```
//main.py

from flask import Flask, render_template, flash,request, url_for,
redirect
import sqlite3
from flask import g
from flask import abort

DATABASE = 'database.db'
app = Flask(__name__)

def get_db():
    db = getattr(g, '_database', None)
    if db is None:
        db = g._database = sqlite3.connect(DATABASE)
    return db

@app.teardown_appcontext
def close_connection(exception):
    db = getattr(g, '_database', None)
    if db is not None:
        db.close()


@app.route('/')
def index():
    cur = get_db().cursor()
    print (cur)
    return 'Hello'

@app.route('/Users', methods=['POST'])
def Users():
    username = request.form['username']
    password = request.form['password']
    email = request.form['email']
    db = get_db()
```

```
    err = db.execute(
    'insert into User (username, password, email) values
(?,?,?)',
    (
        request.form.get('username', type=str),
        request.form.get('password', type=str),
        request.form.get('email', type=str)
    )
    )
    db.commit()
    print(err)

    return email.upper()

@app.route('/login', methods=['POST'])
def Login():
    db = get_db()
    username = request.form['username']
    password = request.form['password']
    cur = db.execute(
        'SELECT * FROM User WHERE username=?', (username,)
    )
    rows = cur.fetchall()
    print (rows)
    if len(rows) <= 0:
        abort(404)
    else:
        if (rows[0][2] != password):
            abort(403)
        else:
            return 'Login is successful'

def init_db():
    db = get_db()
    print ("Initializaing database")
```

```
    with app.open_resource('schema.sql') as f:
        db.executescript(f.read().decode('utf8'))



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

*** To run main.py use following command:**

```
$ FLASK_ENV=development FLASK_APP = main.py flask run
```

Pybot > schema.sql

```
DROP TABLE IF EXISTS User;
DROP TABLE IF EXISTS post;

CREATE TABLE User (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  username TEXT UNIQUE NOT NULL,
  password TEXT NOT NULL,
  firstname TEXT NOT NULL,
  lastname TEXT NOT NULL
  email TEXT NOT NULL
);

CREATE TABLE Vendor (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  name TEXT NOT NULL,
  address TEXT NOT NULL,
  phone_no TEXT NOT NULL,
);
Insert into Vendor
values(110,'Chipotle','Redwood City','713-792-9390');

Insert into Vendor
values('McDonald','Mountain View','713-668-5882');

Insert into Vendor
values('Panda Express','Mountain View','713-668-7898');

Insert into Vendor
values('Burger King','San Jose','713-989-1111');
```

To connect database from terminal:

```
(base) Vinits-MacBook-Pro:Pybot namrata$ sqlite3
Users.php    __pycache__/ database.db  main.py      schema.sql
(base) Vinits-MacBook-Pro:Pybot namrata$ sqlite3 database.db
SQLite version 3.33.0 2020-08-14 13:23:32
Enter ".help" for usage hints.
sqlite> select * from user;


(base) Vinits-MacBook-Pro:Pybot namrata$ sqlite3 database.db < schema.sql
(base) Vinits-MacBook-Pro:Pybot namrata$ sqlite3 database.db
SQLite version 3.33.0 2020-08-14 13:23:32
Enter ".help" for usage hints.
sqlite> select * from user;
sqlite> select * from vendor;
110|Chipotle|Redwood City|713-792-9390
111|McDonald|Mountain View|713-668-5882
222|Panda Express|Mountain View|713-668-7898
333|Burger King|San Jose|713-989-1111
sqlite>
```

```
sqlite> select * from vendor;
110|Chipotle|Redwood City|713-792-9390
111|McDonald|Mountain View|713-668-5882
222|Panda Express|Mountain View|713-668-7898
333|Burger King|San Jose|713-989-1111
```

- You can test the User Signup using following command
ex:
$ curl -X POST -F 'username=namrata' -F 'password=namrata' -F 'emailata@way.com' -F 'firstnamstname=xyz' localhost:5000/Users -v

- You can test the Login using following command

$ curl -X POST -F 'username=namrata' -F 'password=namrata' localhost:5000/login -v