Flask + MySQL

Preparación

Diríjase a la guía Python Flask-MySQL para instalar mysql, las librerías de soporte para la conexión entre flask y mysql y la creación de la base de datos.

User Registration

Add new imports to views.py

```
# views.py

from flask import render_template, flash,
redirect, url_for, session, request, logging
from flask_mysqldb import MySQL
from wtforms import Form, StringField,
TextAreaField, PasswordField, validators
from passlib.hash import sha256_crypt
from functools import wraps

from app import app
from app.data import Articles
```

Add MySQL Configuration to Views.py

```
# Config MySQL
app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] ='Autonoma123*'

app.config['MYSQL_DB'] = 'myflaskapp'
app.config['MYSQL_CURSORCLASS'] =
'DictCursor'

#INIT MYSQL
mysql = MySQL(app)
```

Add MySQL Configuration to Views.py

```
# Config MySQL
app.config['MYSQL_HOST']
app.config['MYSQL_USER']
app.config['MYSQL_PASSWORD'] ='Autonoma123*'

app.config['MYSQL_DB'] = 'myflaskapp'
app.config['MYSQL_CURSORCLASS'] =
'DictCursor'

#INIT MYSQL
mysql = MySQL(app)
```

Change the password accordingly

Create a class for the register form in views.py

```
class RegisterForm(Form):
        name = StringField('Name', [validators.Length(min=1, max=50)])
        username = StringField('Username', [validators.Length(min=4,
max=25)])
        email = StringField('Email', [validators.Length(min=6, max=50)])
        password = PasswordField('Password', [
                validators.DataRequired(),
                validators.EqualTo('confirm', message='Password do not
match')
                ])
        confirm = PasswordField('Confirm Password')
```

Create the register route in views.py

```
 @app.route('/register', methods=['GET', 
'POST'])
def register():
        form = RegisterForm(request.form)
        if request.method == 'POST' and 
form.validate():
                name = form.name.data
                email = form.email.data
                username = form.username.data
                password = 
sha256_crypt.encrypt(str(form.password.data))

                # Create Cursor
                cur = mysql.connection.cursor()

                # Execute Query
                cur.execute("INSERT INTO 
users(name, email, username, password) 
VALUES(%s, %s, %s, %s)", (name, email, 
username,password))
```

```
 # Commit to DB

mysql.connection.commit()

                # Close connection
                cur.close()

                flash('You are now 
registered and can log in', 
'success')

                return 
redirect(url_for('index'))

        return 
render_template('register.html', 
form=form)
```

Create includes directory for helpers my-project/app/templates/includes/

```
$cd templates


$mkdir includes
```

Create form helper in myproject/app/templates/includes/_formhelpers.html

```
{% macro render_field(field) %}
  {{ field.label }}
  {{ field(**kwargs)|safe }}
  {% if field.errors %}
    {% for error in field.errors %}
      <span class="help.inline">{{ error }}</span>
    {% endfor %}
  {% endif %}
{% endmacro %}
```

Create messages helper in myproject/app/templates/includes/_messages.html

```
{% with messages = get_flashed_messages(with_categories=true) %}
  {% if messages %}
      {% for category, message in messages %}
        <div class="alert alert-{{ category }}">{{ message }}</div>
      {% endfor %}
  {% endif %}
{% endwith %}

{% if error %}
  <div class="alert alert-danger">{{error}}</div>
{% endif %}
{% if msg %}
  <div class="alert alert-success">{{msg}}</div>
{% endif %}
```

Add import in base.html

```
<body>
  <div class="container">
    {% include 'includes/_messages.html' %}
     <div class="header clearfix">
      <nav>
        <ul class="nav nav-pills pull-right">
          <li role="presentation"><a href="/">Home</a></li>
```

