

파이썬 플라스크로 배우는 웹프로그래밍 #4

ABCD Foundation



스노우키위, 한 성 일

4. 서비스 구축

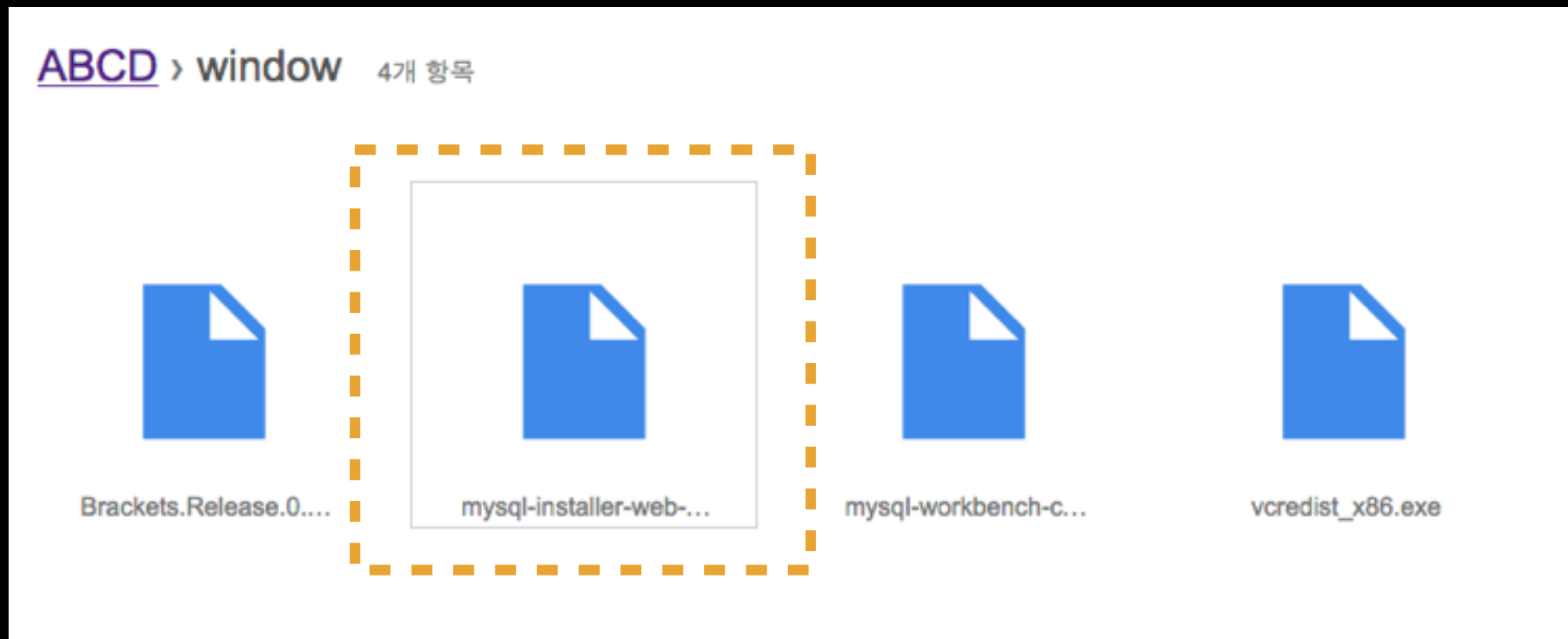


0. 프롤로그

정정

- Commercial 버전이 설치됨
- 제어판 > 프로그램 제거 또는 변경
- MySQL Server 5.6 제거
- C:\ProgramData\MySQL\MySQL Server 5.6 폴더 삭제

https://drive.google.com/folderview?id=0B_dqOWjvHbO9Yk5rSWIBVDd4dEU&usp=sharing&tid=0B_dqOWjvHbO9czFhMHY3VmtsWkk



