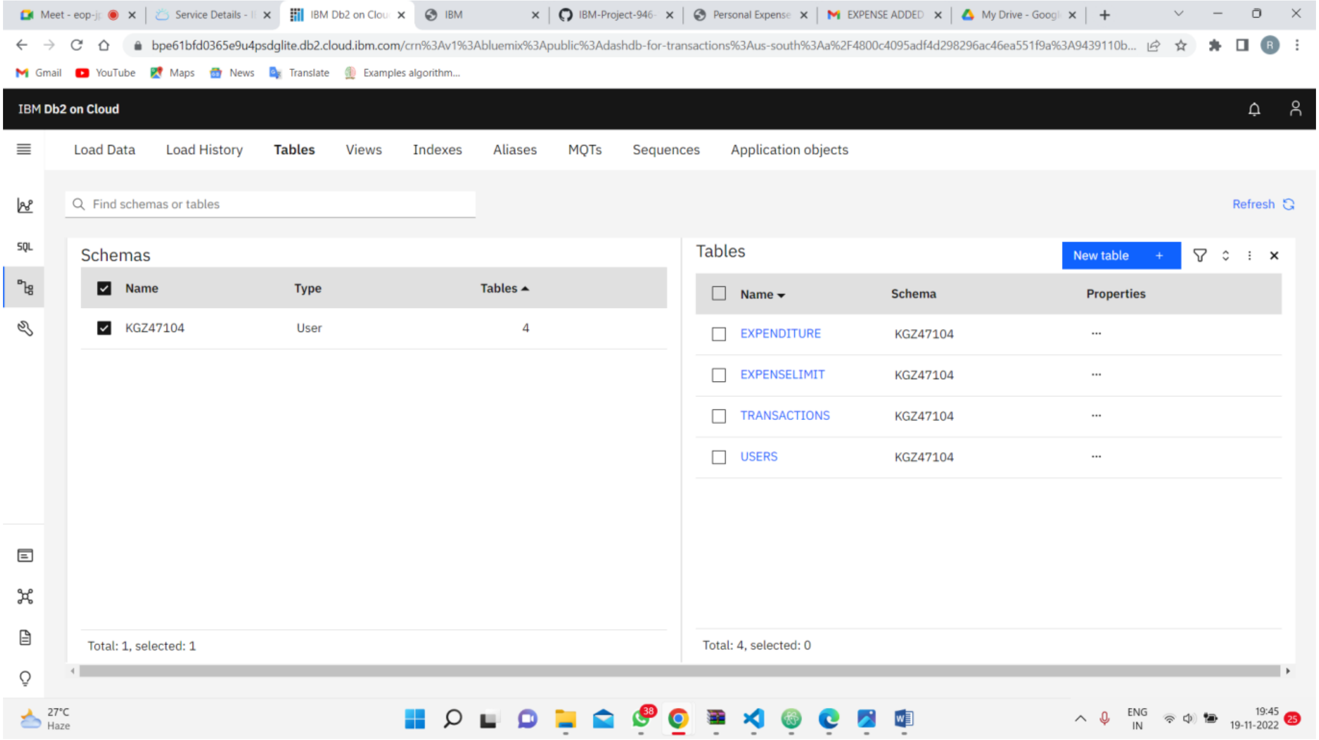
PROJECT DEPLOYMENT PHASE

SPRINT - II :

Team Id : PNT2022TMID26966
Project Name : PERSONAL EXPENSE TRACKER

TABLES :



