User Management System with Python, Flask and MySQL

webdamn.com/user-management-system-with-python-flask-and-mysql

webdamn June 7, 2023

In our previous Python tutorial, we have explained how to develop Weather App in Python using Flask. In this tutorial, we will explain how to develop User Management System with Python, Flask and MySQL.

User section is an important part of any web application in which users are created, updated, deleted, viewed etc. We can easily develop user managment system in Python using Flask and MySQL packages.

Watch Video At: https://youtu.be/UnLlgLS1dQA

So let's proceed to develop User Management System with Python, Flask and MySQL.

Application Setup

We will create application directory `user-management-system-python` using below command.

```
$ mkdir user-management-system-python
```

then moved within project direcotry

```
$ cd user-management-system-python
```

Modules Required

We will use folloing modules in this application from Python.

- **Flask**: It is a micro framework from Python to create web application. So we will install this module to create web applications. We will install it using the below command:

  ```
  pip install Flask
  ```

- **flask_mysqldb**: This is Python package that can be used to connect to MySQL database. We will install it using the below command:

  ```
  pip install flask_mysqldb
  ```

Create MySQL Database and Table

We will create MySQL database `user-system` and create table`user` to store users details.

```
CREATE TABLE `user` (
  `userid` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `password` varchar(255) NOT NULL,
  `role` enum('admin','user') NOT NULL,
  `country` varchar(350) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

ALTER TABLE `user`
  ADD PRIMARY KEY (`userid`);

ALTER TABLE `user`
  MODIFY `userid` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;
```

Create Application File

We will create application file `app.py` into project root directory.

Then we will import installed module `Flask` and `flask_mysqldb` with it’s helpers.

we will create Flask app and assign `app.config` vaues for MySQL database connection to access database.

```
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mysqldb import MySQL
import MySQLdb.cursors
import re

app = Flask(__name__)

app.secret_key = 'xyzsdfg'

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'user-system'

mysql = MySQL(app)

@app.route('/')


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

Implement User Login Section

We will create template file `templates/login.html` and create FORM with input and submit button.

```
<form action="{{ url_for('login') }}" method="post">
        {% if mesage is defined and mesage %}
                <div class="alert alert-warning">{{ mesage }}</div>
        {% endif %}
        <div class="form-group">
                <label for="email">Email:</label>
                <input type="email" class="form-control" id="email" name="email"
placeholder="Enter email" name="email">
        </div>
        <div class="form-group">
                <label for="pwd">Password:</label>
                <input type="password" class="form-control" id="password"
name="password" placeholder="Enter password" name="pswd">
        </div>
        <button type="submit" class="btn btn-primary">Login</button>
</form>
```

Then we will create function `login()` in `app.py` file and call function `render_template()` to render `login.html` file to load user login page. We will implement login functionality by executing SQL query to perform user login.

```
@app.route('/login', methods =['GET', 'POST'])
def login():
    mesage = ''
    if request.method == 'POST' and 'email' in request.form and 'password' in
request.form:
        email = request.form['email']
        password = request.form['password']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM user WHERE email = % s AND password = % s',
(email, password, ))
        user = cursor.fetchone()
        if user:
            if user['role'] == 'admin':
                session['loggedin'] = True
                session['userid'] = user['userid']
                session['name'] = user['name']
                session['email'] = user['email']
                mesage = 'Logged in successfully !'
                return redirect(url_for('users'))
            else:
               mesage = 'Only admin can login'
        else:
            mesage = 'Please enter correct email / password !'
    return render_template('login.html', mesage = mesage)

