
            ibm_db.bind_param(stmt, 1, password)
            ibm_db.bind_param(stmt, 2, email)
            ibm_db.execute(stmt)
            msg = 'Successfully changed your password ! Proceed Login Process'
            return render_template('login.html', msg = msg)
    else:
        msg = 'PLEASE FILL OUT THE CORRECT DETAILS'
        return render_template('forgot.html', msg=msg)


#ADDING----DATA


@app.route("/add")
def adding():
    return render_template('add.html')
```

```
@app.route('/addexpense',methods=['GET', 'POST'])
def addexpense():
    global user_email
    que = "SELECT * FROM expenses where id = ? ORDER BY 'dates' DESC"
    stm = ibm_db.prepare(conn, que)
    ibm_db.bind_param(stm, 1, session['email'])
    ibm_db.execute(stm)
    dictionary=ibm_db.fetch_assoc(stm)
    expense=[]
    while dictionary != False:

exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["AMOUNT"],dictio
nary["PAYMODE"],dictionary["CATEGORY"])
        expense.append(exp)
        dictionary = ibm_db.fetch_assoc(stm)
    i=len(expense)+1
    idx=str(i)
    dates = request.form['date']
    expensename = request.form['expensename']
    amount = request.form['amount']
    paymode = request.form['paymode']
    category = request.form['category']
    query = "INSERT INTO expenses VALUES (?,?,?,?,?,?,?);"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1, session['email'])
    ibm_db.bind_param(stmt, 2, dates)
    ibm_db.bind_param(stmt, 3, expensename)
    ibm_db.bind_param(stmt, 4, amount)
    ibm_db.bind_param(stmt, 5, paymode)
    ibm_db.bind_param(stmt, 6, category)
    ibm_db.bind_param(stmt, 7, idx)
```

```
    ibm_db.execute(stmt)
    print(dates + " " + expensename + " " + amount + " " + paymode + " " + category)

    return redirect("/display")



#DISPLAY---graph


@app.route("/display")
def display():
    query = "SELECT * FROM expenses where id = ? ;"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1, session['email'])
    ibm_db.execute(stmt)
    dictionary=ibm_db.fetch_assoc(stmt)
    rexpense=[]
    while dictionary != False:

exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["AMOUNT"],dictio
nary["PAYMODE"],dictionary["CATEGORY"],dictionary["IDX"])
        rexpense.append(exp)
        dictionary = ibm_db.fetch_assoc(stmt)
    que = "SELECT MONTH(dates) as DATES, SUM(amount) as AMOUNT FROM expenses WHERE id=?
AND YEAR(dates)= YEAR(now()) GROUP BY MONTH(dates);"
    stm = ibm_db.prepare(conn, que)
    ibm_db.bind_param(stm, 1,session['email'])
    ibm_db.execute(stm)
    dictionary=ibm_db.fetch_assoc(stm)
    texpense=[]
    while dictionary != False:
        exp=(dictionary["DATES"],dictionary["AMOUNT"])
```

```
        texpense.append(exp)
        dictionary = ibm_db.fetch_assoc(stm)
    print(texpense)

    quer = "SELECT * FROM expenses WHERE id = ? AND YEAR(dates)= YEAR(now());"
    st = ibm_db.prepare(conn, quer)
    ibm_db.bind_param(st, 1,session['email'])
    ibm_db.execute(st)
    dictionary=ibm_db.fetch_assoc(st)
    expense=[]
    while dictionary != False:

exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["AMOUNT"],dictio
nary["PAYMODE"],dictionary["CATEGORY"],dictionary["IDX"])
        expense.append(exp)
        dictionary = ibm_db.fetch_assoc(st)

    total=0
    t_food=0
    t_entertainment=0
    t_business=0
    t_rent=0
    t_EMI=0
    t_other=0


    for x in expense:
        total += x[3]
        if x[5] == "food":
            t_food += x[3]

        elif x[5] == "entertainment":
```

```
            t_entertainment  += x[3]

        elif x[5] == "business":
            t_business  += x[3]
        elif x[5] == "rent":
            t_rent  += x[3]

        elif x[5] == "EMI":
            t_EMI  += x[3]

        elif x[5] == "other":
            t_other  += x[3]

    print(total)

    print(t_food)
    print(t_entertainment)
    print(t_business)
    print(t_rent)
    print(t_EMI)
    print(t_other)

    qur = "SELECT * FROM expenses WHERE id = ? AND MONTH(dates)= MONTH(now());"
    stt = ibm_db.prepare(conn, qur)
    ibm_db.bind_param(stt, 1, session['email'])
    ibm_db.execute(stt)
    dictionary=ibm_db.fetch_assoc(stt)
    lexpense=[]
    while dictionary != False:

exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["AMOUNT"],dictio
nary["PAYMODE"],dictionary["CATEGORY"],dictionary["IDX"])
```