Schemas

☑	Name	Type	Tables ▲
☑	KGZ47104	User	4

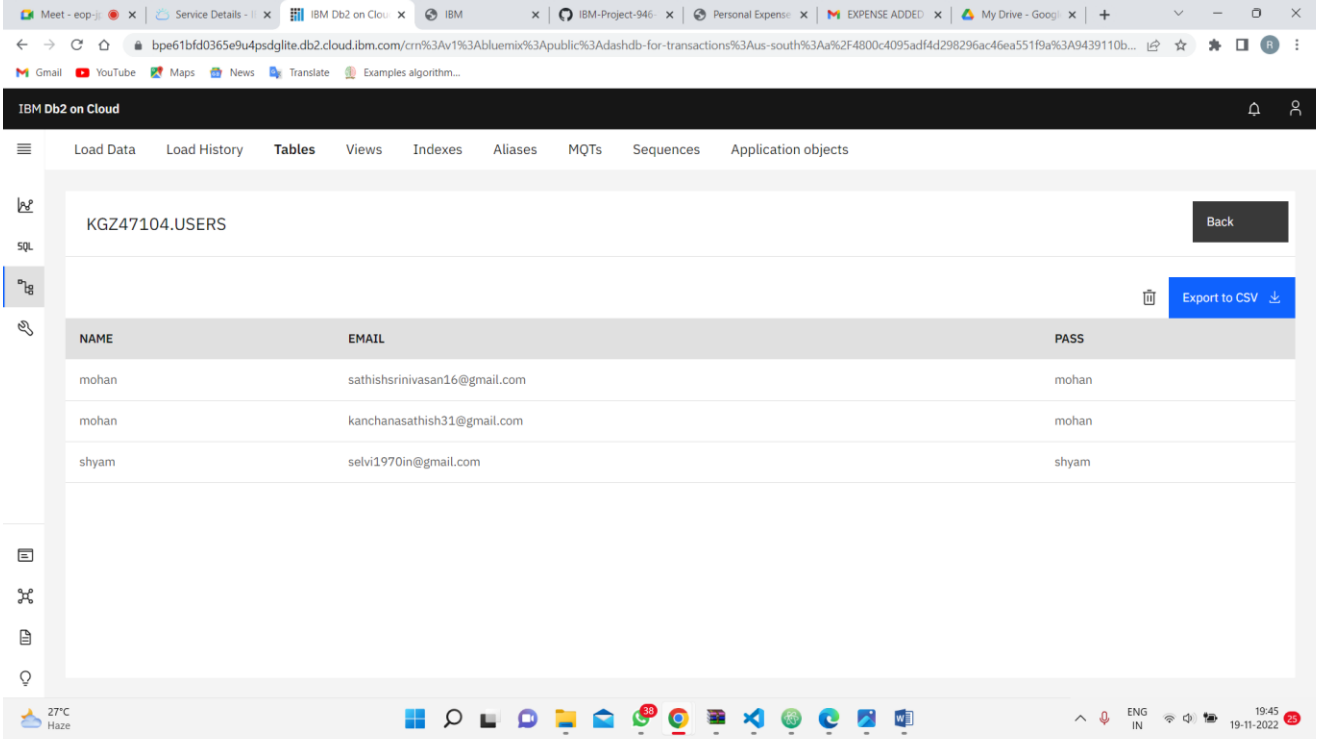
Total: 1, selected: 1

Tables

□	Name ▾	Schema	Properties
□	EXPENDITURE	KGZ47104	···
□	EXPENSELIMIT	KGZ47104	···
□	TRANSACTIONS	KGZ47104	···
□	USERS	KGZ47104	···

Total: 4, selected: 0

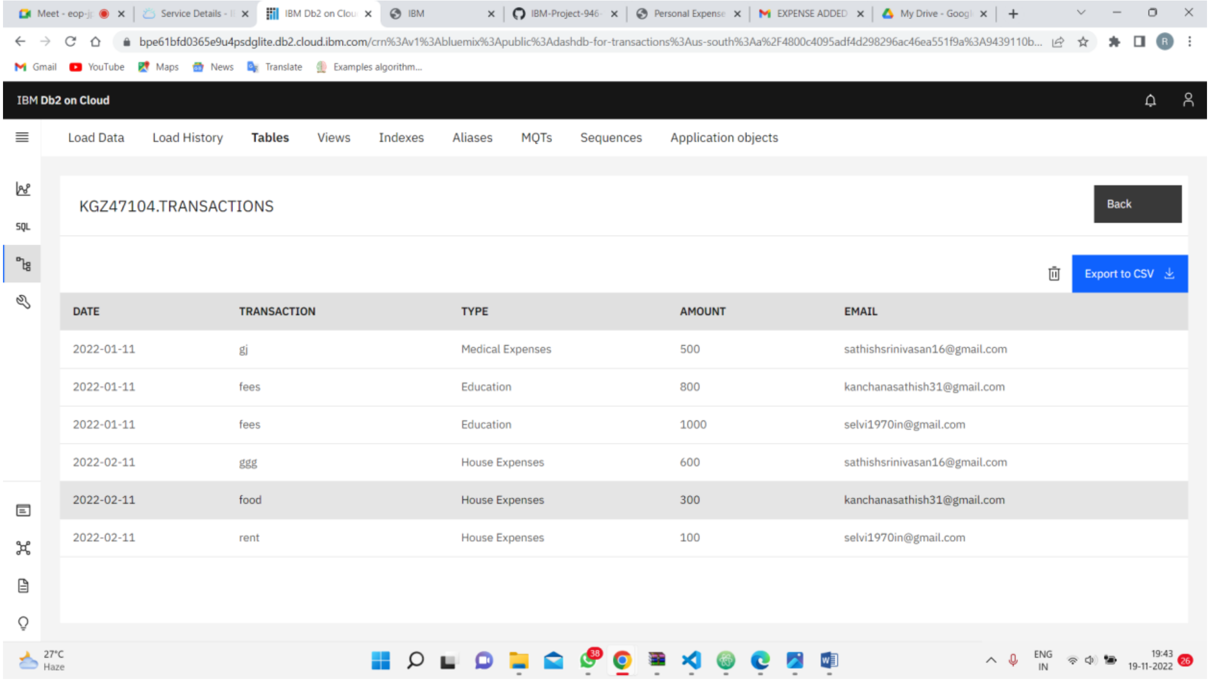
USER TABLE :



KGZ47104.USERS

NAME	EMAIL	PASS
mohan	sathishsrinivasan16@gmail.com	mohan
mohan	kanchanasathish31@gmail.com	mohan
shyam	selvi1970in@gmail.com	shyam

