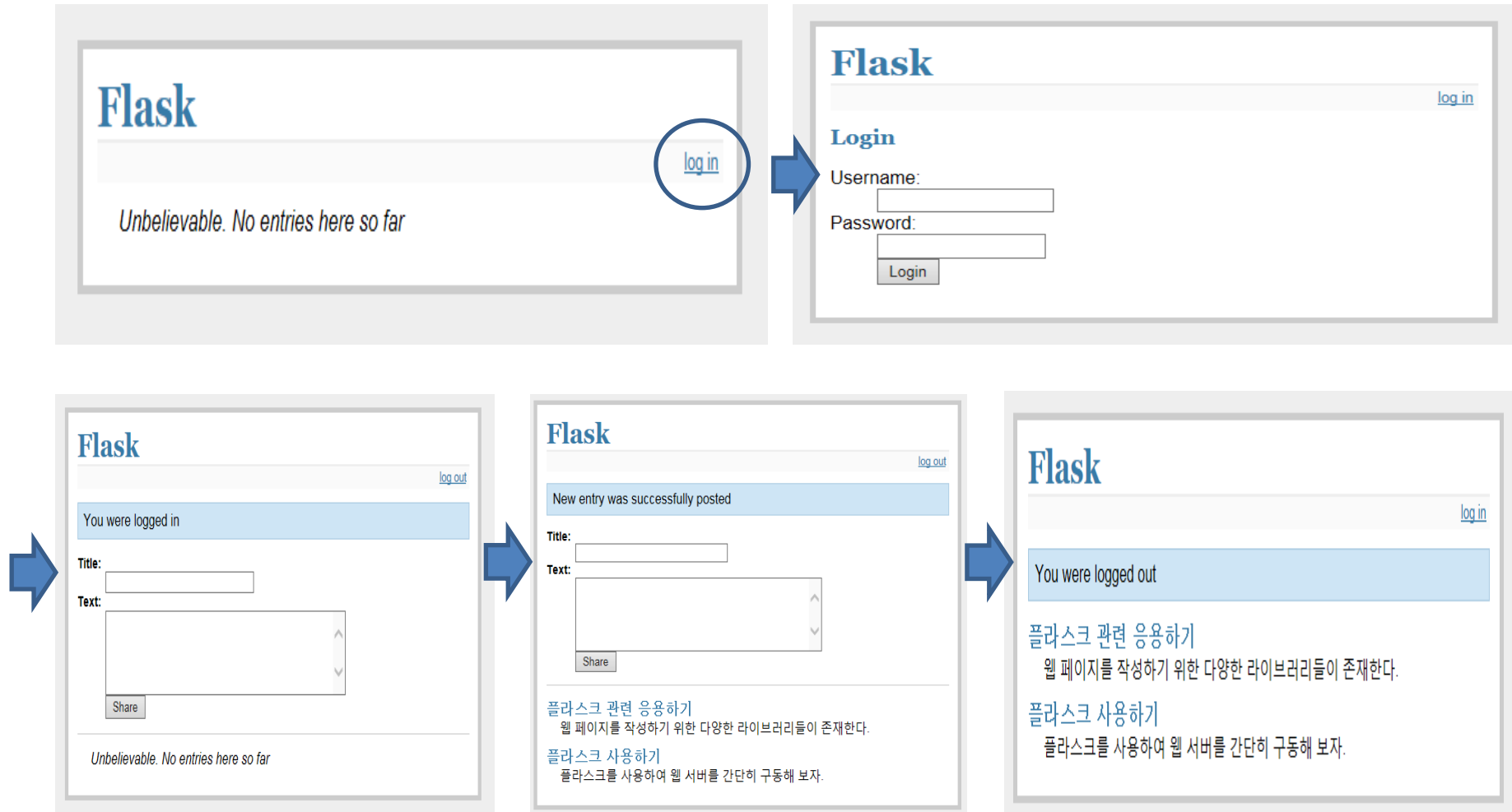마이크로 웹 프레임워크

플라스크(Flask) 2

DB 연동 및 로그인 예제

데이터베이스 스키마

schema.sql

```
drop table if exists entries;
create table entries (
   id integer primary key autoincrement,
   title string not null,
   text string not null
);
```

환경 설정하기

```
import sqlite3
from flask import Flask, request, session, g, redirect, url_for, ₩
    abort, render_template, flash
from contextlib import closing


# configuration
DATABASE = 'flask.db'
DEBUG = True
SECRET_KEY = 'development key'
USERNAME = 'admin'
PASSWORD = 'default'
```

#세션사용, 세션의 안전성 보장

응용프로그램 생성

app = Flask(__name__)
app.config.from_object(__name__)

*app.config는 플라스크 설정과 관련된 dictionary 객체
 -. from_object () 함수는 <mark>대문자로 설정된 값들을 config에 추가시킴</mark>

