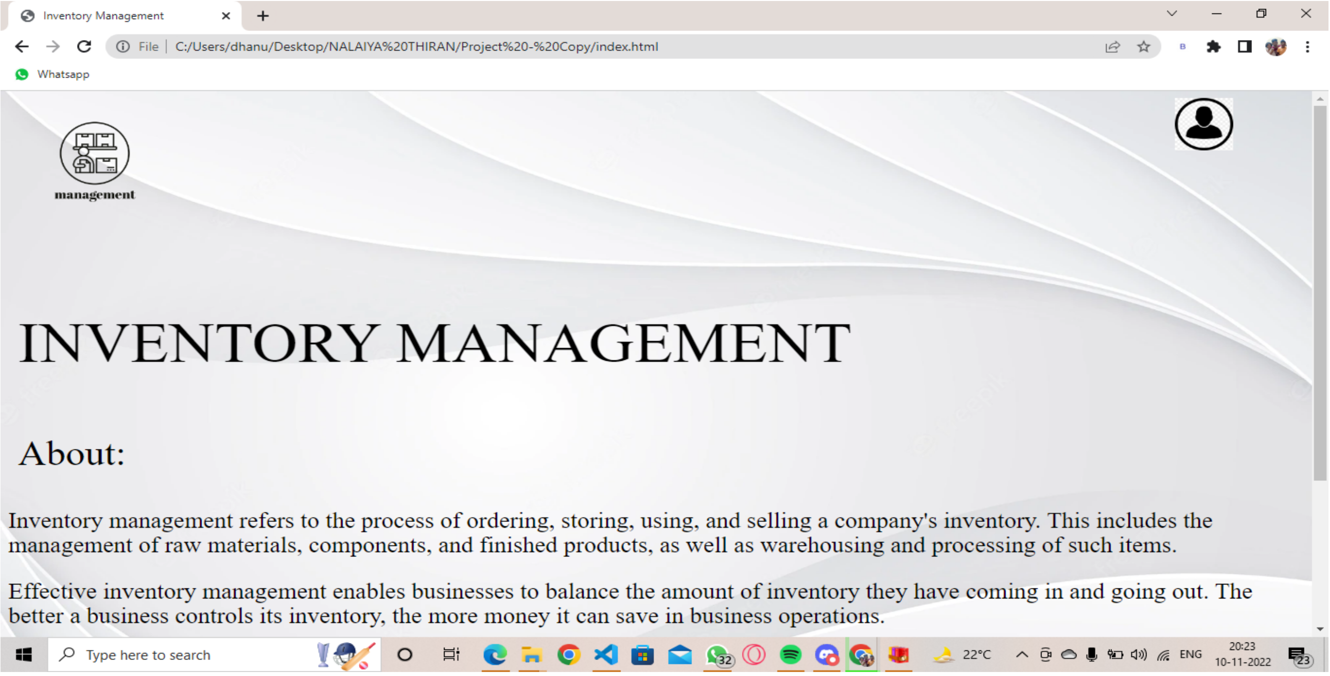
Implementing Web Application

Create UI To Interact With Application

Team ID: PNT2022TMID27088

Project Name Project - **Inventory Management System for Retailers**

Home page and about page:

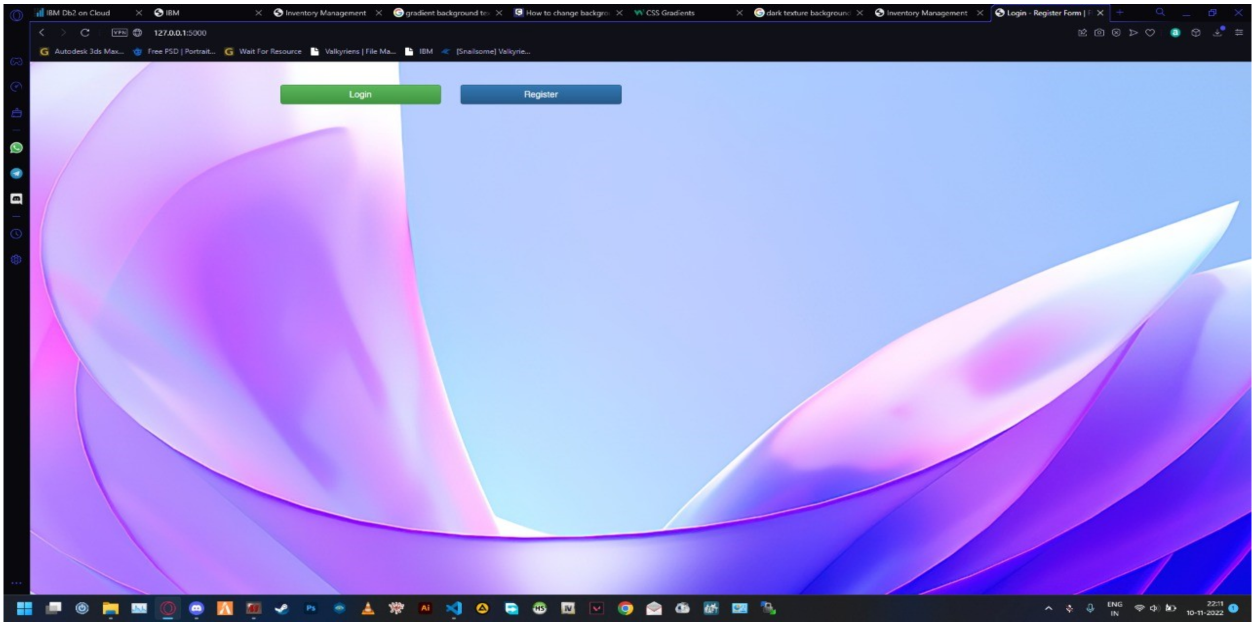


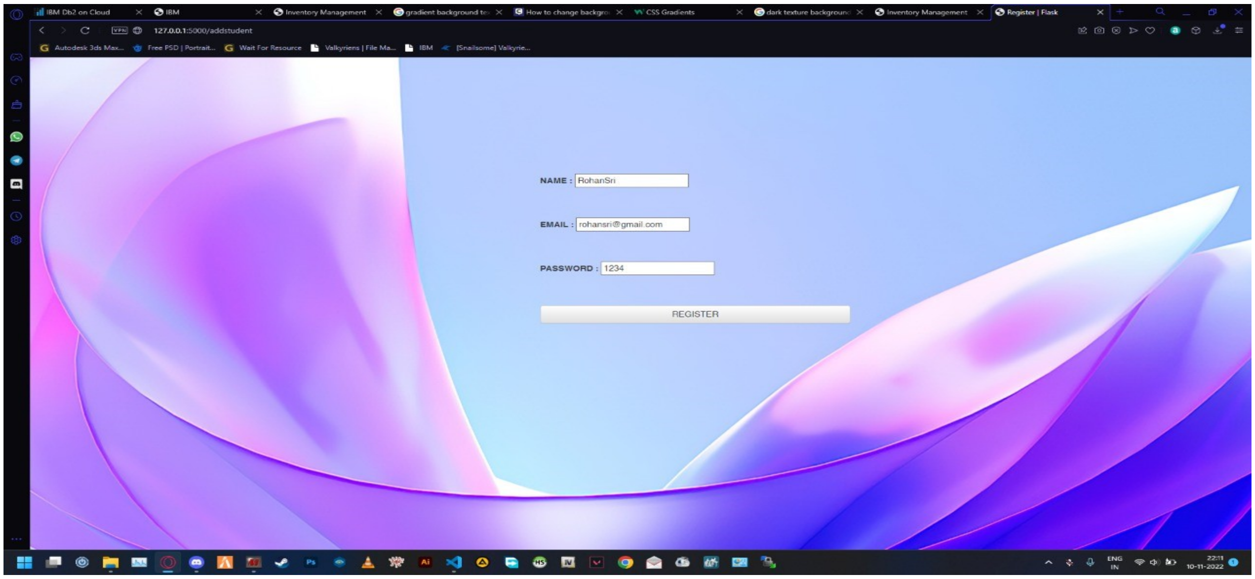
INVENTORY MANAGEMENT

About:

Inventory management refers to the process of ordering, storing, using, and selling a company's inventory. This includes the management of raw materials, components, and finished products, as well as warehousing and processing of such items.