Create the register.html template

```
<!-- register.html-->
{% extends "base.html" %}
{% block title %}Register{% endblock %}
{% block body %}
<h1>Register</h1>
{% from "includes/_formhelpers.html" import
render_field %}
<form method="POST" action="">
  <div class="form-group">
    {{render_field(form.name, class_="form-control")}}
  </div>
  <div class="form-group">
    {{render_field(form.email, class_="form-control")}}
  </div>
  <div class="form-group">
    {{render_field(form.username, class_="form-
control")}}
 </div>
  <div class="form-group">
    {{render_field(form.password,
class_="form-control")}}
  </div>
  <div class="form-group">
    {{render_field(form.confirm,
class_="form-control")}}
  </div>
  <p> <input type="submit" class="btn
btn-primary" value="Submit"> </p>
</form>
{% endblock %}
```

Add Link in base.html, add secrect

In base.html:

```
<li role="presentation"><a href="/register">Register</a></li>
```

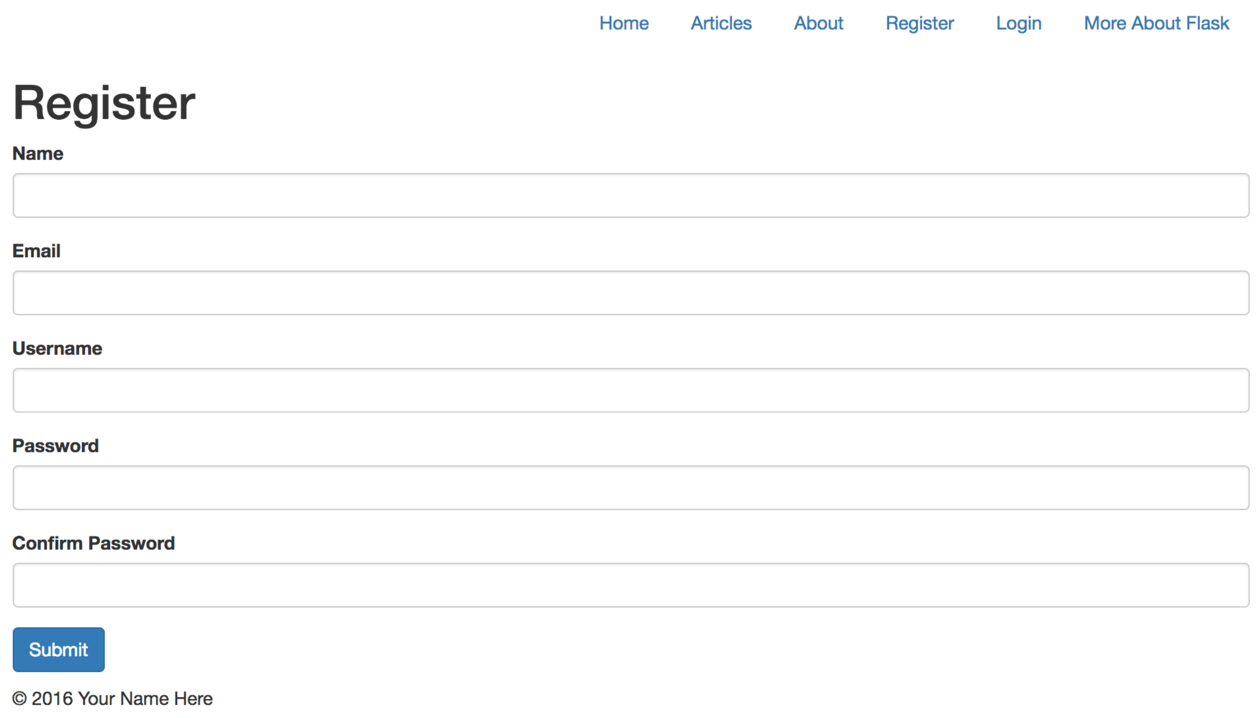
In views.py:

```
Articles = Articles()
…
app.secret_key='secrect123'
…
# Config MySQL
app.config['MYSQL_HOST'] = 'localhost'
```