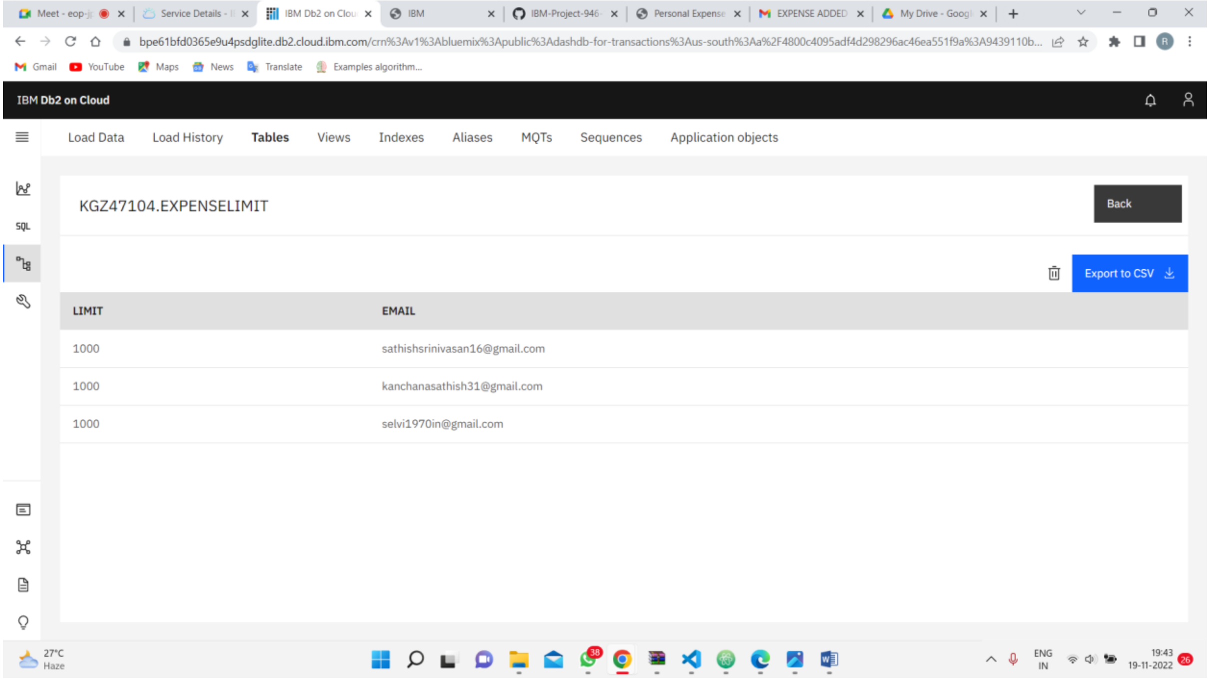
TRANSACTION TABLE :



KGZ47104.TRANSACTIONS

DATE	TRANSACTION	TYPE	AMOUNT	EMAIL
2022-01-11	gj	Medical Expenses	500	sathishsrinivasan16@gmail.com
2022-01-11	fees	Education	800	kanchanasathish31@gmail.com
2022-01-11	fees	Education	1000	selvi1970in@gmail.com
2022-02-11	ggg	House Expenses	600	sathishsrinivasan16@gmail.com
2022-02-11	food	House Expenses	300	kanchanasathish31@gmail.com
2022-02-11	rent	House Expenses	100	selvi1970in@gmail.com