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

Implement Add New User

We will create template file `templates/register.html` and create user FORM with input and submit button.

```
<form action="{{ url_for('register') }}" method="post">
        {% if mesage is defined and mesage %}
                <div class="alert alert-warning">{{ mesage }}</div>
        {% endif %}
        <div class="form-group">
                <label for="name">Name:</label>
                <input type="text" class="form-control" id="name" name="name"
placeholder="Enter name">
        </div>
        <div class="form-group">
                <label for="email">Email:</label>
                <input type="email" class="form-control" id="email" name="email"
placeholder="Enter email">
        </div>
        <div class="form-group">
                <label for="pwd">Password:</label>
                <input type="password" class="form-control" id="password"
name="password" placeholder="Enter password">
        </div>
        <div class="form-group">
                <label for="role">Role:</label>
                <select class="form-control" id="role" name="role">
                        <option value="admin">Admin</option>
                        <option value="user">User</option>
                </select>
        </div>
        <div class="form-group">
                <label for="country">Country:</label>
                <input type="text" class="form-control" id="country"
name="country">
        </div>
        <button type="submit" class="btn btn-primary">Register</button>
</form>
```

We will implement new user add using form submit post values. We will insert users details into `user` table.

```
@app.route('/register', methods =['GET', 'POST'])
def register():
    mesage = ''
    if request.method == 'POST' and 'name' in request.form and 'password' in
request.form and 'email' in request.form :
        userName = request.form['name']
        password = request.form['password']
        email = request.form['email']
        role = request.form['role']
        country = request.form['country']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM user WHERE email = % s', (email, ))
        account = cursor.fetchone()
        if account:
            mesage = 'User already exists !'
        elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
            mesage = 'Invalid email address !'
        elif not userName or not password or not email:
            mesage = 'Please fill out the form !'
        else:
            cursor.execute('INSERT INTO user VALUES (NULL, % s, % s, % s, % s, %
s)', (userName, email, password, role, country))
            mysql.connection.commit()
            mesage = 'New user created!'
    elif request.method == 'POST':
        mesage = 'Please fill out the form !'
    return render_template('register.html', mesage = mesage)
```

Implement User Listing

We will create `templates/users.html` template file and create HTML to display user list. Then implement to loop through user data and display list with details.

```
<table class="table table-striped">
    <thead>
      <tr>
        <th>Name</th>
        <th>Email</th>
        <th>Role</th>
                <th>Country</th>
                <th></th>
                <th></th>
                <th></th>
                <th></th>
      </tr>
    </thead>
    <tbody>
          {% for user in users %}
      <tr>
        <td>{{user.name}}</td>
        <td>{{user.email}}</td>
        <td>{{user.role}}</td>
                <td>{{user.country}}</td>
                <td><a href="{{url_for('view', userid=user.userid)}}" class="btn
btn-success">View</a></td>
                <td><a href="{{url_for('edit', userid=user.userid)}}" class="btn
btn-primary">Edit</a></td>
                <td><a href="{{url_for('password_change', userid=user.userid)}}"
class="btn btn-warning">Change Password</a></td>
                <td><a href="{{url_for('delete', userid=user.userid)}}" class="btn
btn-danger">Delete</a></td>
      </tr>
        {% endfor %}
    </tbody>
</table>
```

We will create function `users()` in `app.py` and implement functionality to get all users data from database table and pass to template `users.html` to display list.

```
@app.route("/users", methods =['GET', 'POST'])
def users():
    if 'loggedin' in session:
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM user')
        users = cursor.fetchall()
        return render_template("users.html", users = users)
    return redirect(url_for('login'))
```

Implement User Edit Section

We will create template file `templates/edit.html` and create user edit form.

```
<form action="{{ url_for('edit') }}" method="post">
        {% if mesage is defined and mesage %}
                <div class="alert alert-warning">{{ mesage }}</div>
        {% endif %}
        <div class="form-group">
                <label for="name">Name:</label>
                <input type="text" class="form-control" id="name" name="name"
value="{{ editUser.name }}">
        </div>
        <div class="form-group">
                <label for="role">Role:</label>
                <select class="form-control" id="role" name="role">
                        <option value="admin" {% if editUser.role == 'admin'
%}selected{% endif %}>Admin</option>
                        <option value="user"  {% if editUser.role == 'user'
%}selected{% endif %}>User</option>
                </select>
        </div>
        <div class="form-group">
                <label for="country">Country:</label>
                <input type="text" class="form-control" id="country"
name="country" value="{{ editUser.country }}">
        </div>

        <input type="hidden" id="userid" name="userid" value="{{ editUser.userid
}}">
        <button type="submit" class="btn btn-primary">Save</button>
</form>
```