Effective inventory management enables businesses to balance the amount of inventory they have coming in and going out. The better a business controls its inventory, the more money it can save in business operations.

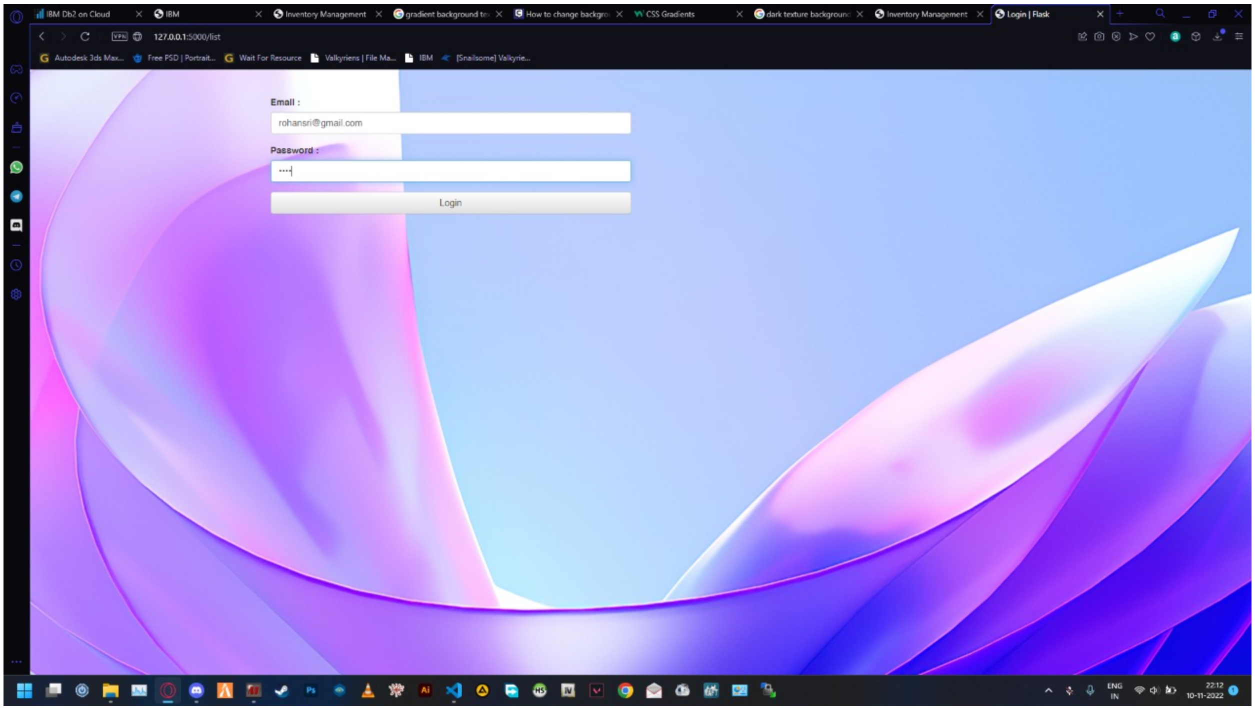




NAME : RohanSri

EMAIL : rohansri@gmail.com

PASSWORD : 1234



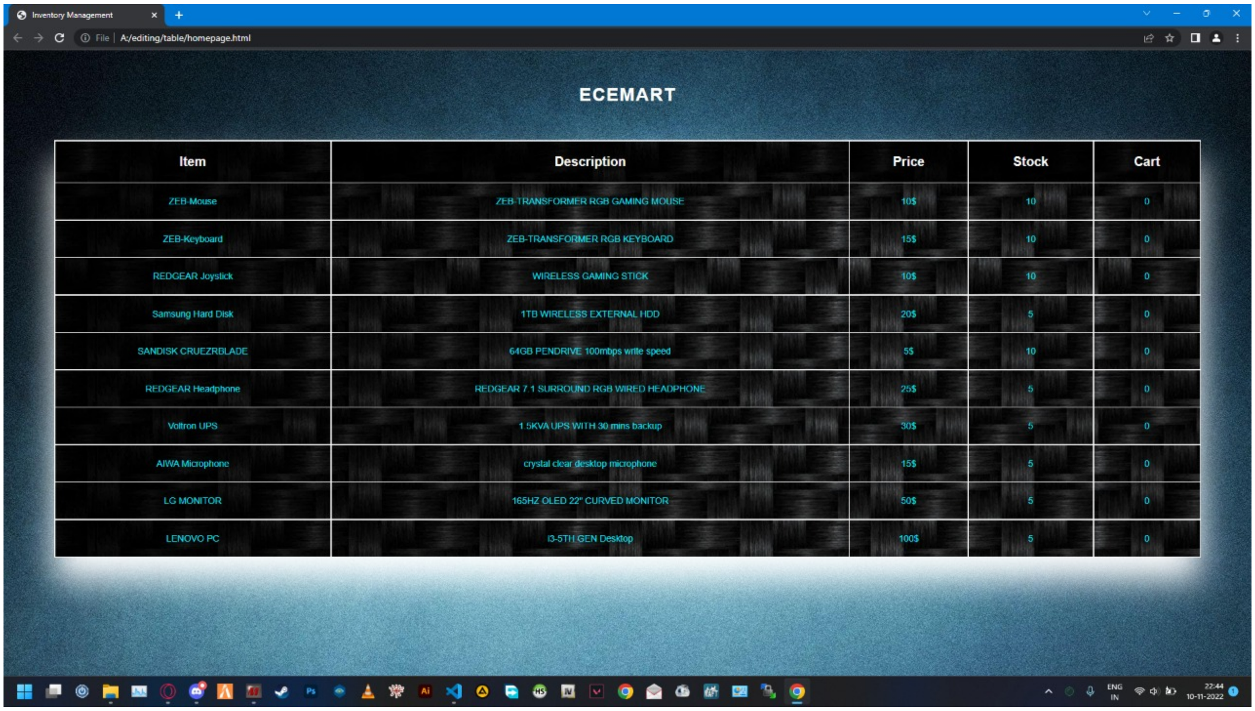
Email :

rohansri@gmail.com

Password :

