Flask: Forms Handling

Recall

Have you got the list displayed by a loop? If it is, you got the right way to display a set of data to the users. If you cannot figure out how to do it, please contact your instructor.

UIC Calendar: login

Till now, we have got the index page finished. Let's move on to the login page. To simplify, the system will only allow a fixed combination of username and password, for example,

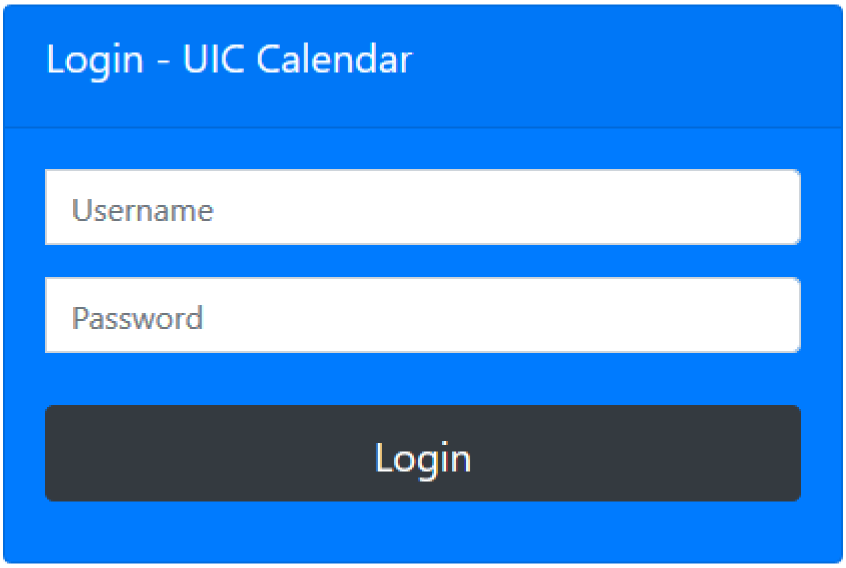
- Username: admin
- Password: admin

The username and password will be checked by the Python code you are going to write, and return a success or failed information which will also show on the login page.

Inside the `code` folder you downloaded, you should see a structure like this,

```
/application.py
/templates
  /index.html
  /query.html
  /login.html
  /admin.html
/static
  /bootstrap.min.css
  /bootstrap.min.map
```

Run the Flask app like the pervious notes, type `http://127.0.0.1:5000/login` in your browser, it should look like the picture below.



Look carefully on the code, you can find a message variable between the password box and the login button. This variable will be fed by the Python code you are going to write.

`POST` a Form

As you may know, to trigger a `POST` in the form, you need to change your HTML file a little bit.

Task 3.1: Now, please add the keyword on `login.html` to let the browser to `POST` your form.

Flask Form Functions

Flask does not directly handle forms. Instead, it depends on a third-party package calls `wtforms`. Now let's install this package by `pip`.

```
pip install wtforms
```

Create a Form with `wtforms`

Task 3.2: Read the following sample code, write the Python class `LoginForm` for the login form. Hint, two lines of code are enough and mind the `name` properties of the form components in `login.html`. The answer can be found in the complete example.

```
# Example code

from wtforms import Form, BooleanField, StringField, PasswordField, validators

class RegistrationForm(Form):
    username = StringField('Username', [validators.Length(min=4, max=25)])
    email = StringField('Email Address', [validators.Length(min=6, max=35)])
    password = PasswordField('New Password', [
        validators.DataRequired(),
        validators.EqualTo('confirm', message='Passwords must match')
    ])
    confirm = PasswordField('Repeat Password')
    accept_tos = BooleanField('I accept the TOS', [validators.DataRequired()])
```

The View Function

> The word "view" here is about software engineering. View functions can be considered as a set of functions handle template rendering in this project.

The form you just created at task 3.2 is the backbone of the form. We are now going to connect it to your template.

Go to our Python file, add the code below to the file.