1. 프로세스

웹 서비스 구축 프로세스

1 디자인



2 html 제작



3 삽질 시작



2. 구성

가상환경 작업(Virtual Environment)



```
C:\Users\mark>cd \flask-space
```

```
C:\flask-space>mkdir ch4
```

```
C:\flask-space>cd ch4
```

```
C:\flask-space\ch4>virtualenv flask
```

```
C:\flask-space\ch4>flask\Scripts\pip install flask
```

```
C:\flask-space\ch4>flask\Scripts\pip install flask-sqlalchemy
```

```
C:\flask-space\ch4>flask\Scripts\easy_install MySQL-python
```


가상환경 작업(Virtual Environment)



```
$ cd ~
```

```
$ cd Documents/flask-space
```

```
$ mkdir ch4
```

```
$ cd ch4
```

```
$ virtualenv flask
```

```
$ flask/bin/pip install flask
```

```
$ flask/bin/pip install flask-sqlalchemy
```

가상환경 작업(Virtual Environment)



```
$ export PATH=$PATH:/usr/local/mysql/bin/
```

```
$ flask/bin/pip install MySQL-python
```

```
$ sudo ln -s /usr/local/mysql/lib/libmysqlclient.18.dylib /usr/lib/
```

디렉토리 구조

- app, static, templates 폴더를 생성해 주세요 !!

ch4

app

__init__.py

controllers.py

database.py

models.py

static

templates

run.py

flask - 생성되어있음

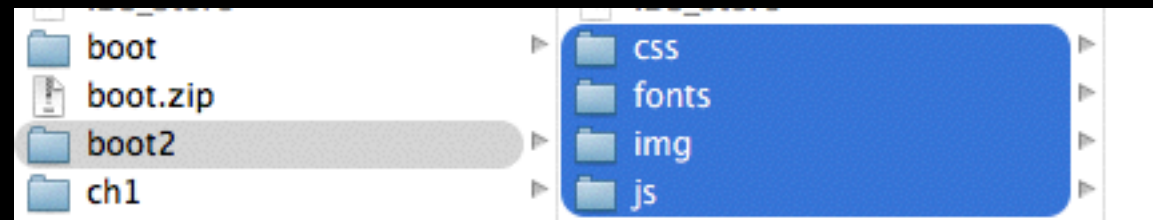
기본작업1

1 퍼블리싱 데이터 다운로드

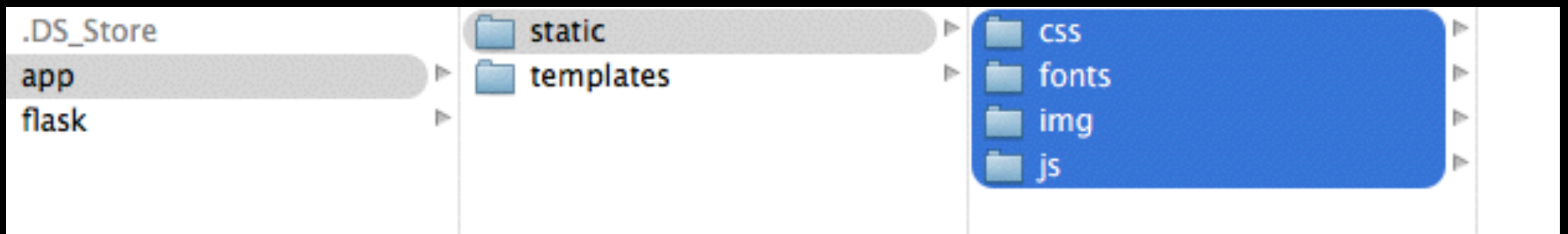
https://drive.google.com/folderview?id=0B_dqOWjvHbO9STBKd0JfMIA1TmM&usp=sharing