····

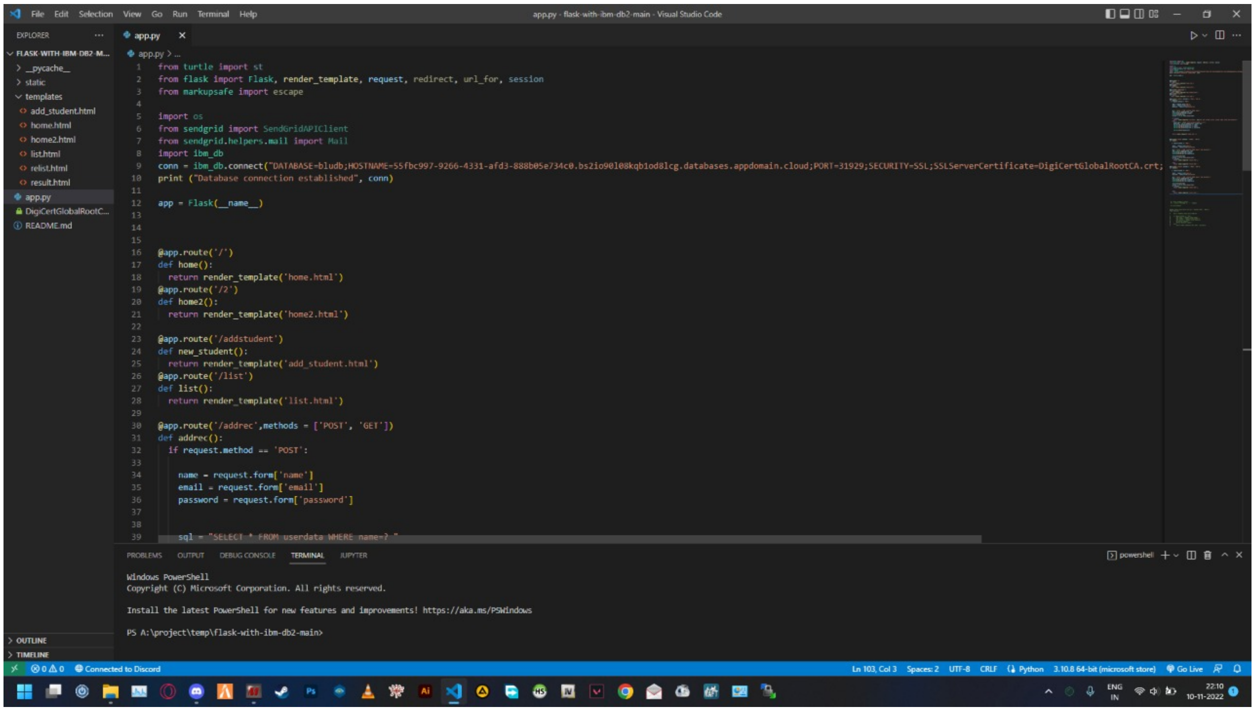
Login



ECEMART

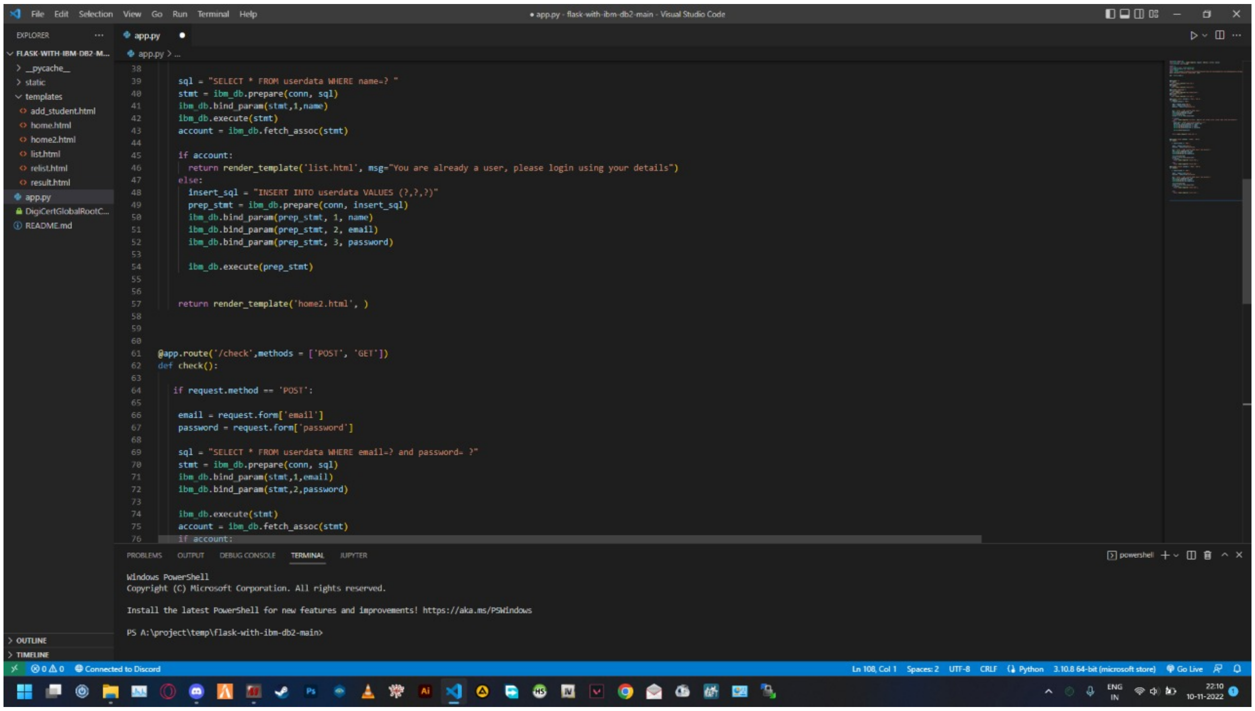
Item	Description	Price	Stock	Cart
ZEB-Mouse	ZEB-TRANSFORMER RGB GAMING MOUSE	10$	10	0
ZEB-Keyboard	ZEB-TRANSFORMER RGB KEYBOARD	15$	10	0
REDGEAR Joystick	WIRELESS GAMING STICK	10$	10	0
Samsung Hard Disk	1TB WIRELESS EXTERNAL HDD	20$	5	0
SANDISK CRUEZRBLADE	64GB PENDRIVE 100mbps write speed	5$	10	0
REDGEAR Headphone	REDGEAR 7.1 SURROUND RGB WIRED HEADPHONE	25$	5	0
Voltron UPS	1.5KVA UPS WITH 30 mins backup	30$	5	0
AIWA Microphone	crystal clear desktop microphone	15$	5	0
LG MONITOR	165HZ OLED 22" CURVED MONITOR	50$	5	0
LENOVO PC	i3-5TH GEN Desktop	100$	5	0