```
    lexpense.append(exp)
    dictionary = ibm_db.fetch_assoc(stt)

ttotal=0
to_food=0
to_entertainment=0
to_business=0
to_rent=0
to_EMI=0
to_other=0



for x in lexpense:
    ttotal += x[3]
    if x[5] == "food":
        to_food += x[3]

    elif x[5] == "entertainment":
        to_entertainment  += x[3]

    elif x[5] == "business":
        to_business  += x[3]
    elif x[5] == "rent":
        to_rent  += x[3]

    elif x[5] == "EMI":
        to_EMI  += x[3]

    elif x[5] == "other":
        to_other  += x[3]
```

```
print(ttotal)



qy = "SELECT max(IDX) as IDX FROM limits where id=?;"
smt = ibm_db.prepare(conn, qy)
ibm_db.bind_param(smt, 1, session['email'])
ibm_db.execute(smt)
dictionary = ibm_db.fetch_assoc(smt)
uexpense=[]
while dictionary != False:
   exp=(dictionary["IDX"])
   uexpense.append(exp)
   dictionary = ibm_db.fetch_assoc(smt)
k=uexpense[0]
qu = "SELECT NUMBER FROM limits where id=? and idx=?"
sm = ibm_db.prepare(conn, qu)
ibm_db.bind_param(sm, 1, session['email'])
ibm_db.bind_param(sm, 2, k)
ibm_db.execute(sm)
dictionary = ibm_db.fetch_assoc(sm)
fexpense=[]
while dictionary != False:
   exp=(dictionary["NUMBER"])
   fexpense.append(exp)
   dictionary = ibm_db.fetch_assoc(stmt)


if len(fexpense) <= 0:
   print("Enter the limit First")
else:
   if ttotal > fexpense[0]:
      m=sendemail.sendgridmail(session["email"])
```

```
        print(m)
    else: print("Error")
  return render_template("display.html",rexpense=rexpense, texpense = texpense, expense = expense,  total = total ,
                    t_food = t_food,t_entertainment =  t_entertainment,
                    t_business = t_business,  t_rent =  t_rent,
                    t_EMI =  t_EMI,  t_other =  t_other )


#delete---the--data

@app.route('/delete/<idx>', methods = ['POST', 'GET' ])
def delete(idx):
  query = "DELETE FROM expenses WHERE id=? and idx=?;"
  stmt = ibm_db.prepare(conn, query)
  ibm_db.bind_param(stmt, 1, session["email"])
  ibm_db.bind_param(stmt, 2, idx)
  ibm_db.execute(stmt)
  print('deleted successfully')
  return render_template("display.html")



#UPDATE---DATA

@app.route('/edit/<id>', methods = ['POST', 'GET' ])
def edit(id):
  query = "SELECT * FROM expenses WHERE id=? and idx=?;"
  stmt = ibm_db.prepare(conn, query)
  ibm_db.bind_param(stmt, 1, session['email'])
  ibm_db.bind_param(stmt, 2, id)
```

```
    ibm_db.execute(stmt)
    dictionary=ibm_db.fetch_assoc(stmt)
    expense=[]
    while dictionary != False:

exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["AMOUNT"],dictio
nary["PAYMODE"],dictionary["CATEGORY"],dictionary["IDX"])
        expense.append(exp)
        dictionary = ibm_db.fetch_assoc(stmt)
    print(expense)
    return render_template('edit.html', expenses = expense[0])




@app.route('/update/<id>', methods = ['POST'])
def update(id):
  if request.method == 'POST' :
      dates = request.form['date']
      expensename = request.form['expensename']
      amount = request.form['amount']
      paymode = request.form['paymode']
      category = request.form['category']
      query = "UPDATE expenses SET dates = ? , expensename = ? , amount = ?, paymode = ?, category
= ? WHERE id = ? and idx=?;"
      stmt = ibm_db.prepare(conn, query)
      ibm_db.bind_param(stmt, 1, dates)
      ibm_db.bind_param(stmt, 2, expensename)
      ibm_db.bind_param(stmt, 3, amount)
      ibm_db.bind_param(stmt, 4, paymode)
      ibm_db.bind_param(stmt, 5, category)
      ibm_db.bind_param(stmt, 6, session['email'])
```

