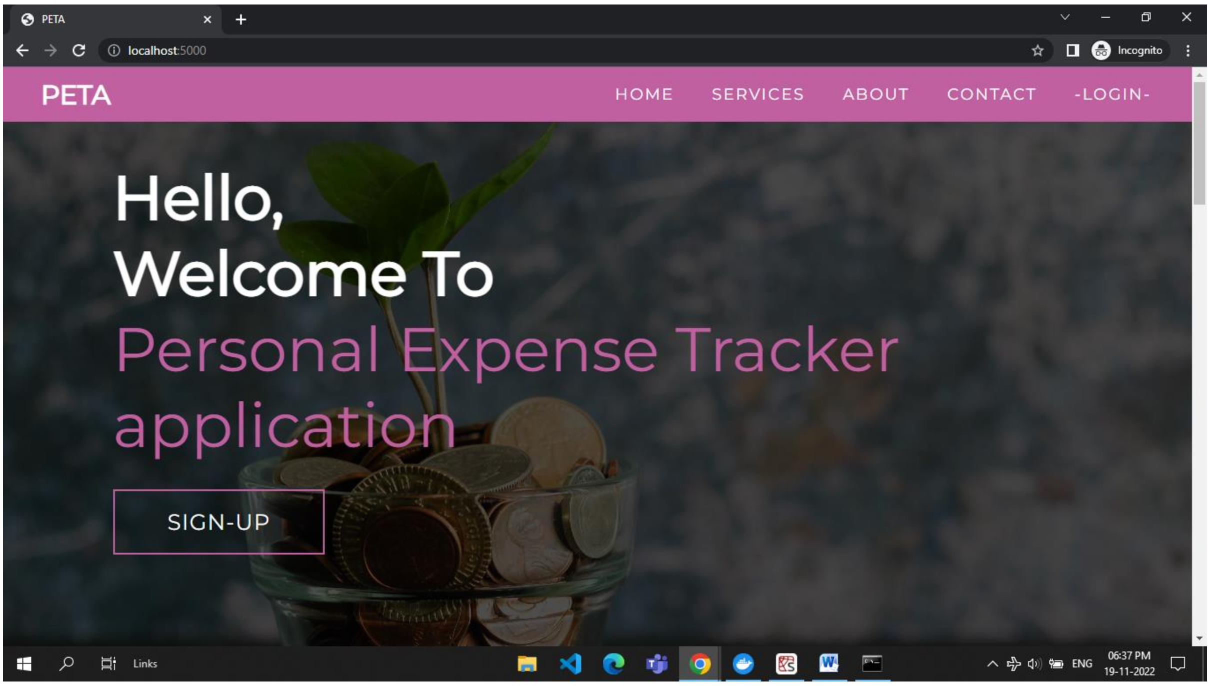
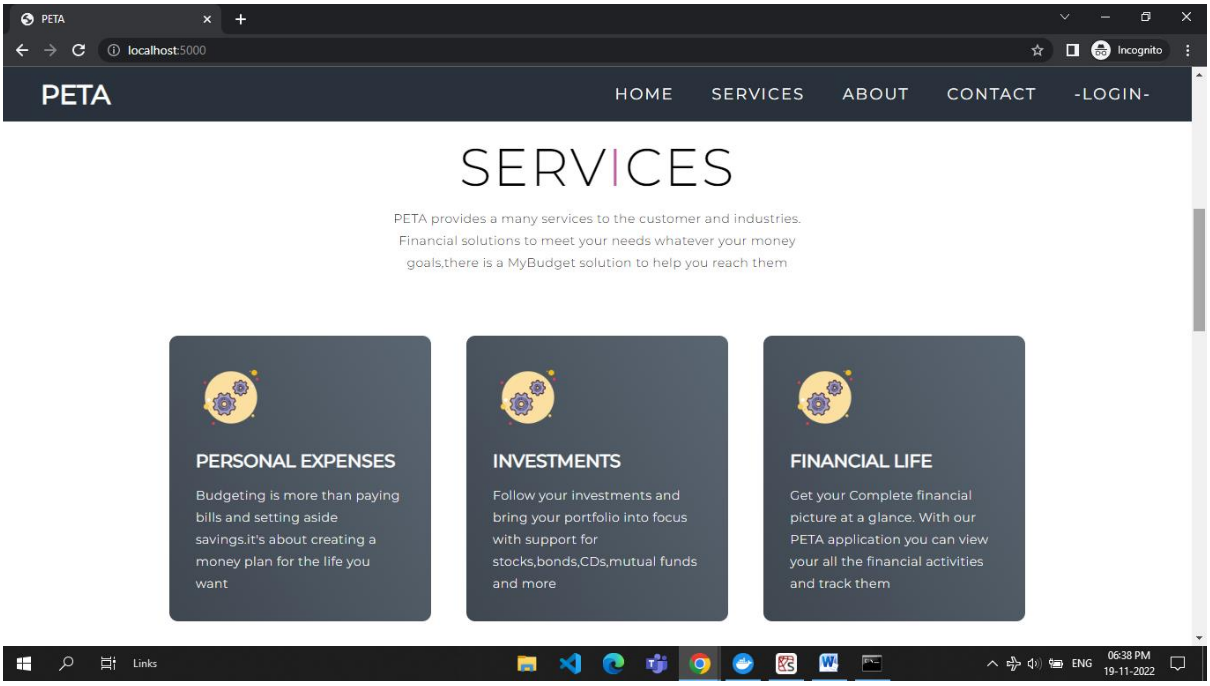
SPRINT 1