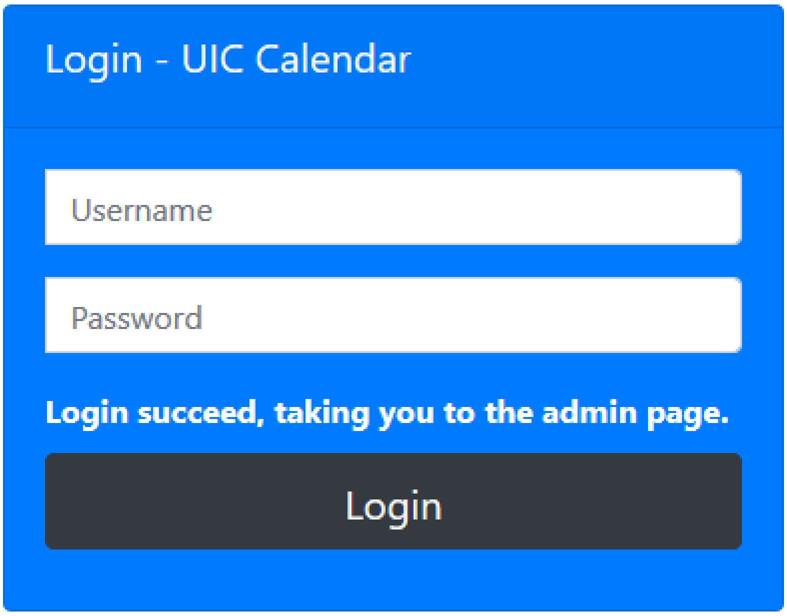
```
# application.py


from flask import requests

@app.route('/login', methods=['GET', 'POST'])
def login():
    form = LoginForm(request.form)
    if request.method == 'POST' and form.validate():
        # if the inputs are correct, do something
        # otherwise, do something else
    else:
        message = 'Test username: admin, password: admin'
        return render_template('login.html', message=message)
```

The above handles two types of HTTP requests. When the request is `POST`, we jump to form handling, otherwise, we display the login page.

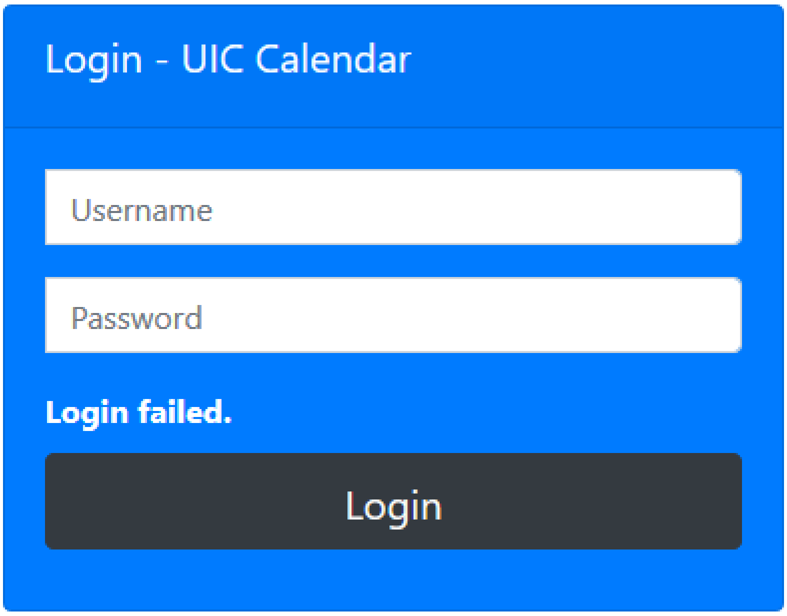
Task 3.3: To simplified, when the username and password are "admin", we can allow the user to log in. Please add the code for the view function `login()` to validate the input from the user. If it is correct, render `login.html` with success message, otherwise, render `login.html` with failure information. Hint, input data can be found in `form.item.data` attribute. The result of this task should like the pictures below.



Login - UIC Calendar

Login succeed, taking you to the admin page.

Username and password are correct.



Login - UIC Calendar

Login failed.

Username or password is incorrect.

References

- http://flask.pocoo.org/docs/1.0/patterns/wtforms/
- https://wtforms.readthedocs.io/en/stable/crash_course.html
- https://spacewander.github.io/explore-flask-zh/11-handling_forms.html