2 boot2 압축해제후 폴더의 css, font, img, js 복사

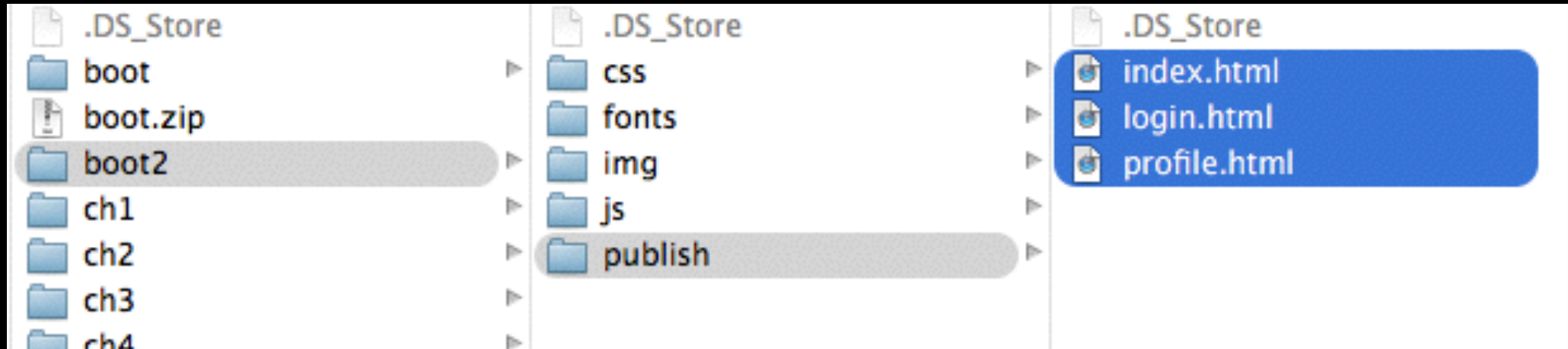


3 ch4/app/static 폴더로 붙여넣기

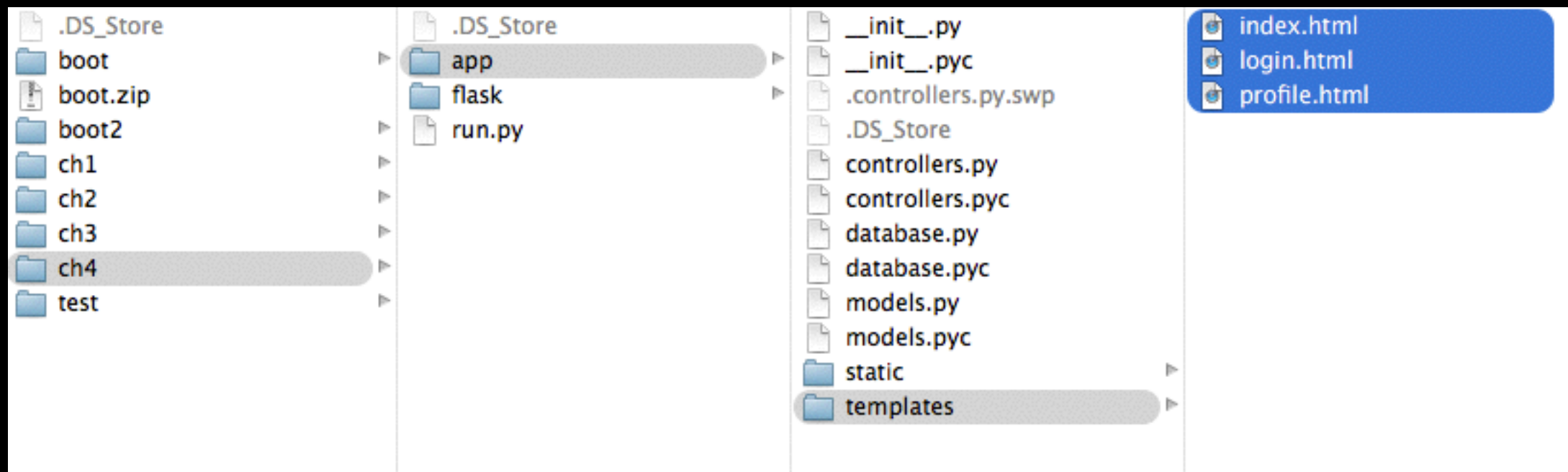


기본작업2

- 1 boot2 폴더의 publish 폴더의 index.html, login.html, profile.html 폴더 복사



- 2 ch4/app/template 폴더로 붙여넣기



3. 데이터베이스 설정



데이터베이스 설정

```
SHOW DATABASES;  
USE abcd;
```




데이터베이스 설정

1

```
CREATE TABLE users  
(  
    id INTEGER NOT NULL AUTO_INCREMENT,  
    name VARCHAR(50) NOT NULL,  
    password VARCHAR(50) NOT NULL,  
    email VARCHAR(120) NOT NULL,  
    image VARCHAR(200) NOT NULL,  
    company VARCHAR(100),  
    gender VARCHAR(50),  
    location VARCHAR(100),  
    tel VARCHAR(100),  
    description VARCHAR(500),  
    PRIMARY KEY(id),  
    UNIQUE (email)  
);
```

2

```
SELECT * FROM users
```

id	name	email	image	description	password	company	gender	location	tel
----	------	-------	-------	-------------	----------	---------	--------	----------	-----



데이터베이스 설정

기초데이터 입력

1 SELECT * FROM users;

2

id	name	email	image	description	password	company	gender	location	tel
1	한성일	iheart79@gmail.com	img/rabbit.jpg	바보다..	1234	snowkiwi	남자	서울	010-2926-1837
2	장그래	test@gmail.com	img/favicon.ico	장그래	1234	실업	남자1	강원도	010-2926-1837
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

users 4

3 Apply Revert

4. 구현

구동 프로세스

url 구조

/	/profile/사용자 아이디	/login-form
회원리스트	프로필 화면	로그인 화면
	/add	/login
	포스트 등록	로그인
	/delete/사용자 아이디/포스트 아이디	
	포스트 삭제	

Flask

구동파일

run.py

```
#!/flask/bin/python
# -*- coding: utf-8 -*-
from app import app

app.secret_key = 'abcd_key'
app.config['SQLALCHEMY_ECHO'] = True

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)
```

Flask

구동파일

app/___init___py

```
from flask import Flask
```

```
app = Flask(__name__)
```

```
from app import controllers
```


Flask

Database

app/database.py

```
# -*- coding: utf-8 -*-
from sqlalchemy import create_engine
from sqlalchemy.orm import scoped_session, sessionmaker
from sqlalchemy.ext.declarative import declarative_base

engine = create_engine('mysql://abcd:1234@127.0.0.1/abcd?
                        charset=utf8', convert_unicode=True)
db_session = scoped_session(sessionmaker(autocommit=False,