Team ID	PNT2022TMID35573
Project Name	Personal Expense Tracker Application

Home page:



Hello,
Welcome To
Personal Expense Tracker application



SERVICES

PETA provides a many services to the customer and industries. Financial solutions to meet your needs whatever your money goals,there is a MyBudget solution to help you reach them

PERSONAL EXPENSES

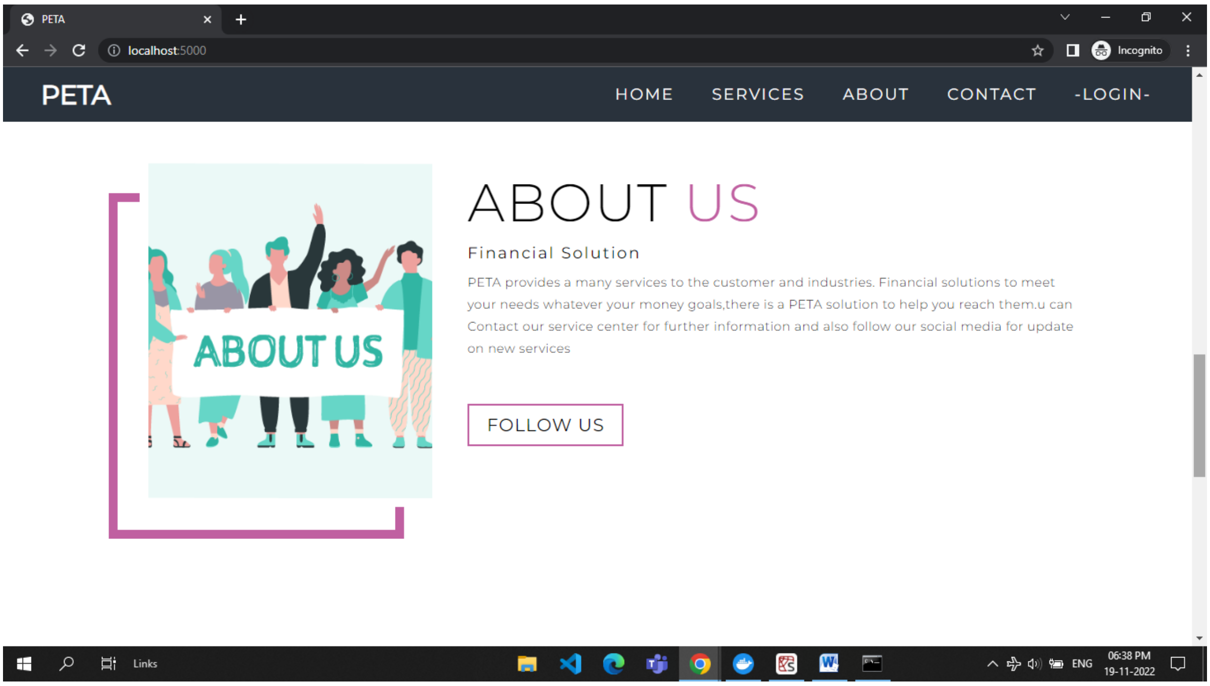
Budgeting is more than paying bills and setting aside savings.it's about creating a money plan for the life you want