EXPENSELIMIT



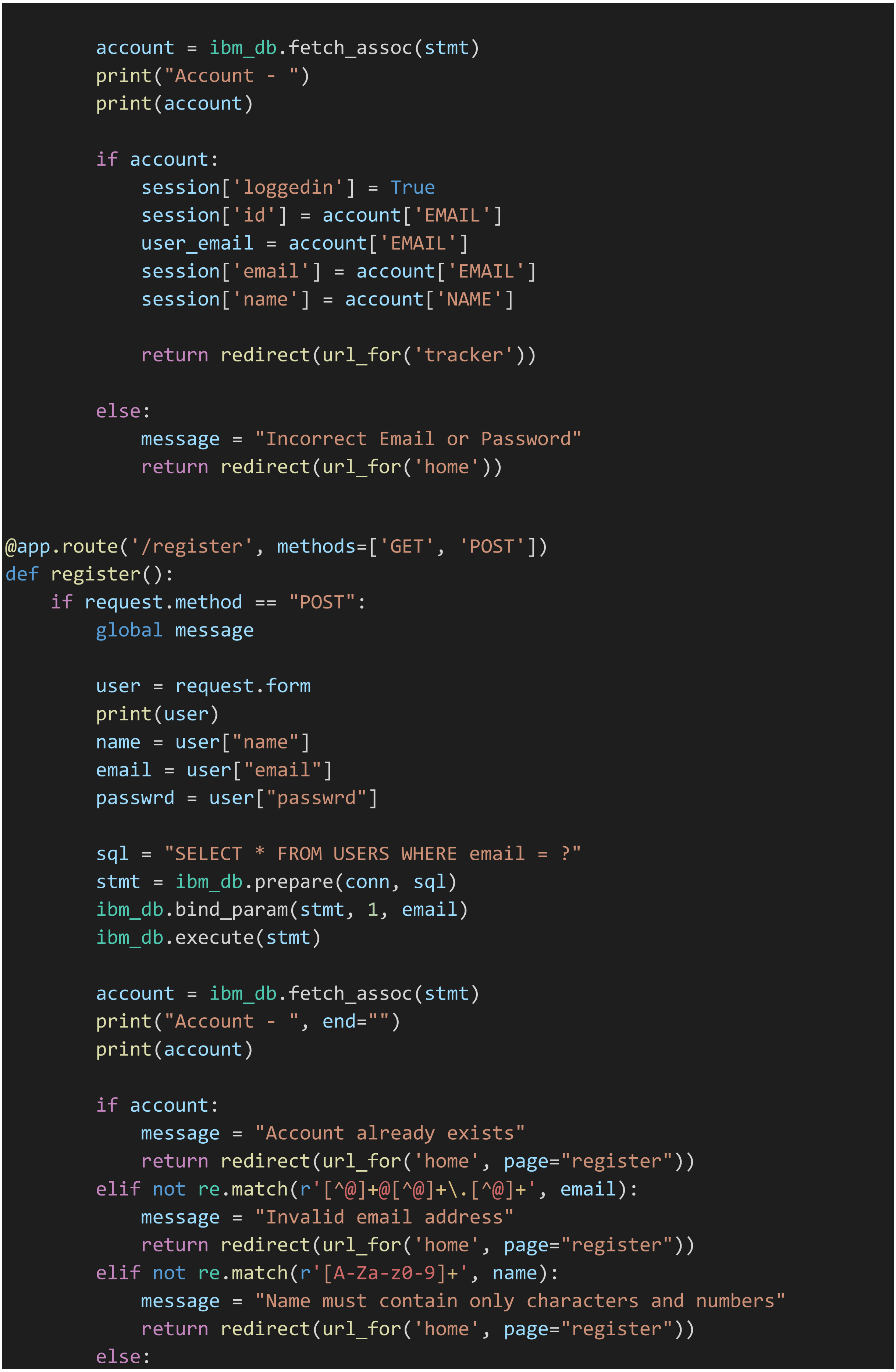
KGZ47104.EXPENSELIMIT

LIMIT	EMAIL
1000	sathishsrinivasan16@gmail.com
1000	kanchanasathish31@gmail.com
1000	selvi1970in@gmail.com

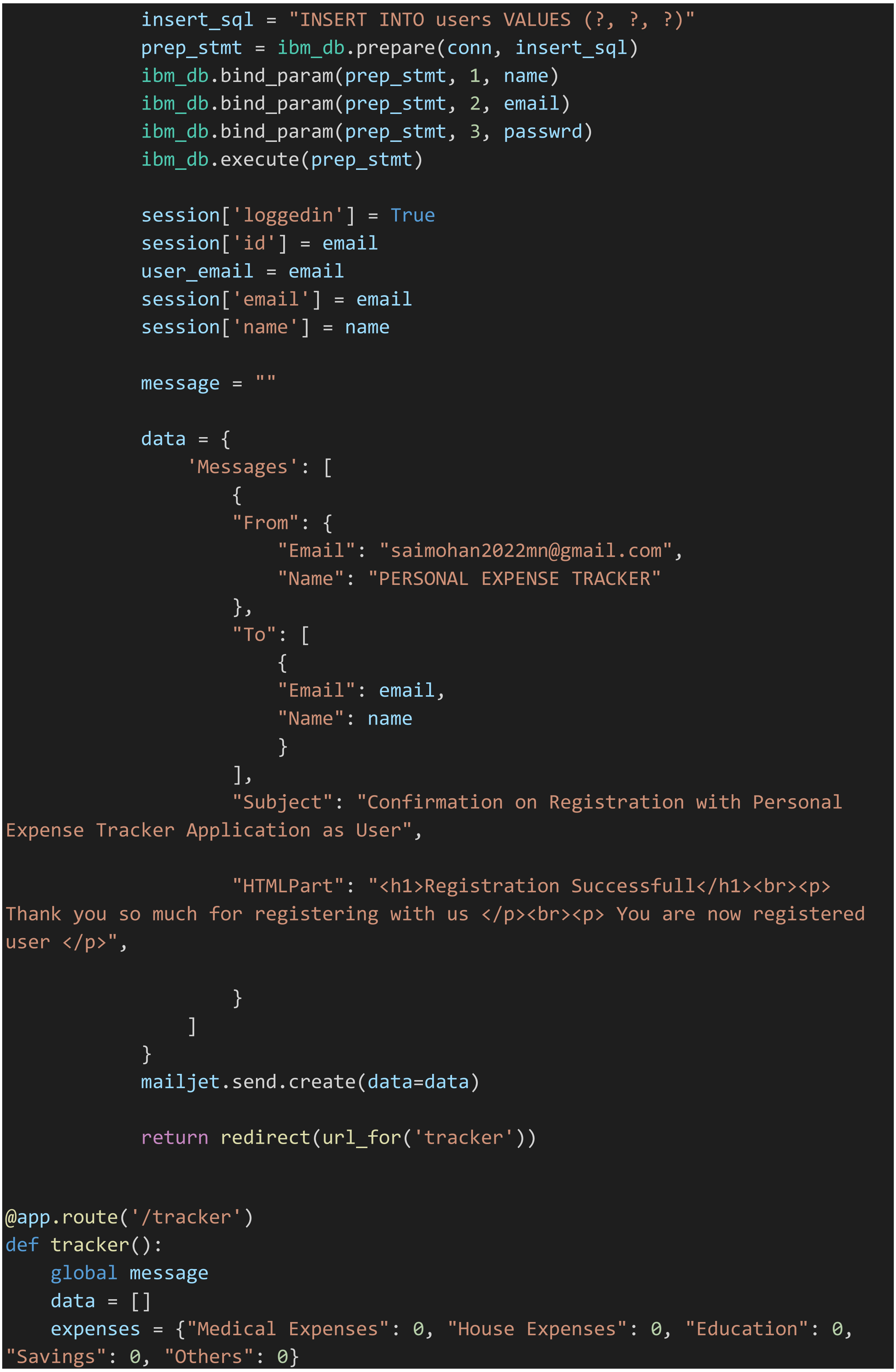
SOURCE CODE :