                                          autoflush=False,
                                          bind=engine))

Base = declarative_base()
Base.query = db_session.query_property()

def init_db():
    # import all modules here that might define models so that
    # they will be registered properly on the metadata.  Otherwise
    # you will have to import them first before calling init_db()
    import models
    Base.metadata.create_all(bind=engine)
```

Flask

Database

쿼리스트링 의미

mysql://abcd:1234@127.0.0.1/abcd?charset=utf8

mysql://db계정:db비밀번호@127.0.0.1/데이터베이스?charset=utf8

Flask

Model 1

app/models.py

```
# -*- coding: utf-8 -*-  
import datetime  
from sqlalchemy import Column, Integer, String, Text, DateTime,  
                        ForeignKey  
from sqlalchemy.orm import sessionmaker, relationship  
from database import Base
```

users 테이블에 맵핑

Model 2

app/models.py

Flask

```
class User(Base):
```

```
    __tablename__ = 'users'
```

```
    id = Column(Integer, primary_key=True)
```

```
    name = Column(String(50), unique=False)
```

```
    password = Column(String(50), unique=False, nullable=False)
```

```
    email = Column(String(120), unique=True, nullable=False)
```

```
    image = Column(String(200), unique=False)
```

```
    company = Column(String(100), unique=False)
```

```
    gender = Column(String(50), unique=False)
```

```
    location = Column(String(100), unique=False)
```

```
    tel = Column(String(100), unique=False)
```

```
    description = Column(String(500), unique=False)
```

```
posts = relationship('Post', backref='user', lazy='dynamic')
```

```
def __init__(self, name=None, password=None, email=None,  
              image=None, company=None, gender=None, location=None,  
              tel=None, description=None):
```

```
    self.name = name
```

```
    self.password = password
```

```
    self.email = email
```

```
    self.image = image
```

```
    self.company = company
```

```
    self.gender = gender
```

```
    self.location = location
```

```
    self.tel = tel
```

```
    self.description = description
```