INVESTMENTS

Follow your investments and bring your portfolio into focus with support for stocks,bonds,CDs,mutual funds and more

FINANCIAL LIFE

Get your Complete financial picture at a glance. With our PETA application you can view your all the financial activities and track them



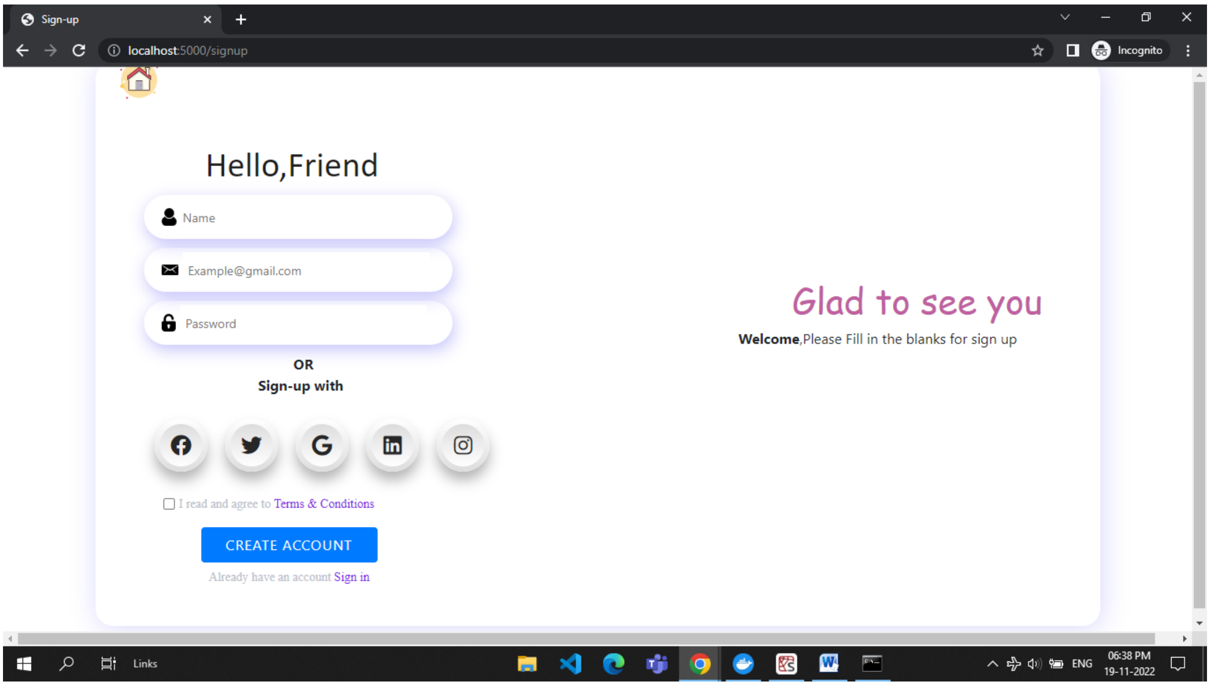
ABOUT US

Financial Solution

PETA provides a many services to the customer and industries. Financial solutions to meet your needs whatever your money goals,there is a PETA solution to help you reach them.u can Contact our service center for further information and also follow our social media for update on new services

FOLLOW US

Register page:



Hello,Friend

Name

Example@gmail.com

Password

OR

Sign-up with

□ I read and agree to Terms & Conditions

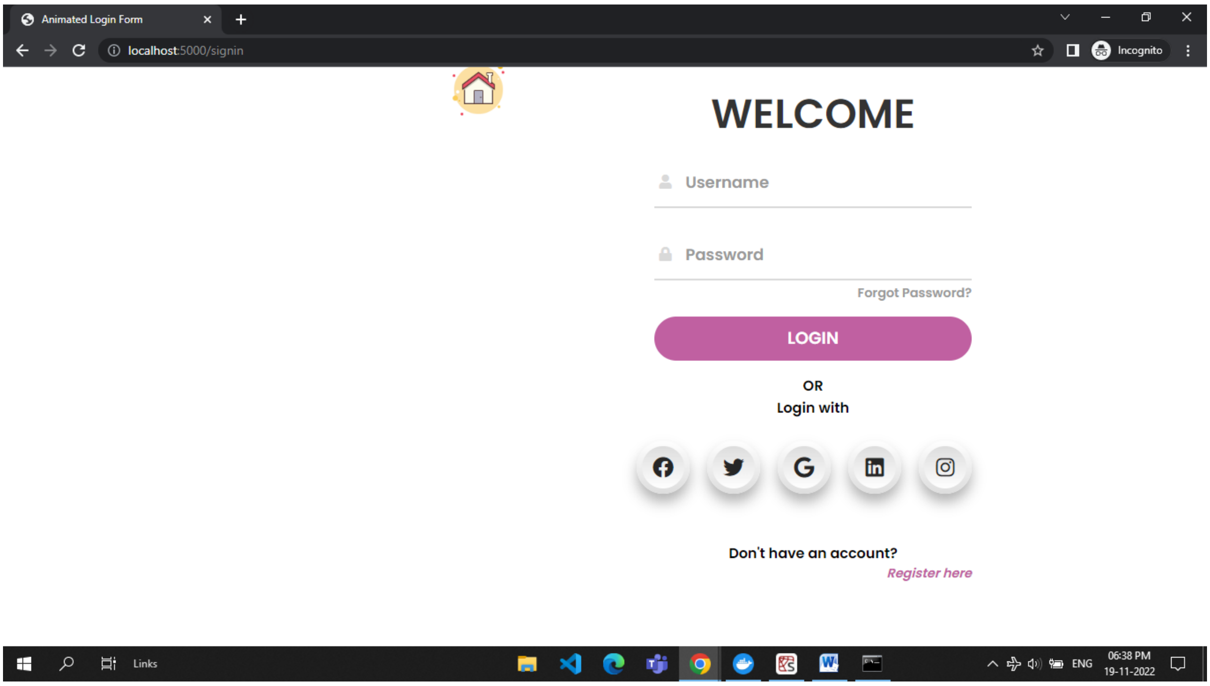
CREATE ACCOUNT

Already have an account Sign in

Glad to see you

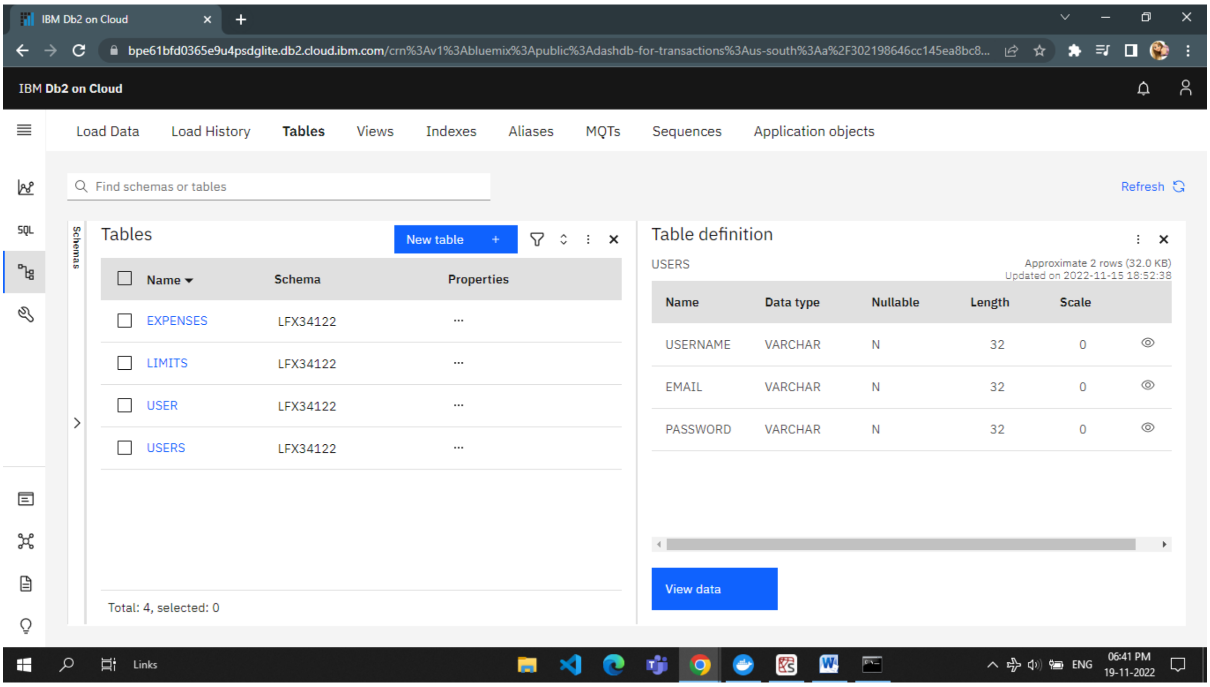
Welcome,Please Fill in the blanks for sign up

Login page:



Don't have an account?