```
    ibm_db.bind_param(stmt, 7, id)
    ibm_db.execute(stmt)

    print('successfully updated')
    return redirect("/display")
```

```
 #limit
@app.route("/limit" )
def limit():
     return render_template('limit.html')

@app.route("/limitnum" , methods = ['POST' ])
def limitnum():
   que = "SELECT * FROM limits where id = ? ;"
   stm = ibm_db.prepare(conn, que)
   ibm_db.bind_param(stm, 1, session['email'])
   ibm_db.execute(stm)
   if request.method == "POST":
      dictionary=ibm_db.fetch_assoc(stm)
      expense=[]
      while dictionary != False:
         exp=(dictionary['ID'],dictionary['NUMBER'],dictionary['IDX'])
         expense.append(exp)
         dictionary = ibm_db.fetch_assoc(stm)
```

```
        i=len(expense)+1
        idx=str(i)
        number= request.form['number']
        query = "INSERT INTO limits VALUES(?,?,?)"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt, 1, session['email'])
        ibm_db.bind_param(stmt, 2, number)
        ibm_db.bind_param(stmt, 3, idx)
        ibm_db.execute(stmt)
        return redirect('/limitn')



@app.route("/limitn")
def limitn():
    query = "SELECT max(IDX) as IDX FROM limits where id=?;"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1, session['email'])
    ibm_db.execute(stmt)
    dictionary = ibm_db.fetch_assoc(stmt)
    expense=[]
    while dictionary != False:
        exp=(dictionary["IDX"])
        expense.append(exp)
        dictionary = ibm_db.fetch_assoc(stmt)
    k=expense[0]
    que = "SELECT NUMBER FROM limits where id=? and idx=?"
    stmt = ibm_db.prepare(conn, que)
    ibm_db.bind_param(stmt, 1, session['email'])
    ibm_db.bind_param(stmt, 2, k)
    ibm_db.execute(stmt)
    dictionary = ibm_db.fetch_assoc(stmt)
```

```
    texpense=[]
    while dictionary != False:
        exp=(dictionary["NUMBER"])
        texpense.append(exp)
        dictionary = ibm_db.fetch_assoc(stmt)
    s=texpense[0]
    return render_template("limit.html" , y= s)



#REPORT


@app.route("/today")
def today():
    query = "SELECT dates, amount FROM expenses  WHERE id = ? AND DATE(dates) = DATE(NOW());
"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1, str(session['email']))
    ibm_db.execute(stmt)
    dictionary=ibm_db.fetch_assoc(stmt)
    texpense=[]
    while dictionary != False:
      exp=(dictionary["DATES"],dictionary["AMOUNT"])
      texpense.append(exp)
      dictionary = ibm_db.fetch_assoc(stmt)
    print(texpense)


    query = "SELECT * FROM expenses WHERE id = ? AND DATE(dates) = DATE(NOW())"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1, session['email'])
    ibm_db.execute(stmt)
    dictionary=ibm_db.fetch_assoc(stmt)
```

```
expense=[]
while dictionary != False:
    exp=(dictionary["AMOUNT"],dictionary["PAYMODE"],dictionary["CATEGORY"])
    expense.append(exp)
    dictionary = ibm_db.fetch_assoc(stmt)

total=0
t_food=0
t_entertainment=0
t_business=0
t_rent=0
t_EMI=0
t_other=0


for x in expense:
   total += x[0]
   if x[2] == "food":
      t_food += x[0]

   elif x[2] == "entertainment":
      t_entertainment  += x[0]

   elif x[2] == "business":
      t_business  += x[0]
   elif x[2] == "rent":
      t_rent  += x[0]

   elif x[2] == "EMI":
      t_EMI  += x[0]
```

```
        elif x[2] == "other":
            t_other  += x[0]

    print(total)

    print(t_food)
    print(t_entertainment)
    print(t_business)
    print(t_rent)
    print(t_EMI)
    print(t_other)

    return render_template("today.html", texpense = texpense, expense = expense,  total = total ,
                    t_food = t_food,t_entertainment =  t_entertainment,
                    t_business = t_business,  t_rent =  t_rent,
                    t_EMI =  t_EMI,  t_other =  t_other )