DB 관련 작업

```
def connect_db():
    return sqlite3.connect(app.config['DATABASE'])

def init_db():
    db = connect_db()
    with open('schema.sql') as f:
        db.cursor().executescript(f.read())
    db.commit()
```

```
def init_db():
    with closing(connect_db()) as db:
        with app.open_resource('schema.sql', mode='r') as f:
            db.cursor().executescript(f.read())
        db.commit()
```

데이터베이스 커넥션 요청하기

*request 실행 전에 호출

@app.before_request
def before_request():
    g.db = connect_db()

    * g : flask의 전역 클래스 인스턴스

*request 의 마지막에 호출
@app.teardown_request
def teardown_request(exception):
    g.db.close()

작성된글 보여주기

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

def show_entries():    * cursor 객체를 생성한후 질의를 수행하는 비표준 방법

    cur = g.db.execute('select title, text from entries order by id desc')

    entries = [dict(title=row[0], text=row[1]) for row in cur.fetchall()]

    print(entries)

    return render_template('show_entries.html', entries=entries)
```

로그인 하기

```
@app.route('/login', methods=['GET', 'POST'])
def login():
    error = None
    if request.method == 'POST':
        if request.form['username'] != app.config['USERNAME']:
            error = 'Invalid username'
        elif request.form['password'] != app.config['PASSWORD']:
            error = 'Invalid password'
        else:
            session['logged_in'] = True
            flash('You were logged in')
            return redirect(url_for('show_entries'))
    return render_template('login.html', error=error)
```

*flash() : 다음 request에 메시지를 전달하는 함수
-. 받는 쪽에서는 get_flashed_messages() 함수 이용

새로운 글 추가하기

```
@app.route('/add', methods=['POST'])
def add_entry():
    if not session.get('logged_in'):
        abort(401)
    g.db.execute('insert into entries (title, text) values (?, ?)',
            [request.form['title'], request.form['text']])
    g.db.commit()
    flash('New entry was successfully posted')
    return redirect(url_for('show_entries'))
```

*abort() : HTTPException 발생

로그 아웃

```python
@app.route('/logout')
def logout():
    session.pop('logged_in', None)
    flash('You were logged out')
    return redirect(url_for('show_entries'))
```

layout.html

```
<!doctype html>
<title>Flaskr</title>
<link rel=stylesheet type=text/css href="{{ url_for('static', filename='stylesheet.css') }}">
<div class=page>
    <h1>Flask</h1>
    <div class=metanav>
        {% if not session.logged_in %}
        <a href="{{ url_for('login') }}">log in</a>
        {% else %}
        <a href="{{ url_for('logout') }}">log out</a>
        {% endif %}
    </div>
    {% for message in get_flashed_messages() %}
    <div class=flash>{{ message }}</div>
    {% endfor %}
    {% block body %}{% endblock %}
</div>
```

show_entries.html

```
{% extends "layout.html" %}
{% block body %}
  {% if session.logged_in %}
<form action="{{ url_for('add_entry') }}" method=post class=add-entry>
    <dl>
      <dt>Title:
      <dd><input type=text size=30 name= title>
      <dt>Text:
      <dd><textarea name=text rows=5 cols=40></textarea>
      <dd><input type=submit value=Share>
    </dl>
</form>
  {% endif %}
<ul class=entries>
    {% for entry in entries %}
    <li>
      <h2>{{ entry.title }}</h2>{{ entry.text|safe }}
    {% else %}
    <li>
      <em>Unbelievable.  No entries here so far</em>
    {% endfor %}
</ul>
{% endblock %}
```

login.html

```
{% extends "layout.html" %}
{% block body %}
<h2>Login</h2>
  {% if error %}<p class=error>
   <strong>Error:</strong> {{ error }}{% endif %}
  <form action="{{ url_for('login') }}" method=post>
     <dl>
        <dt>Username:
        <dd><input type=text name=username>
        <dt>Password:
        <dd><input type=password name=password>
        <dd><input type=submit value=Login>
     </dl>
  </form>
  {% endblock %}
```

stylesheet.css

```css
body            { font-family: sans-serif; background: #eee; }
a, h1, h2       { color: #377BA8; }
h1, h2          { font-family: 'Georgia', serif; margin: 0; }
h1              { border-bottom: 2px solid #eee; }
h2              { font-size: 1.2em; }

.page           { margin: 2em auto; width: 35em; border: 5px solid #ccc;
                  padding: 0.8em; background: white; }
.entries        { list-style: none; margin: 0; padding: 0; }
.entries li     { margin: 0.8em 1.2em; }
.entries li h2  { margin-left: -1em; }
.add-entry      { font-size: 0.9em; border-bottom: 1px solid #ccc; }
.add-entry dl   { font-weight: bold; }
.metanav        { text-align: right; font-size: 0.8em; padding: 0.3em;
                  margin-bottom: 1em; background: #fafafa; }
.flash          { background: #CEE5F5; padding: 0.5em;
                  border: 1px solid #AACBE2; }
.error          { background: #F0D6D6; padding: 0.5em; }
```