IBM DB:



Tables

□	Name	Schema	Properties
□	EXPENSES	LFX34122	...
□	LIMITS	LFX34122	...
□	USER	LFX34122	...
□	USERS	LFX34122	...

Total: 4, selected: 0

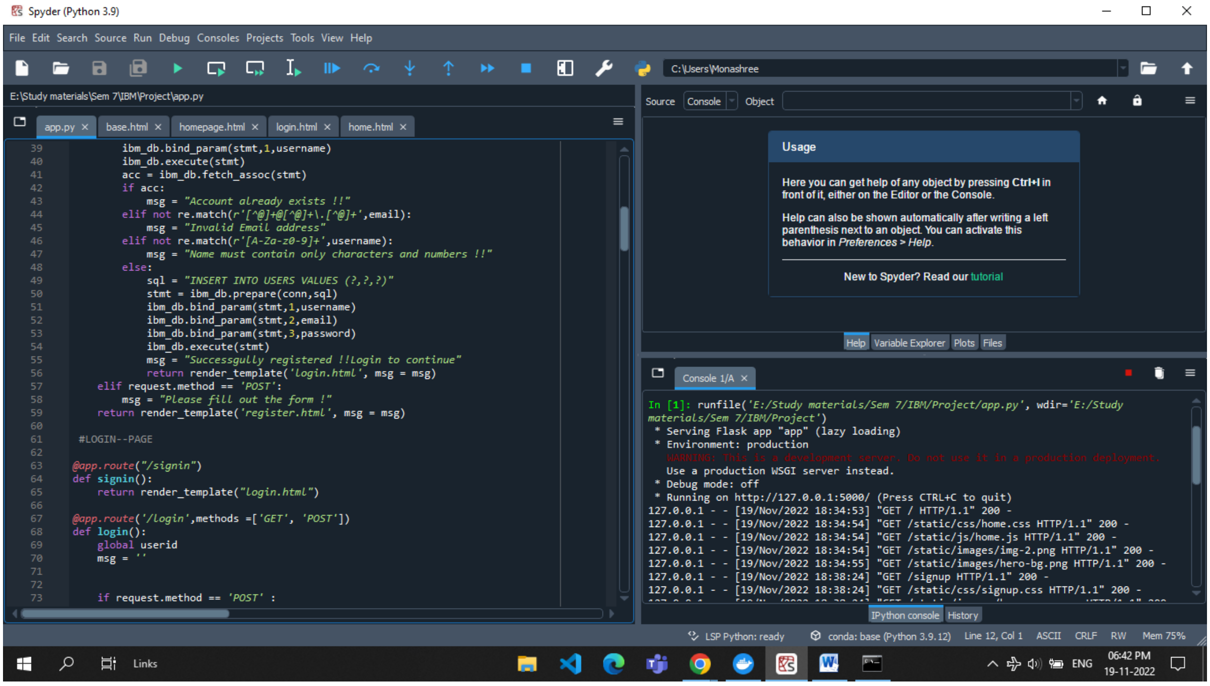
Table definition

USERS

Approximate 2 rows (32.0 KB)
Updated on 2022-11-15 18:52:38

Name	Data type	Nullable	Length	Scale
USERNAME	VARCHAR	N	32	0
EMAIL	VARCHAR	N	32	0
PASSWORD	VARCHAR	N	32	0

Code:



```python
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_assoc(stmt)
        if acc:
            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:
            sql = "INSERT INTO USERS VALUES (?,?,?)"
            stmt = ibm_db.prepare(conn,sql)
            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)
            msg = "Successgully registered !!Login to continue"
            return render_template('login.html', msg = msg)
    elif request.method == 'POST':
        msg = "Please fill out 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 userid
    msg = ''


    if request.method == 'POST' :
```