code:



```python
from turtle import st
from flask import Flask, render_template, request, redirect, url_for, session
from markupsafe import escape

import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31929;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;
print ("Database connection established", conn)

app = Flask(__name__)



@app.route('/')
def home():
  return render_template('home.html')
@app.route('/2')
def home2():
  return render_template('home2.html')

@app.route('/addstudent')
def new_student():
  return render_template('add_student.html')
@app.route('/list')
def list():
  return render_template('list.html')

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

    name = request.form['name']
    email = request.form['email']
    password = request.form['password']


    sql = "SELECT * FROM userdata WHERE name=? "
```

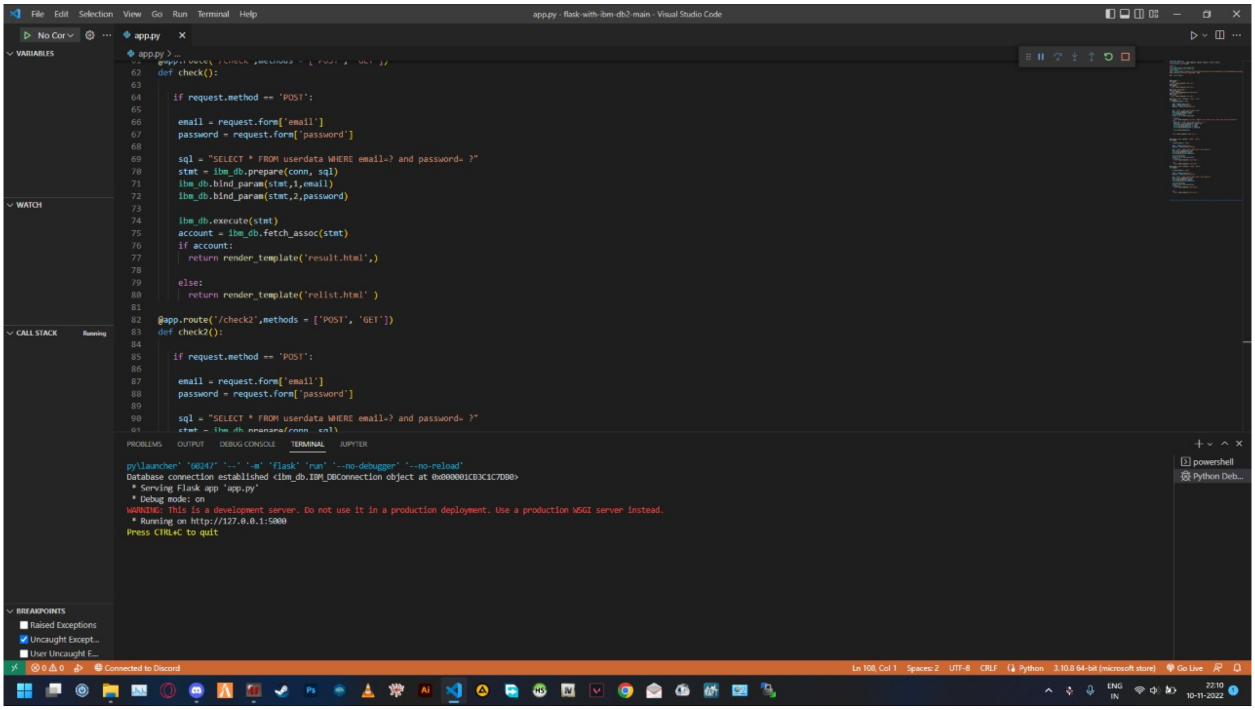
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS A:\project\temp\flask-with-ibm-db2-main>
```