Flask

Model 3

app/models.py

posts 테이블에 맵핑

```
class Post(Base):
    __tablename__ = 'posts'
    id = Column(Integer, primary_key=True)
    contents = Column(Text, unique=False)
    userid = Column(Integer, nullable=False)
    writer = Column(Integer, ForeignKey('users.id'),
                     nullable=False)
    date = Column(DateTime, default=datetime.datetime.utcnow,
                  unique=False)
    users = relationship('User', backref='post', lazy='joined')

    def __init__(self, contents=None, userid=None, writer=None):
        self.contents = contents
        self.userid = userid
        self.writer = writer
```

Flask

로그확인

```
app.logger.debug(request.form['email'])  
app.logger.debug(user.name)
```

Flask

Controller

app/**controllers.py**

```
# -*- coding: utf-8 -*-
from database import init_db, db_session
from flask import Flask, request, session,
    redirect, url_for, abort, render_template, flash
from models import User, Post

from app import app

# 메인페이지
@app.route('/')
@app.route('/index')
def index():
    users = User.query.all()
    return render_template('index.html', users=users)
```


Flask

Controller

app/**controllers.py**

```
# 프로필
```

```
@app.route('/profile/<int:userId>')
```

```
def profile(userId):
```

```
    # 사용자 정보
```

```
    user = User.query.filter_by(id=userId).first()
```

```
# 컨텐츠 리스트
```

```
posts = Post.query.filter(Post.userid==User.id).order_by(Post.id.desc()).all();
```

```
#app.logger.debug(len(posts))
```

```
return render_template('profile.html', user=user, posts=posts, userId=userId)
```

Flask

Controller

app/`controllers.py`

```
# 로그인 화면
@app.route('/login-form')
def loginForm():
    return render_template('login.html')
```

Flask

Controller

app/**controllers.py**

로그인

```
@app.route('/login', methods=['POST'])
```

```
def login():
```

```
    if request.method == 'POST':
```

```
        email = request.form['email']
```

```
        password = request.form['password']
```

```
        user = User.query.filter_by(email=email).filter_by(password=password).first()
```

```
        if (user):
```

```
            session['logged_in'] = True
```

```
            session['id'] = user.id
```

```
            session['email'] = user.email
```

```
            session['name'] = user.name
```

```
            session['image'] = user.image
```

```
            return redirect("/profile/" + str(user.id))
```

```
        else:
```

```
            return '로그인 정보가 맞지 않습니다.'
```

```
else:
```

```
    return '잘못된 접근'
```

Flask

Controller

app/`controllers.py`

```
# 로그아웃
@app.route('/logout')
def logout():
    session['logged_in'] = False
    session.clear()
    return redirect(url_for('index'))
```

Flask

Controller

app/controllers.py

컨텐츠 등록

```
@app.route('/add/<int:userId>', methods=['POST'])
def add(userId):
    if request.method == 'POST':
        if (session['logged_in']):
            post = Post(request.form['contents'], userId,
                        session['id'])
            db_session.add(post)
            db_session.commit()
            return redirect("/profile/" + str(userId))
        else :
            return '로그인해주세요'
    else:
        return '잘못된 접근'
```

Flask

Controller

app/controllers.py

컨텐츠 삭제

```
@app.route('/delete/<int:userId>/<int:postId>')
def delete(userId, postId):
    if (session['logged_in']) :
        Post.query.filter_by(id=postId).delete()
        return redirect("/profile/" + str(userId))
    else:
        return '잘못된 접근'
```