Register a User

Run in Deguggin mode:

```
export FLASK_ENV=development
python3 -m flask run --host=0.0.0.0
```



Register

Name

Email

Username

Password

Confirm Password

© 2016 Your Name Here

Verify in Database

```
mysql -u root -p

mysql> use myflaskapp;

mysql> select * from users;
```

User Login

Create the login route in views.py

```
# User login
@app.route('/login', methods=['GET', 'POST'])
def login():
        if request.method == 'POST':

                # Get Form Fields
                username = request.form['username']
                password_candidate = request.form['password']

                # Create cursor
                cur = mysql.connection.cursor()

                # Get user by username
                result = cur.execute("SELECT * FROM users WHERE username =%s",[username])

                if result > 0:
                        # Get stored hash
                        data = cur.fetchone()
                        password = data['password']

                        # Compare Passwords
                        if sha256_crypt.verify(password_candidate, password):
                                # Passed
                                session['logged_in'] = True
                                session['username'] = username
                                flash('You are logged in', 'success')
                                app.logger.info('PASSWORD MATCHED')
                                return redirect(url_for('dashboard'))
```

Create the login route in views.py

```
                        else:
                                app.logger.info('PASSWORD NO MATCHED')
                                error = 'Invalid login'
                                return render_template('login.html', 
error=error)

                        # Close connection
                        cur.close()

                else:
                        app.logger.info('NO USER')
                        error = 'Username not found'
                        return render_template('login.html', error=error)


        return render_template('login.html')
```

Create route to dashboard in views.py

```
# Check if user logged in
def is_logged_in(f):
        @wraps(f)
        def wrap(*args, **kwargs):
                if 'logged_in' in session:
                        return f(*args, **kwargs)
                else:
                        flash('Unauthorized, Plese login', 'danger')
                        return redirect(url_for('login'))
        return wrap

# Dashboard
@app.route('/dashboard')
@is_logged_in
def dashboard():
        return render_template('dashboard.html')
```

Create route to logout in views.py

```
# Logout
@app.route('/logout')
def logout():
        session.clear()
        flash('You are now logged out','success')
        return redirect(url_for('login'))
```

Add Links in base.html

```
<nav>
          <ul class="nav nav-pills pull-right">
            <li role="presentation"><a href="/">Home</a></li>
            <li role="presentation"><a href="/articles">Articles</a></li>
            <li role="presentation"><a href="/about">About</a></li>

            {% if session.logged_in %}
                <li role="presentation"><a href="/dashboard">Dashboard</a></li>
                <li role="presentation"><a href="/logout">Logout</a></li>
            {% else %}
                <li role="presentation"><a href="/register">Register</a></li>
                <li role="presentation"><a href="/login">Login</a></li>
            {% endif %}
            <li role="presentation"><a href="http://flask.pocoo.org"
target="_blank">More About Flask</a></li>
          </ul>
</nav>
```

Create the login.html template

```
{% extends 'base.html' %}
{% block body %}
 <h1>Login</h1>
 <form action="" method="post">
   <div class="form-group">
     <label>Username</label>
     <input type="text" name="username"
class="form-control"
value="{{request.form.username}}">
   </div>
   <div class="form-group">
     <label>Password</label>
     <input type="password"
name="password" class="form-control"
value="{{request.form.password}}">
   </div>
```

```
<button type="submit" class="btn
btn-primary">Submit</button>
 </form>
{% endblock %}
```

Create the dashboard.html template

```
{% extends 'base.html' %}
{% block body %}
<h1>Dashboard <small> Welcome {{session.username}} </small> </h1>
<a class="btn btn-success" href="/add_article">Add Article</a>
<hr>
<table class="table table-striped">
  <tr>
    <th>ID</th>
    <th>Title</th>
    <th>Author</th>
    <th>Date</th>
    <th></th>
    <th></th>
  </tr>
```

