Calculator using Python-Flask

G V V Sharma*

Contents

***Abstract*—This manual shows how to build a calculator using Python-Flask. The user interface is through a browser while the computations are done in Python.**

1 Python-flask

Flask is Python framework for creating web applications.

1.1 Installation:

```
sudo apt-get update
sudo apt-get install python-pip
sudo pip install flask
```

1.2 Calculator UI in HTML: Download the following code and open it using a browser. You will see the calculator UI.

1.3 Save **calc.html** in a folder called **templates**.

1.4 Type the following code in a file called **calc_ui.py**.

1.5 Make sure that the python file is outside the **templates** directory. Now type

```
python calc_ui.py
```

on the terminal. An address will be displayed on the terminal.

1.6 Enter the above address in a browser. You should see the calculator UI.

*GVV Sharma is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

1.1 Fetching the stored Data from the Database

1) Save the following code in a file called **display.html**.

```
<html>
<body>
  <table border=1>
        <thead>
        <th>Name</th>
        <th>Roll</th>
        </thead>
        {% for row in rows %}
  <tr>
        <td>{{ row[0]}}</td>
        <td>{{ row[1]}}</td>
  </tr>
        {% endfor %}
  </table>
        <p><a href="/">Back To
           Home Page</a></p>
        <p><a href="/update">
           Update</a></p>
</body>
</html>
```

2) Save the following code in a file titled **display.py**.

3)

```
from flask import Flask,
   render_template,request
import mysql.connector as
   mariadb
app=Flask(__name__)
@app.route('/')
def list():
        conn=mariadb.connect(
           user='root',password
           ='123',database='
           Test')
        # Connecting to
           Database
```

```
        cur=conn.cursor()
        cur.execute("Select * from test") #This query is used to fetch the Data from the Database
        rows=cur.fetchall()
        return render_template("display.html",rows=rows)
        # Returning display.html File
if __name__ == '__main__':
        app.run(debug = True)
```

4) Now open the terminal and type

```
python display.py
```

An address will be displayed.

5) Open this address in a browser. You can see all the Name and Roll No entries in the database.

1.2 Updating the Database

1)

2) Save the following code in a file with titled **show.html**.

```
<html>
<body>
  <table border=1>
        <td>Name</td>
        <td>Roll</td>
        <td>update</td>
        {% for row in rows %}
  <tr>
        <form action="/testupdate" method="POST">
        <td><input type ="text" name ="name" value ={{ row[0]}}></td>
        <td><input type ="text" name ="roll" value ={{ row[1]}}></td>
        <td><input type = "submit" value ="update"></td>
        </form>
        </tr>
        {% endfor %}
  </table>
</body>
</html>
```

3) Save the following code in a file titled **update.py**.

4)

```
from flask import Flask, render_template,request
import mysql.connector as mariadb
app=Flask(__name__)
@app.route('/')
def list():
        conn=mariadb.connect(user='root',password='123',database='Test')
        # connecting to the database
        cur=conn.cursor()
        cur.execute("Select * from test")
        # fetching all the data from test table.
        rows=cur.fetchall()
        return render_template("show.html",rows=rows)
        #returning show.html file

@app.route ('/testupdate', methods =['GET','POST'])
def testupdate():
        conn=mariadb.connect(user='root',password='123',database='Test')
        cur=conn.cursor()
        name=request.form['name']
        roll=request.form['roll']
        print(roll)
        print(name)
        cur.execute("UPDATE test set roll='{}' where name='{}'".format(roll,name))
        # Query for updating
```

```
            the data in test
            table.
        conn.commit()
        return render_template(
            'message.html',msg="
            Data updated")
@app.route('/backhome')
def backhome():
        return render_template(
            'student.html')
        # returing to the main
            page after updating
if __name__ == '__main__':
        app.run(debug = True)
```

5) Now open the terminal and run the **update.py** file.
6) Update whatever data you wish to and click the Update button.
7) Run **display.py** to verify that your data is indeed updated.

1.3 Linking all modules to create the Database application

1) Save the following code in a file called **output.html**.

```
<html>
<body>
        <p>output:{{msg}}</p>
        <p><a href="/">Home</a>
            </p>
        <p><a href="/display">
            Show List</a></p>
        <p><a href="/update">
            Update</a></p>
</body>
</html>
```

2) Save the following code in a file titled **app.py**

```
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']
          roll=request.form['
              roll']
          conn=mariadb.connect(
              user='root',
              password='123',
              database='Test')
          cur=conn.cursor()
          sql="INSERT INTO test
              (name,roll)values
              ('{}','{}')".
              format(name,roll)
          cur.execute(sql)
          conn.commit()
          return
              render_template("
              output.html",msg="
              Data Has Been
              Stored")
        except:
          return "Database
              connection error"
@app.route('/display')
def display():
  conn=mariadb.connect(user='
      root',password='123',
      database='Test')
  cur=conn.cursor()
  cur.execute("Select * from
      test")
  rows=cur.fetchall()
  return render_template("
      display.html",rows=rows)
@app.route('/update')
def list():
  conn=mariadb.connect(user='
      root',password='123',
      database='Test')
  cur=conn.cursor()
  cur.execute("Select * from
      test")
  rows=cur.fetchall()
  return render_template("show.
      html",rows=rows)
```

```
@app.route ('/testupdate',
    methods =['GET','POST'])
def testupdate():
   conn=mariadb.connect(user='
      root',password='123',
      database='Test')
   cur=conn.cursor()
   name=request.form['name']
   roll=request.form['roll']
   print(roll)
   print(name)
   cur.execute("UPDATE test set
      roll='{}' where name='{}'"
      .format(roll,name))
   conn.commit()
   return render_template('
      student.html',msg="Data
      updated")
@app.route('/backhome')
def backhome():
   return render_template('
      student.html')

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

3) Run **app.py**
4) Start using your application.
5) Modify your application so that you may delete a record.