Flask Jinja2

Template

app/templates/index.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="shortcut icon"
        href="{{ url_for('static', filename='img/favicon.ico') }}">
  <title>홈</title>
  <!-- 부트스트랩 코어 CSS -->
  <link href="{{ url_for('static', filename='css/bootstrap.css') }}"
        rel="stylesheet">
  <link href="{{ url_for('static', filename='css/custom-index.css') }}"
        rel="stylesheet">
</head>
<body>
  <div class="container">
    <h2>등록된 사용자</h2>
    {% for item in users %}
      <a href="/profile/{{item.id}}">{{item.name}}</a><br/>
    {% endfor %}
  </div>
  <!-- jQuery 필수 -->
  <script src="http://code.jquery.com/jquery.js"></script>
  <!-- 부트스트랩 코어 자바스크립트 -->
  <script src="{{ url_for('static', filename='js/bootstrap.min.js') }}">
  </script>
</body>
</html>
```


app/templates/login.html

```
<body>
  <div class="container">
    <form class="form-signin" action="/login" method="post">
      <h3 class="form-signin-heading">로그인</h3>
      <input type="text" name="email" class="form-control"
        placeholder="이메일" autofocus>
      <input type="password" name="password" class="form-control"
        placeholder="패스워드">
      <button class="btn btn-lg btn-primary btn-block" type="submit">
        로그인
      </button>
    </form>
  </div>
</body>
```

```
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
  <!-- 브랜드와 토글을 모바일을 위해 그룹화 합니다. -->
  <div class="navbar-header">
    <button type="button" class="navbar-toggle"
      data-toggle="collapse"
      data-target=".navbar-ex1-collapse">
      <span class="sr-only">Toggle navigation</span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
      <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="#">ABCD</a>
    {% if session.logged_in %}
      <p class="navbar-text">{{session.name}}님 환영합니다.</p>
    {% endif %}
  </div>

  <!-- 토글되는 네비게이션 링크들, 폼, 기타 콘텐츠를 모아놓습니다. -->
  <div class="collapse navbar-collapse navbar-ex1-collapse">
    <ul class="nav navbar-nav">
      {% if not session.logged_in %}
        <li><a href="/login-form">로그인</a></li>
      {% else %}
        <li><a href="/logout">로그아웃</a></li>
      {% endif %}
    </ul>
    <ul class="nav navbar-nav navbar-right"></ul>
  </div>
</nav>
```

Flask Jinja2

Template

app/templates/profile.html

```
<div class="row">
<!-- 개인정보 부분 -->
<!-- xs 18, md 12 -->
<div class="col-md-4">
    <div class="well">
        
        <h3>{{user.name}}</h3>
        <h5>{{user.company}}</h5>
        <div>
            <span class="glyphicon glyphicon-star"></span>
            전화 : {{user.tel}}
        </div>
        <div>
            <span class="glyphicon glyphicon-star"></span>
            성별 : {{user.gender}}
        </div>
        <div>
            <span class="glyphicon glyphicon-star"></span>
            지역 : {{user.location}}
        </div>
        <h5>{{user.description}}</h5>
    </div>
</div>
```

app/templates/profile.html

```
{% if session.logged_in %}
<!-- 입력 부분 -->
<div class="well">
    <form class="form-signin" action="/add/{{userId}}" method="post">
        <textarea class="form-control" name="contents"
            rows="3" placeholder="오늘 어때요?"></textarea>
        <button type="submit" class="btn btn-primary btn-block">
            글쓰기
        </button>
    </form>
</div>
{% endif %}
```

Flask Jinja2

Template

app/templates/profile.html

```
<!-- 컨텐츠 부분 : loop -->
<div class="well">
    {% for item in posts %}
    <div class="media">
        <a class="pull-left" href="/profile/{{item.users.id}}">
            
        </a>
        <div class="media-body">
            {{ item.contents }}
            <div>
                <b> {{ item.users.name }} </b>
                {{ item.date }}
                {% if session.id == item.users.id or session.id == userId %}
                <a href="/delete/{{userId}}/{{item.id}}">
                    <span class="glyphicon glyphicon-remove"></span>
                </a>
                {% endif %}
            </div>
        </div>
    </div>
    {% endfor %}
</div>
```

5. 구동

구동