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

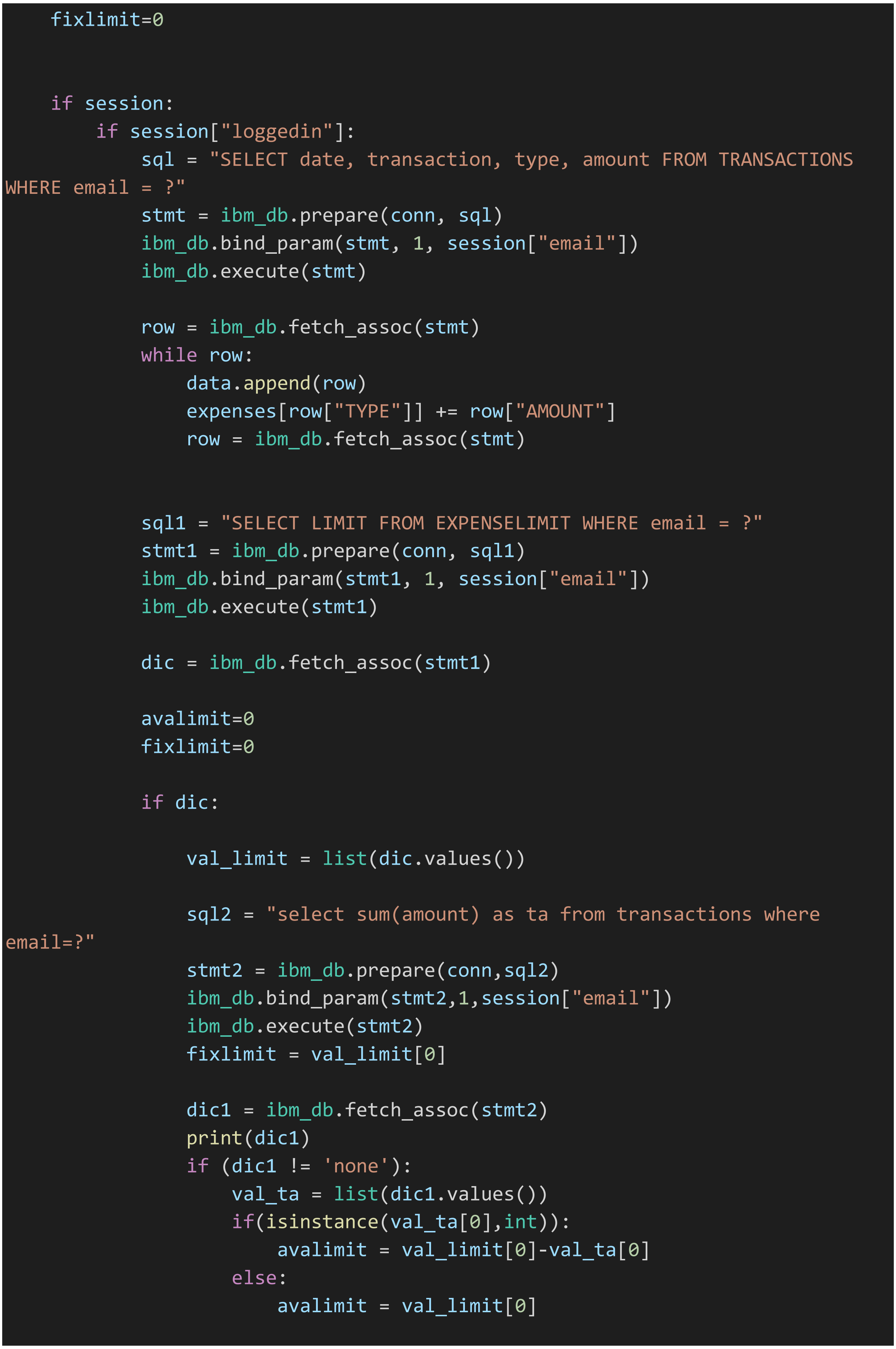
```python
from mailjet_rest import Client
import os
api_key = '8fea4c85e7f2268c6aee7877f645f032'
api_secret = 'ec568bbd44bebbaec71f0281f4c68966'
mailjet = Client(auth=(api_key, api_secret), version='v3.1')

app = Flask(__name__)
app.secret_key = "ibm"

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=kgz47104;PWD=n1CWPdWt76M5DoWw",'','')

message = ""

@app.route('/', methods=['GET', 'POST'])
def home():
    print(session)
    print("Message - " + message)
    if session:
        if session["loggedin"]:
            return redirect(url_for('tracker'))
    else:
        login_page = True
        print(request.values.get('page'))
        if request.values.get('page') == "register":
            login_page = False
        return render_template('index.html', login=login_page, message=message)


@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == "POST":
        global message

        user = request.form
        print(user)
        email = user["email"]
        passwrd = user["passwrd"]

        print("Email - " + email + ", Password - " + passwrd)

        sql = "SELECT * FROM users WHERE email = ? AND pass = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.bind_param(stmt, 2, passwrd)
        ibm_db.execute(stmt)
```



