# Creating a web app using python/flask/mariadb

In this tutorial, we'll be using to create a simple web app using html, python flask, MariaDb.

We'll use simple example for understand how to build web app.

1. Introduction to flask:
Flask is a python framework for creating web application.

Setup flask in Ubuntu 14.04

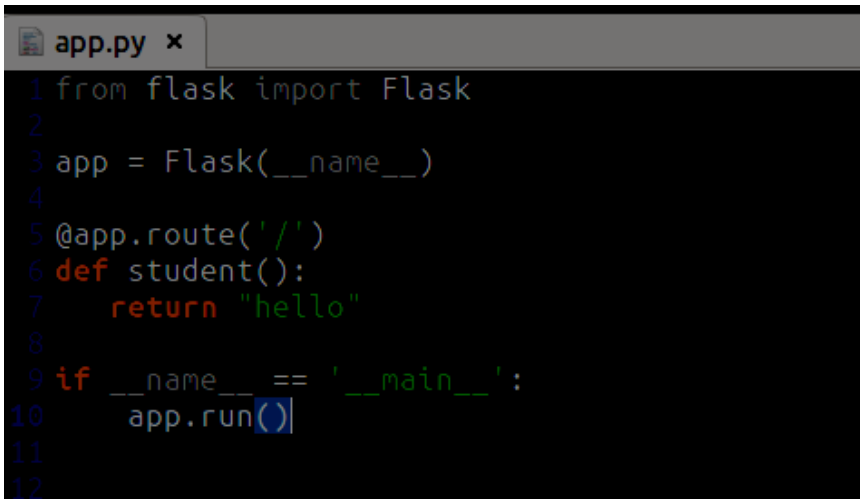
1. Update
sudo apt-get update

2.Install python-pip

sudo apt-get install python-pip

3. Install Flask
pip install Flask

Installation in complete now verify that flask is working or not.




And run the application python app.py

Go to your browser and enter http://localhost:5000/

You can also write app.run (port=5000,debug = True)
And you can give whatever port you want.

-> In First line we are importing Flask

-> app = Flask(__name__) creates an app object from Flask.

-> app.run() starts the web server and ready to handle request

define request

```
@app.route('/')
def student():
    return "hello"
```

Create web application

1. create folder which should have two folder
1. templates (put all .html files inside it)
2. statics (put .css file and image,video,logo )

and .py file

now start with simple example:

HTMl

student.html

```
<html>
   <body>

      <form action = "/act" method = "POST">
         <p>Name <input type = "text" name = "name" /></p>
         <p>Age <input type= "text" name = "age" /></p>
         <p><input type = "submit" value = "submit" /></p>
      </form>


   </body>
</html>
```

now make the app.py file

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
```

```
def student():
   return render_template('student.html')

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

now run this than you will get


when we will fill the name and age and then submit after submission we have to store data somewhere for that we create database in mariadb

first we will install mariadb
for installation you can use this link
http://www.liquidweb.com/kb/how-to-install-mariadb-5-5-on-ubuntu-14-04-lts/

step 1. add the mariadb repository

The software-properties-common package should already be installed, but just in case:

sudo apt-get install software-properties-common

We'll import the MariaDB public key used by the package management system:

sudo apt-key adv --recv-keys --keyserver hkp://keyserver.ubuntu.com:80 0xcbcb082a1bb943db

Then we'll add the MariaDB repository:

sudo add-apt-repository 'deb http://mirror.jmu.edu/pub/mariadb/repo/5.5/ubuntu trusty main'

Now reload the package database:

sudo apt-get update

step 2. install mariadb
sudo apt-get install mariadb-server

You may receive the following prompt or something similar:

After this operation, 116 MB of additional disk space will be used.

Do you want to continue? [Y/n]

Enter Y to continue.

Next you'll be asked:

New password for the MariaDB "root" user:

This is an administrative account in MariaDB with elevated privileges; enter a strong password.

Then you'll be asked to verify the root MariaDB password:

Repeat password for the MariaDB "root" user:

That's it! Your basic MariaDB installation is now complete!