```
C:\Users\mark>cd \flask-space\ch4
```

```
C:\flask-space\ch4>run.py
```



```
$ cd ~
```

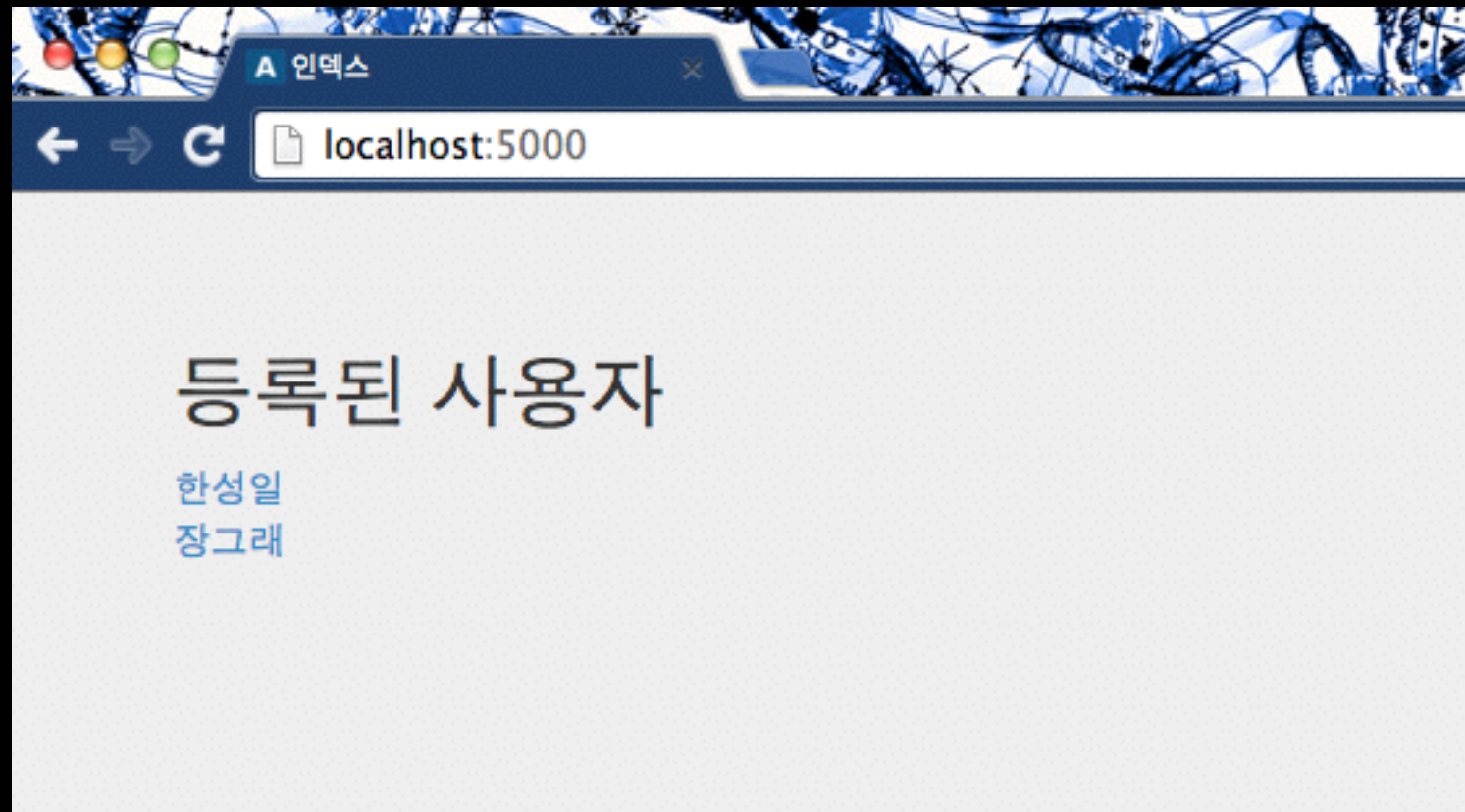
```
$ cd Documents/flask-space/ch4
```