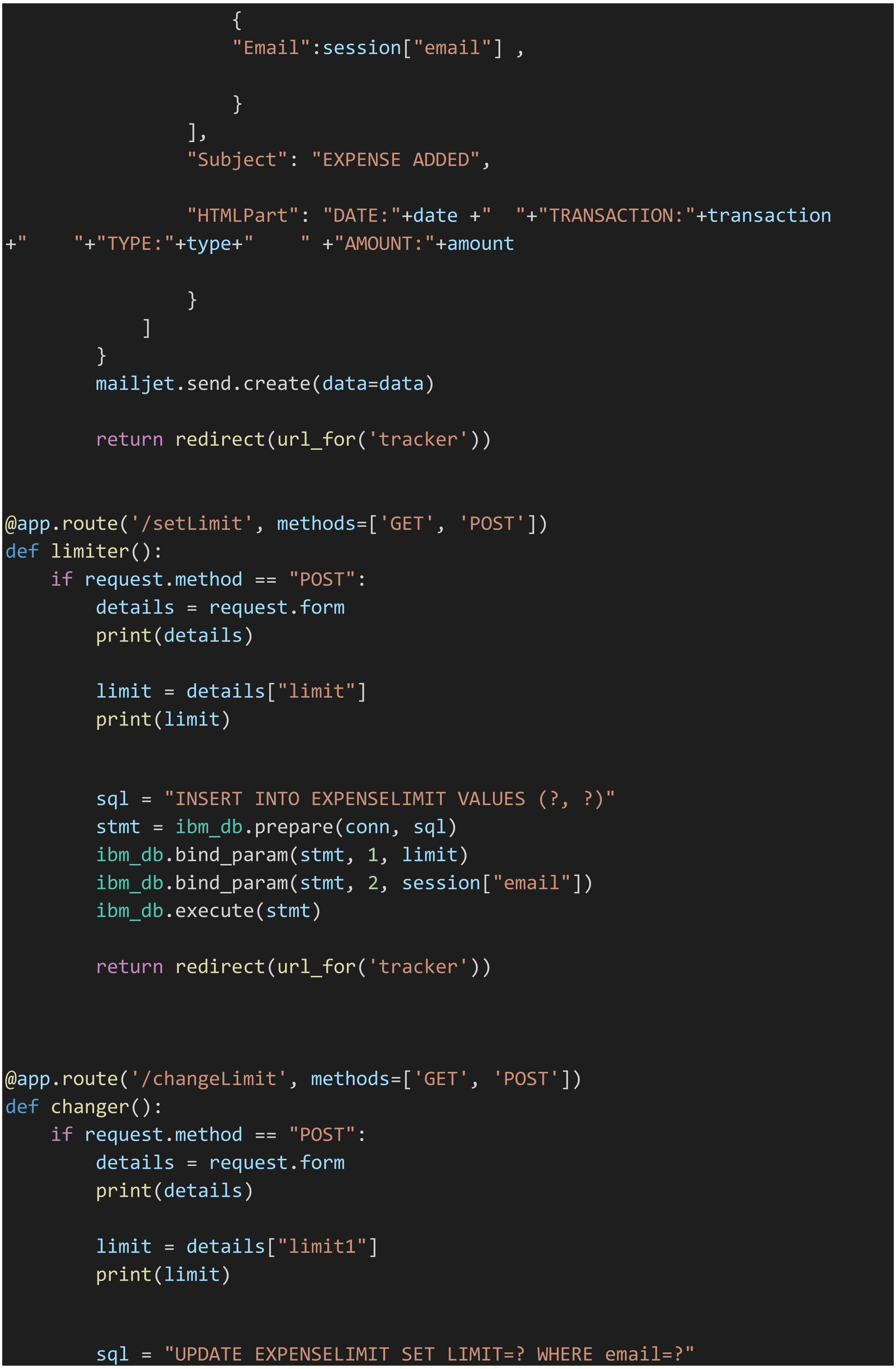
```
        account = ibm_db.fetch_assoc(stmt)
        print("Account - ")
        print(account)

        if account:
            session['loggedin'] = True
            session['id'] = account['EMAIL']
            user_email = account['EMAIL']
            session['email'] = account['EMAIL']
            session['name'] = account['NAME']

            return redirect(url_for('tracker'))

        else:
            message = "Incorrect Email or Password"
            return redirect(url_for('home'))


@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == "POST":
        global message

        user = request.form
        print(user)
        name = user["name"]
        email = user["email"]
        passwrd = user["passwrd"]

        sql = "SELECT * FROM USERS WHERE email = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.execute(stmt)

        account = ibm_db.fetch_assoc(stmt)
        print("Account - ", end="")
        print(account)

        if account:
            message = "Account already exists"
            return redirect(url_for('home', page="register"))
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            message = "Invalid email address"
            return redirect(url_for('home', page="register"))
        elif not re.match(r'[A-Za-z0-9]+', name):
            message = "Name must contain only characters and numbers"
            return redirect(url_for('home', page="register"))
        else:
```