We will create `edit()` function in `app.py` and implement user edit functionality and render `edit.html` template.

```
@app.route("/edit", methods =['GET', 'POST'])
def edit():
    msg = ''
    if 'loggedin' in session:
        editUserId = request.args.get('userid')
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM user WHERE userid = % s', (editUserId, ))
        editUser = cursor.fetchone()
        if request.method == 'POST' and 'name' in request.form and 'userid' in
request.form and 'role' in request.form and 'country' in request.form :
            userName = request.form['name']
            role = request.form['role']
            country = request.form['country']
            userId = request.form['userid']
            if not re.match(r'[A-Za-z0-9]+', userName):
                msg = 'name must contain only characters and numbers !'
            else:
                cursor.execute('UPDATE user SET  name =% s, role =% s, country =%
s WHERE userid =% s', (userName, role, country, (userId, ), ))
                mysql.connection.commit()
                msg = 'User updated !'
                return redirect(url_for('users'))
        elif request.method == 'POST':
            msg = 'Please fill out the form !'
        return render_template("edit.html", msg = msg, editUser = editUser)
    return redirect(url_for('login'))
```

Implement User Password Change

We will create template file `templates/password_change.html` and create password change form to implement functionality.

```
<form action="{{ url_for('password_change') }}" method="post">
        {% if mesage is defined and mesage %}
                <div class="alert alert-warning">{{ mesage }}</div>
        {% endif %}
        <div class="form-group">
                <label for="country">Password:</label>
                <input type="password" class="form-control" id="password"
name="password">
        </div>
        <div class="form-group">
                <label for="country">Confirm Password:</label>
                <input type="password" class="form-control" id="confirm_pass"
name="confirm_pass">
        </div>
        <input type="hidden" id="userid" name="userid" value="{{ changePassUserId
}}">
        <button type="submit" class="btn btn-primary">Update</button>
</form>
```

We will create a function `password_change()` in `app.py` and implement user password change functionality.

```
@app.route("/password_change", methods =['GET', 'POST'])
def password_change():
    mesage = ''
    if 'loggedin' in session:
        changePassUserId = request.args.get('userid')
        if request.method == 'POST' and 'password' in request.form and
'confirm_pass' in request.form and 'userid' in request.form  :
            password = request.form['password']
            confirm_pass = request.form['confirm_pass']
            userId = request.form['userid']
            if not password or not confirm_pass:
                mesage = 'Please fill out the form !'
            elif password != confirm_pass:
                mesage = 'Confirm password is not equal!'
            else:
                cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
                cursor.execute('UPDATE user SET  password =% s WHERE userid =% s',
(password, (userId, ), ))
                mysql.connection.commit()
                mesage = 'Password updated !'
        elif request.method == 'POST':
            mesage = 'Please fill out the form !'
        return render_template("password_change.html", mesage = mesage,
changePassUserId = changePassUserId)
    return redirect(url_for('login'))
```

Implement View User Details

We will create `templates/view.html` file and implement to display user details.

```
<h3>User Details</h3>
<br>
<h4>{{user.name}}</h4>
<p><strong>Email: </strong> {{user.email}}. </p>
<p><strong>Role: </strong> {{user.role}} </p>
<p><strong>Skills: </strong> {{user.country}}</p>
```

We will create function `view()` in `app.py` and implement to get user details from database table and pass to template `view.html` to display user details.

```
@app.route("/view", methods =['GET', 'POST'])
def view():
    if 'loggedin' in session:
        viewUserId = request.args.get('userid')
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM user WHERE userid = % s', (viewUserId, ))
        user = cursor.fetchone()
        return render_template("view.html", user = user)
    return redirect(url_for('login'))
```

You can download the complete source code of project from the Download link below.
Download