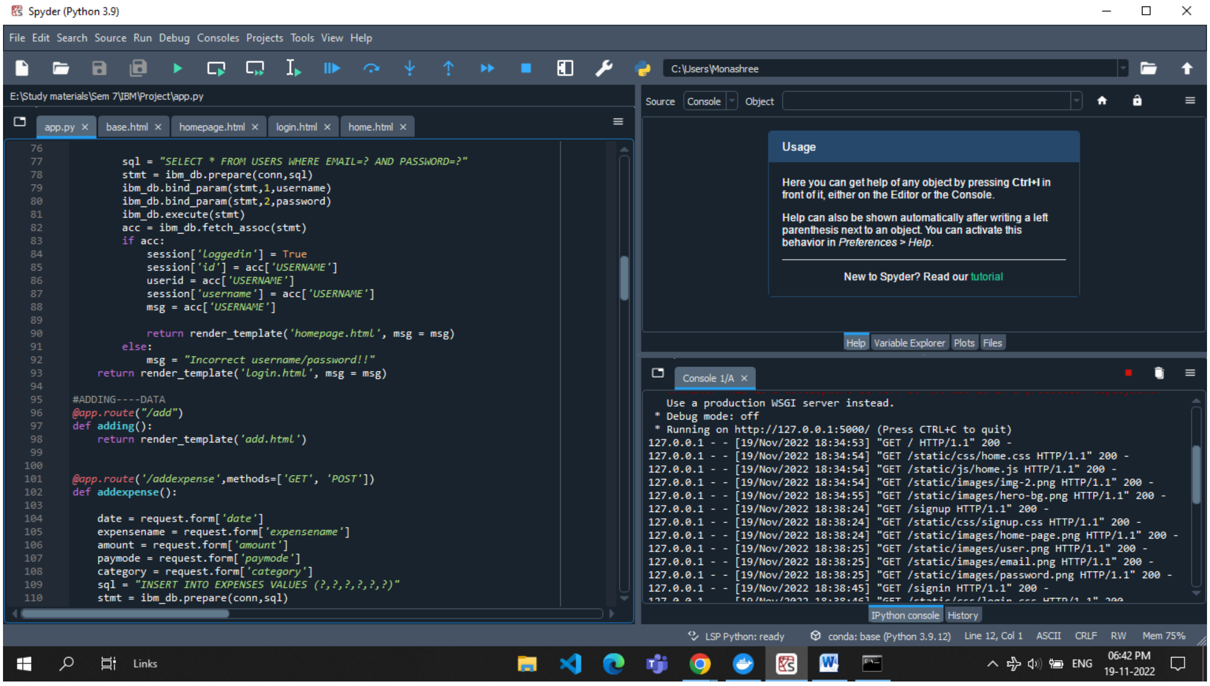
Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
In [1]: runfile('E:/Study materials/Sem 7/IBM/Project/app.py', wdir='E:/Study
materials/Sem 7/IBM/Project')
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [19/Nov/2022 18:34:53] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:34:54] "GET /static/css/home.css HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:34:54] "GET /static/js/home.js HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:34:54] "GET /static/images/img-2.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:34:55] "GET /static/images/hero-bg.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:24] "GET /signup HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:24] "GET /static/css/signup.css HTTP/1.1" 200 -
```



```python

        sql = "SELECT * FROM USERS WHERE EMAIL=? AND PASSWORD=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        acc = ibm_db.fetch_assoc(stmt)
        if acc:
            session['loggedin'] = True
            session['id'] = acc['USERNAME']
            userid = acc['USERNAME']
            session['username'] = acc['USERNAME']
            msg = acc['USERNAME']

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

#ADDING----DATA
@app.route("/add")
def adding():
    return render_template('add.html')


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

    date = request.form['date']
    expensename = request.form['expensename']
    amount = request.form['amount']
    paymode = request.form['paymode']
    category = request.form['category']
    sql = "INSERT INTO EXPENSES VALUES (?,?,?,?,?,?)"
    stmt = ibm_db.prepare(conn,sql)
```

Usage

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [19/Nov/2022 18:34:53] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:34:54] "GET /static/css/home.css HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:34:54] "GET /static/js/home.js HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:34:54] "GET /static/images/img-2.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:34:55] "GET /static/images/hero-bg.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:24] "GET /signup HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:24] "GET /static/css/signup.css HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:24] "GET /static/images/home-page.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:25] "GET /static/images/user.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:25] "GET /static/images/email.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:25] "GET /static/images/password.png HTTP/1.1" 200 -
127.0.0.1 - - [19/Nov/2022 18:38:45] "GET /signin HTTP/1.1" 200 -
```