@app.route("/month")
def month():
    query = "SELECT dates, SUM(amount) as AMOUNT FROM expenses WHERE id= ? AND
MONTH(dates)= MONTH(now()) GROUP BY dates ORDER BY dates;"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1, str(session['email']))
    ibm_db.execute(stmt)
    dictionary=ibm_db.fetch_assoc(stmt)
    texpense=[]
    while dictionary != False:
      exp=(dictionary["DATES"],dictionary["AMOUNT"])
```

```
    texpense.append(exp)
    dictionary = ibm_db.fetch_assoc(stmt)
  print(texpense)

  query = "SELECT * FROM expenses WHERE id = ? AND MONTH(dates)= MONTH(now());"
  stmt = ibm_db.prepare(conn, query)
  ibm_db.bind_param(stmt, 1, session['email'])
  ibm_db.execute(stmt)
  dictionary=ibm_db.fetch_assoc(stmt)
  expense=[]
  while dictionary != False:

exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["AMOUNT"],dictionary["PAYMODE"],dictionary["CATEGORY"],dictionary["IDX"])
    expense.append(exp)
    dictionary = ibm_db.fetch_assoc(stmt)

  total=0
  t_food=0
  t_entertainment=0
  t_business=0
  t_rent=0
  t_EMI=0
  t_other=0


  for x in expense:
    total += x[3]
    if x[5] == "food":
      t_food += x[3]

    elif x[5] == "entertainment":
```

```
            t_entertainment  += x[3]

        elif x[5] == "business":
            t_business  += x[3]
        elif x[5] == "rent":
            t_rent  += x[3]

        elif x[5] == "EMI":
            t_EMI  += x[3]

        elif x[5] == "other":
            t_other  += x[3]

    print(total)

    print(t_food)
    print(t_entertainment)
    print(t_business)
    print(t_rent)
    print(t_EMI)
    print(t_other)



    return render_template("today.html", texpense = texpense, expense = expense,  total = total ,
                    t_food = t_food,t_entertainment =  t_entertainment,
                    t_business = t_business,  t_rent =  t_rent,
                    t_EMI =  t_EMI,  t_other =  t_other )

@app.route("/year")
def year():
```

```
    query = "SELECT MONTH(dates) as DATES, SUM(amount) as AMOUNT FROM expenses WHERE id=? AND YEAR(dates)= YEAR(now()) GROUP BY MONTH(dates);"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1,session['email'])
    ibm_db.execute(stmt)
    dictionary=ibm_db.fetch_assoc(stmt)
    texpense=[]
    while dictionary != False:
     exp=(dictionary["DATES"],dictionary["AMOUNT"])
     texpense.append(exp)
     dictionary = ibm_db.fetch_assoc(stmt)
    print(texpense)

    query = "SELECT * FROM expenses WHERE id = ? AND YEAR(dates)= YEAR(now());"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt, 1,session['email'])
    ibm_db.execute(stmt)
    dictionary=ibm_db.fetch_assoc(stmt)
    expense=[]
    while dictionary != False:

exp=(dictionary["ID"],dictionary["DATES"],dictionary["EXPENSENAME"],dictionary["AMOUNT"],dictionary["PAYMODE"],dictionary["CATEGORY"],dictionary["IDX"])
      expense.append(exp)
      dictionary = ibm_db.fetch_assoc(stmt)

    total=0
    t_food=0
    t_entertainment=0
    t_business=0
    t_rent=0
    t_EMI=0
```

```
t_other=0



for x in expense:
    total += x[3]
    if x[5] == "food":
        t_food += x[3]

    elif x[5] == "entertainment":
        t_entertainment  += x[3]

    elif x[5] == "business":
        t_business  += x[3]
    elif x[5] == "rent":
        t_rent  += x[3]

    elif x[5] == "EMI":
        t_EMI  += x[3]

    elif x[5] == "other":
        t_other  += x[3]

print(total)

print(t_food)
print(t_entertainment)
print(t_business)
print(t_rent)
print(t_EMI)
print(t_other)
```

```
    return render_template("today.html", texpense = texpense, expense = expense,  total = total ,
                    t_food = t_food,t_entertainment =  t_entertainment,
                    t_business = t_business,  t_rent =  t_rent,
                    t_EMI =  t_EMI,  t_other =  t_other )

#log-out

@app.route('/logout')

def logout():
  session.pop('loggedin', None)
  session.pop('id', None)
  session.pop('username', None)
  return render_template('home.html')



if __name__ == "__main__":
   app.run(debug=True)
```