File Edit Selection View Go Run Terminal Help
● app.py - flask-with-ibm-db2-main - Visual Studio Code
EXPLORER
FLASK-WITH-IBM-DB2-M...
__pycache__
static
templates
add_student.html
home.html
home2.html
list.html
relist.html
result.html
app.py
DigiCertGlobalRootC...
README.md
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
powershell
Connected to Discord
Ln 108, Col 1 Spaces: 2 UTF-8 CRLF Python 3.10.8 64-bit (microsoft store) Go Live
ENG IN
22:10
10-11-2022

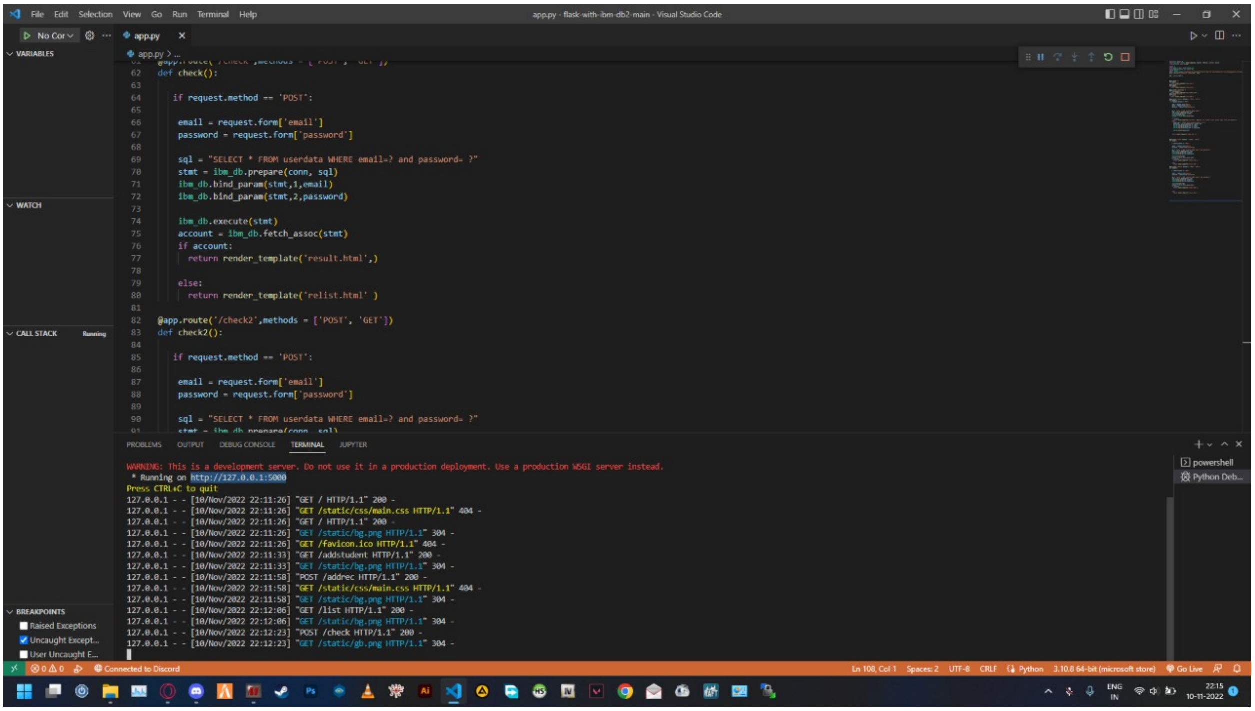
```python

    sql = "SELECT * FROM userdata WHERE name=? "
    stmt = ibm_db.prepare(conn, sql)
    ibm_db.bind_param(stmt,1,name)
    ibm_db.execute(stmt)
    account = ibm_db.fetch_assoc(stmt)

    if account:
      return render_template('list.html', msg="You are already a user, please login using your details")
    else:
      insert_sql = "INSERT INTO userdata 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, password)

      ibm_db.execute(prep_stmt)


    return render_template('home2.html', )