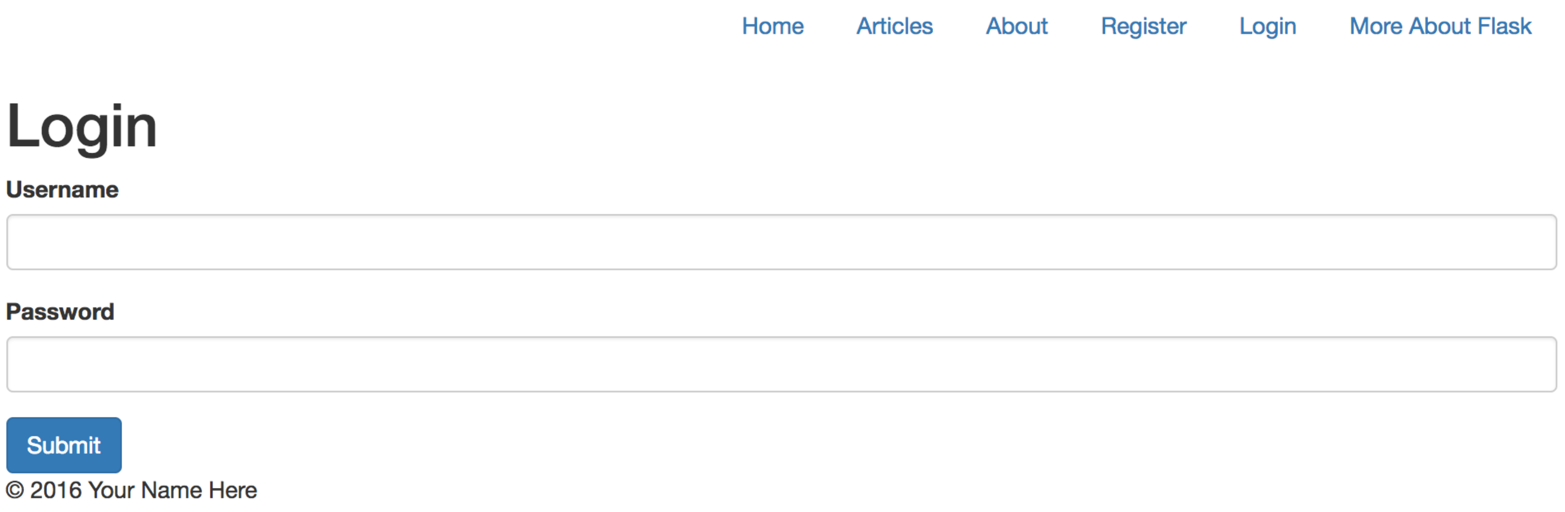
Create the dashboard.html template

```
{% for article in articles %}
  <tr>
  <td>{{article.id}}</td>  <!-- id -->
  <td>{{article.title}}</td>  <!-- title -->
  <td>{{article.author}}</td> <!-- author -->
  <td>{{article.create_date}}</td> <!-- create_date -->
  <td><a href="edit_article/{{article.id}}" class="btn btn-default pull-
right">Edit</a>
  <td>
    <form  action="{{url_for('delete_article', id=article.id)}}" method="post">
      <input type="hidden" name="_method" value="DELETE">
      <input type="submit" name="" value="Delete" class="btn btn-danger">
</form></td><td></td></tr>
{% endfor %}
</table> {% endblock %}
```

Test Login

Run in Deguggin mode:

```
export FLASK_ENV=development
python3 -m flask run --host=0.0.0.0
```



© 2016 Your Name Here

References

Instal MySQL:

https://www.hostinger.com/tutorials/how-to-install-mysql-on-centos-7

Change MySQL default Password:

https://dev.mysql.com/doc/refman/8.0/en/default-privileges.html

Python Flask from Scratch:

https://www.youtube.com/watch?v=zRwy8gtgJ1A&t=1239s

Flask 101: Adding, Editing and Displaying Data

https://www.blog.pythonlibrary.org/2017/12/14/flask-101-adding-editing-and-displaying-data/