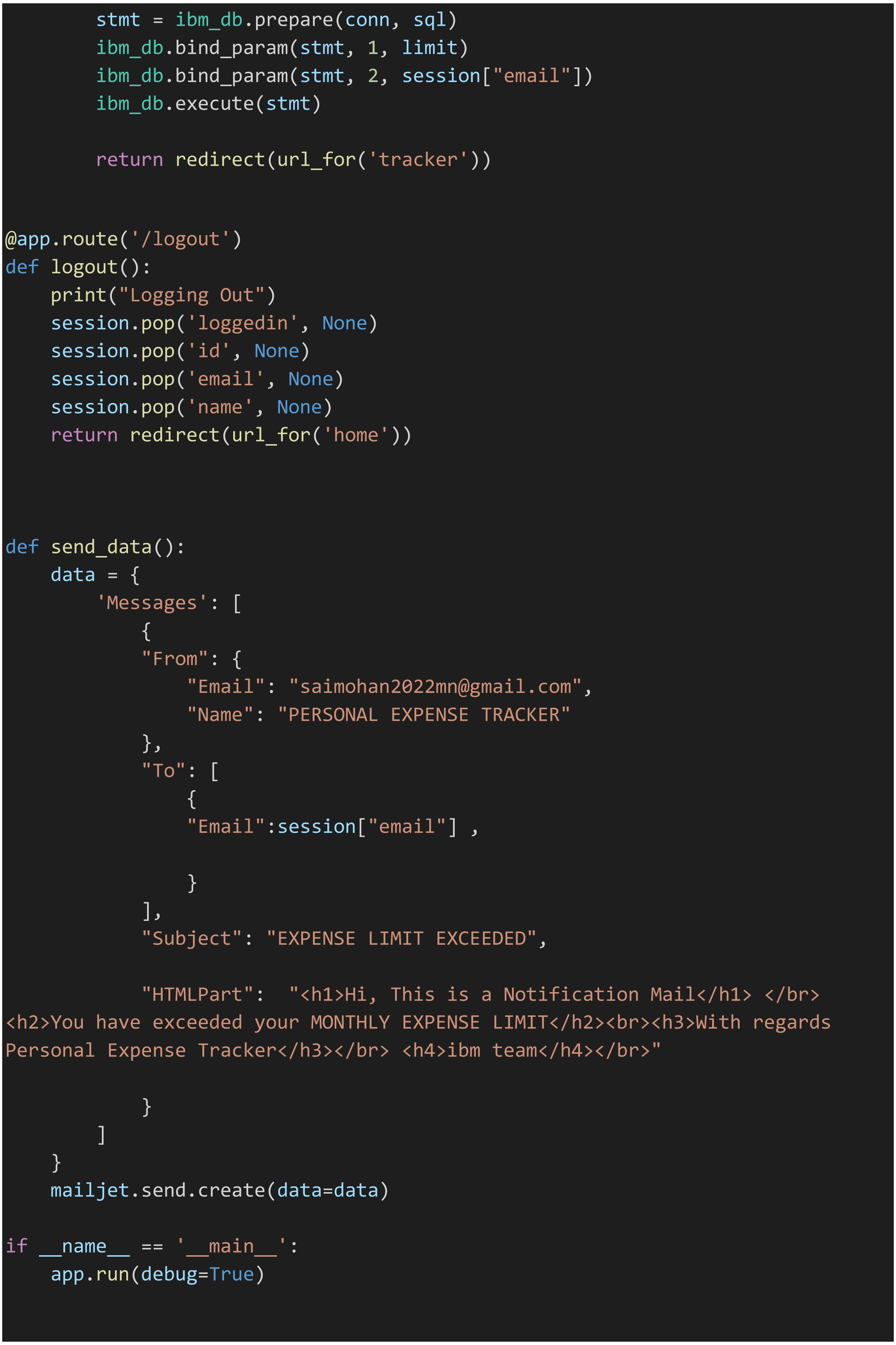
```
            insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, passwrd)
            ibm_db.execute(prep_stmt)

            session['loggedin'] = True
            session['id'] = email
            user_email = email
            session['email'] = email
            session['name'] = name

            message = ""

            data = {
                'Messages': [
                    {
                    "From": {
                        "Email": "saimohan2022mn@gmail.com",
                        "Name": "PERSONAL EXPENSE TRACKER"
                    },
                    "To": [
                        {
                        "Email": email,
                        "Name": name
                        }
                    ],
                    "Subject": "Confirmation on Registration with Personal 
Expense Tracker Application as User",

                    "HTMLPart": "<h1>Registration Successfull</h1><br><p> 
Thank you so much for registering with us </p><br><p> You are now registered 
user </p>",

                    }
                ]
            }
            mailjet.send.create(data=data)

            return redirect(url_for('tracker'))


@app.route('/tracker')
def tracker():
    global message
    data = []
    expenses = {"Medical Expenses": 0, "House Expenses": 0, "Education": 0, 
"Savings": 0, "Others": 0}
```



```
    fixlimit=0


    if session:
        if session["loggedin"]:
            sql = "SELECT date, transaction, type, amount FROM TRANSACTIONS
WHERE email = ?"
            stmt = ibm_db.prepare(conn, sql)
            ibm_db.bind_param(stmt, 1, session["email"])
            ibm_db.execute(stmt)

            row = ibm_db.fetch_assoc(stmt)
            while row:
                data.append(row)
                expenses[row["TYPE"]] += row["AMOUNT"]
                row = ibm_db.fetch_assoc(stmt)


            sql1 = "SELECT LIMIT FROM EXPENSELIMIT WHERE email = ?"
            stmt1 = ibm_db.prepare(conn, sql1)
            ibm_db.bind_param(stmt1, 1, session["email"])
            ibm_db.execute(stmt1)

            dic = ibm_db.fetch_assoc(stmt1)

            avalimit=0
            fixlimit=0

            if dic:

                val_limit = list(dic.values())

                sql2 = "select sum(amount) as ta from transactions where
email=?"
                stmt2 = ibm_db.prepare(conn,sql2)
                ibm_db.bind_param(stmt2,1,session["email"])
                ibm_db.execute(stmt2)
                fixlimit = val_limit[0]

                dic1 = ibm_db.fetch_assoc(stmt2)
                print(dic1)
                if (dic1 != 'none'):
                    val_ta = list(dic1.values())
                    if(isinstance(val_ta[0],int)):
                        avalimit = val_limit[0]-val_ta[0]
                    else:
                        avalimit = val_limit[0]
                    if(avalimit < 0):
                        print("\n\nFrom add expenditure :",avalimit)
                        # send_Email(session['email'])
                        send_data()

            print(data)
            print(expenses)
            print(avalimit)

            message = ""

            return render_template('home.html', name=session['name'],
data=data[::-1], expenses=expenses,avalimit=avalimit,fixlimit=fixlimit)
    else:
        message = "Session Expired"
    return redirect(url_for("home"))


@app.route('/add-expenditure', methods=['GET', 'POST'])
def add_expenditure():
    if request.method == "POST":
        details = request.form
        print(details)

        date = details["date"][-2:] + "/" + details["date"][5:7] + "/" +
details["date"][:4]
        transaction = details["transaction"]
        type = details["type"]
        amount = details["amount"]
        print(date, transaction, type, amount)

        sql = "INSERT INTO transactions VALUES (?, ?, ?, ?, ?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, date)
        ibm_db.bind_param(stmt, 2, transaction)
        ibm_db.bind_param(stmt, 3, type)
        ibm_db.bind_param(stmt, 4, amount)
        ibm_db.bind_param(stmt, 5, session["email"])

        ibm_db.execute(stmt)

        data = {
            'Messages': [
                {
                "From": {
                    "Email": "saimohan2022mn@gmail.com",
                    "Name": "PERSONAL EXPENSE TRACKER"
                },
                "To": [
```



```
                    {
                    "Email":session["email"] ,

                    }
                ],
                "Subject": "EXPENSE ADDED",

                "HTMLPart": "DATE:"+date +"  "+"TRANSACTION:"+transaction
+"    "+"TYPE:"+type+"    " +"AMOUNT:"+amount

                }
            ]
        }
        mailjet.send.create(data=data)

        return redirect(url_for('tracker'))


@app.route('/setLimit', methods=['GET', 'POST'])
def limiter():
    if request.method == "POST":
        details = request.form
        print(details)

        limit = details["limit"]
        print(limit)


        sql = "INSERT INTO EXPENSELIMIT VALUES (?, ?)"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, limit)
        ibm_db.bind_param(stmt, 2, session["email"])
        ibm_db.execute(stmt)

        return redirect(url_for('tracker'))



@app.route('/changeLimit', methods=['GET', 'POST'])
def changer():
    if request.method == "POST":
        details = request.form
        print(details)

        limit = details["limit1"]
        print(limit)


        sql = "UPDATE EXPENSELIMIT SET LIMIT=? WHERE email=?"
```



```
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt, 1, limit)
        ibm_db.bind_param(stmt, 2, session["email"])
        ibm_db.execute(stmt)

        return redirect(url_for('tracker'))


@app.route('/logout')
def logout():
    print("Logging Out")
    session.pop('loggedin', None)
    session.pop('id', None)
    session.pop('email', None)
    session.pop('name', None)
    return redirect(url_for('home'))



def send_data():
    data = {
        'Messages': [
            {
            "From": {
                "Email": "saimohan2022mn@gmail.com",
                "Name": "PERSONAL EXPENSE TRACKER"
            },
            "To": [
                {
                "Email":session["email"] ,

                }
            ],
            "Subject": "EXPENSE LIMIT EXCEEDED",

            "HTMLPart":  "<h1>Hi, This is a Notification Mail</h1> </br>
<h2>You have exceeded your MONTHLY EXPENSE LIMIT</h2><br><h3>With regards
Personal Expense Tracker</h3></br> <h4>ibm team</h4></br>"

            }
        ]
    }
    mailjet.send.create(data=data)

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