@app.route('/check',methods = ['POST', 'GET'])
def check():

   if request.method == 'POST':

    email = request.form['email']
    password = request.form['password']

    sql = "SELECT * FROM userdata 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)
    if account:
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS A:\project\temp\flask-with-ibm-db2-main>
```


File Edit Selection View Go Run Terminal Help
app.py - flask-with-ibm-db2-main - Visual Studio Code
No Cor
app.py
VARIABLES
WATCH
CALL STACK
Running
BREAKPOINTS
Raised Exceptions
Uncaught Except...
User Uncaught E...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
powershell
Python Deb...
Connected to Discord
Ln 108, Col 1 Spaces: 2 UTF-8 CRLF Python 3.10.8 64-bit (microsoft store) Go Live
ENG IN
22:10
10-11-2022

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

    if request.method == 'POST':

     email = request.form['email']
     password = request.form['password']

     sql = "SELECT * FROM userdata 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)
     if account:
       return render_template('result.html',)

     else:
       return render_template('relist.html' )

@app.route('/check2',methods = ['POST', 'GET'])
def check2():

    if request.method == 'POST':

     email = request.form['email']
     password = request.form['password']

     sql = "SELECT * FROM userdata WHERE email=? and password= ?"
     stmt = ibm_db.prepare(conn, sql)
```

```
py\launcher' '60247' '--' '-m' 'flask' 'run' '--no-debugger' '--no-reload'
Database connection established <ibm_db.IBM_DBConnection object at 0x000001CB3C1C7DB0>
 * Serving Flask app 'app.py'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```


File Edit Selection View Go Run Terminal Help
app.py - flask-with-ibm-db2-main - Visual Studio Code
No Cor
app.py
VARIABLES
WATCH
CALL STACK
Running
BREAKPOINTS
Raised Exceptions
Uncaught Except...
User Uncaught E...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
powershell
Python Deb...
Connected to Discord
Ln 108, Col 1 Spaces: 2 UTF-8 CRLF Python 3.10.8 64-bit (microsoft store) Go Live
ENG IN
22:15
10-11-2022

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

    if request.method == 'POST':

     email = request.form['email']
     password = request.form['password']

     sql = "SELECT * FROM userdata 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)
     if account:
       return render_template('result.html',)

     else:
       return render_template('relist.html' )

@app.route('/check2',methods = ['POST', 'GET'])
def check2():

    if request.method == 'POST':

     email = request.form['email']
     password = request.form['password']

     sql = "SELECT * FROM userdata WHERE email=? and password= ?"
     stmt = ibm_db.prepare(conn, sql)
```

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [10/Nov/2022 22:11:26] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2022 22:11:26] "GET /static/css/main.css HTTP/1.1" 404 -
127.0.0.1 - - [10/Nov/2022 22:11:26] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2022 22:11:26] "GET /static/bg.png HTTP/1.1" 304 -
127.0.0.1 - - [10/Nov/2022 22:11:26] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [10/Nov/2022 22:11:33] "GET /addstudent HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2022 22:11:33] "GET /static/bg.png HTTP/1.1" 304 -
127.0.0.1 - - [10/Nov/2022 22:11:58] "POST /addrec HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2022 22:11:58] "GET /static/css/main.css HTTP/1.1" 404 -
127.0.0.1 - - [10/Nov/2022 22:11:58] "GET /static/bg.png HTTP/1.1" 304 -
127.0.0.1 - - [10/Nov/2022 22:12:06] "GET /list HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2022 22:12:06] "GET /static/bg.png HTTP/1.1" 304 -
127.0.0.1 - - [10/Nov/2022 22:12:23] "POST /check HTTP/1.1" 200 -
127.0.0.1 - - [10/Nov/2022 22:12:23] "GET /static/gb.png HTTP/1.1" 304 -
```