Be sure to stop MariaDB before proceeding to the next step:

sudo service mysql stop

Step 3: Configure and Secure MariaDB for Use
Now we'll instruct MariaDB to create its database directory structure:

sudo apt-get install software-properties-common

We'll import the MariaDB public key used by the package management system:

sudo mysql_install_db

Start MariaDB:

sudo service mysql start

And now let's secure MariaDB by removing the test databases and anonymous user created by default:

sudo mysql_secure_installation

You'll be prompted to enter your current password. Enter the root MariaDB password set during installation:

Enter current password for root (enter for none):

Then, assuming you set a strong root password, go ahead and enter n at the following prompt:

Change the root password? [Y/n] n

Remove anonymous users, Y:

Remove anonymous users? [Y/n] Y

Disallow root logins remotely, Y:

Disallow root login remotely? [Y/n] Y

Remove test database and access to it, Y:

Remove test database and access to it? [Y/n] Y

And reload privilege tables, Y:

Reload privilege tables now? [Y/n] Y

now you can use mariadb

To install MySQL on Ubuntu, use the following command:

```
sudo apt-get install mysql-server
```

Creating a Database in MySQL and MariaDB

write in terminal

```
mysql -u username -p
CREATE DATABASE testdb;
USE testdb;
```

How to Create a Table in MySQL and MariaDB

```
CREATE TABLE test(

     name  varchar(20) not null, age varchar(3) not null

    );
```

now database is created

testdb = database name

test = table name

using this command you see table in terminal

```
show columns in test;
```

now make the mariadb connection

```
import mysql.connector as mariadb

conn = mariadb.connect(user='root',password='gaytri713@', database='testdb')

cur = conn.cursor()
```

```
from flask import Flask, render_template
import mysql.connector as mariadb

app = Flask(__name__)

@app.route('/')
def student():
   return render_template('student.html')
@app.route('/act', methods = ['GET','POST'])
def act():
    if(request.method == 'POST'):
        try:

            conn = mariadb.connect(user='root',password='gaytri713@', database='testdb')
            cur = conn.cursor()
            return "connection successful"
        except:
            return "Data Base Connection Error!"
```

now your mariadb connection is successful

now modify your upper code

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

def act():

      if(request.method == 'POST'):

            try:

                  name = request.form['name']

                  age = request.form['age']

                  conn = mariadb.connect(user='root',password='gaytri713@',
database='testdb')

                  cur = conn.cursor()
```

```
                sql = "INSERT INTO
test(name,age)values('{}','{}')".format(name,age);

                cur.execute(sql)

                conn.commit()
```

after modify run this code when you will fill name and age and press submit then all data will saved in database

we can show this database on web also for that we have to make one more html webpage list.html

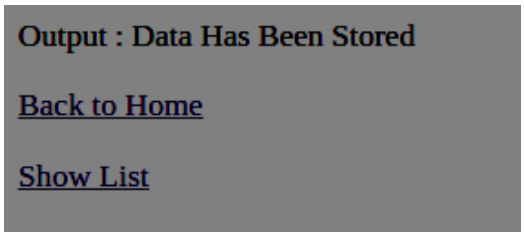
```
<html>
<body>

    <table border=1>
        <thead>
            <th>Name</th>
            <th>Age</th>
        </thead>

        {% for row in rows %}
            <tr>
                <td>{{row[0]}}</td>
                <td>{{row[1]}}</td>
            </tr>
        {% endfor %}

    </table>
    <p><a href="{{ url_for('student')}}">Back To Home Page</a></p>
</body>
</html>
```

now add this code to show list of database

```
@app.route('/list')

def list():

            conn = mariadb.connect(user='root',password='gaytri713@',
database='testdb')

            cur = conn.cursor()
```

```
        cur.execute("select *from test")

        rows= cur.fetchall()

        return render_template("list.html",rows=rows)
```

now you can see the all fill data




i am giving to menu here back to home and show list and print massage, for printing this two menu you can modify your code

```
            msg = "Data Has Been Stored"
            return render_template('output.html',msg=msg)
        except:
            return "Data Base Connection Error!"
```

you need to make outpur.html file

```
<html>
<body>
    <p>Output : {{ msg }}</p>
    <p><a href ="{{ url_for('student') }}">Back to Home</a></p>
    <p><a href="{{ url_for('list') }}">Show List</a></p>
</body>

<html>
```

now your web app is ready

final code in python flask is here

```python
from flask import Flask, render_template, request
import mysql.connector as mariadb

app = Flask(__name__)

@app.route('/')
def student():
   return render_template('student.html')
@app.route('/act', methods = ['GET','POST'])
def act():
    if(request.method == 'POST'):
        try:
            name = request.form['name']
            age = request.form['age']
            conn = mariadb.connect(user='root',password='gaytri713@', database='testdb')
            cur = conn.cursor()
            sql = "INSERT INTO test(name,age)values('{}','{}')".format(name,age);
            cur.execute(sql)
            conn.commit()
            msg = "Data Has Been Stored"
            return render_template('output.html',msg=msg)
        except:
            return "Data Base Connection Error!"
@app.route('/list')
def list():
        conn = mariadb.connect(user='root',password='gaytri713@', database='testdb')
        cur = conn.cursor()
        cur.execute("select *from test")
        rows= cur.fetchall()
        return render_template("list.html",rows=rows)
if __name__ == '__main__':
    app.run(port = 5001, debug = True)
```