```
$ chmod 755 run.py
```

```
$ ./run.py
```

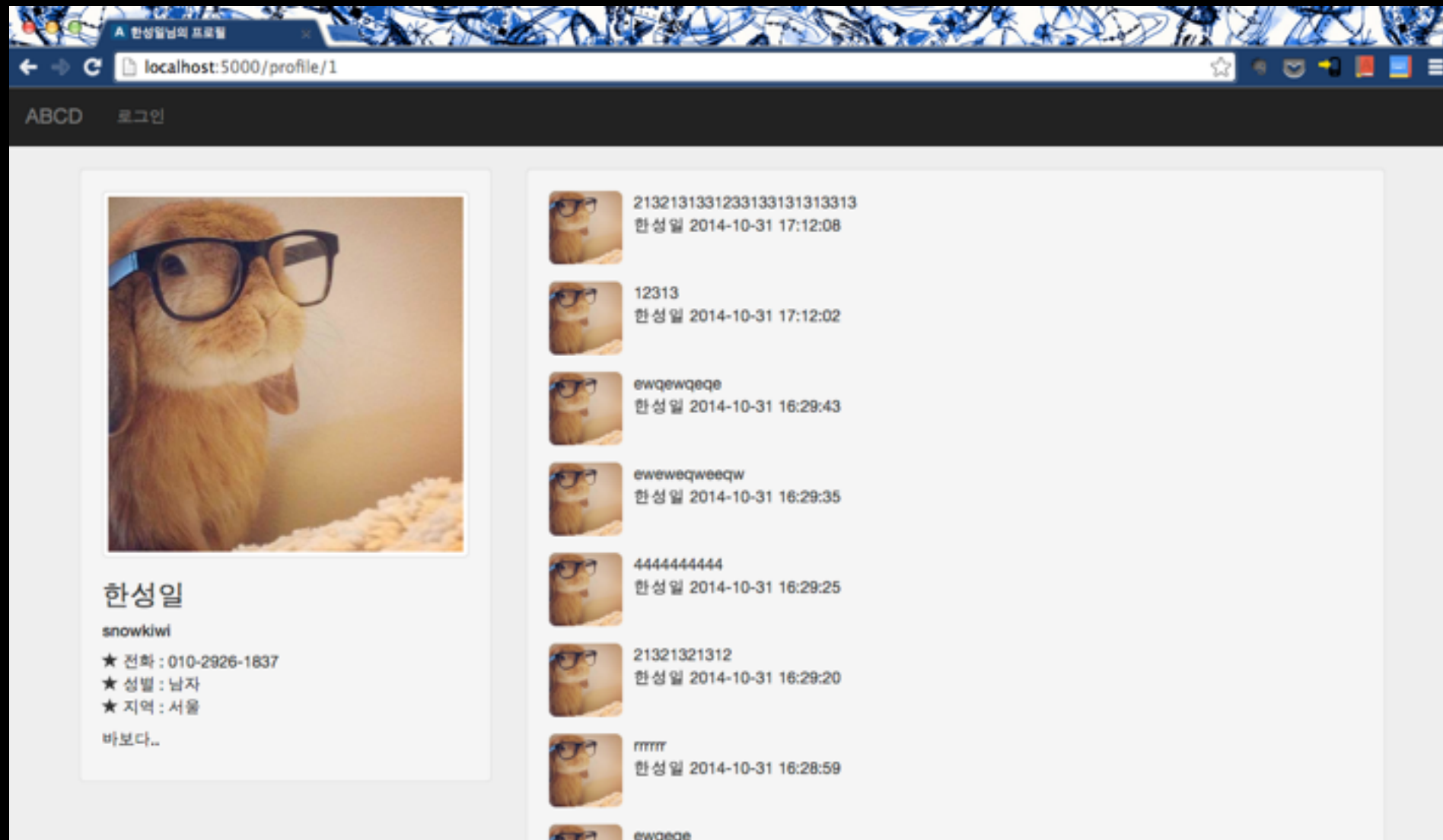
구동 1

<http://localhost:5000/>



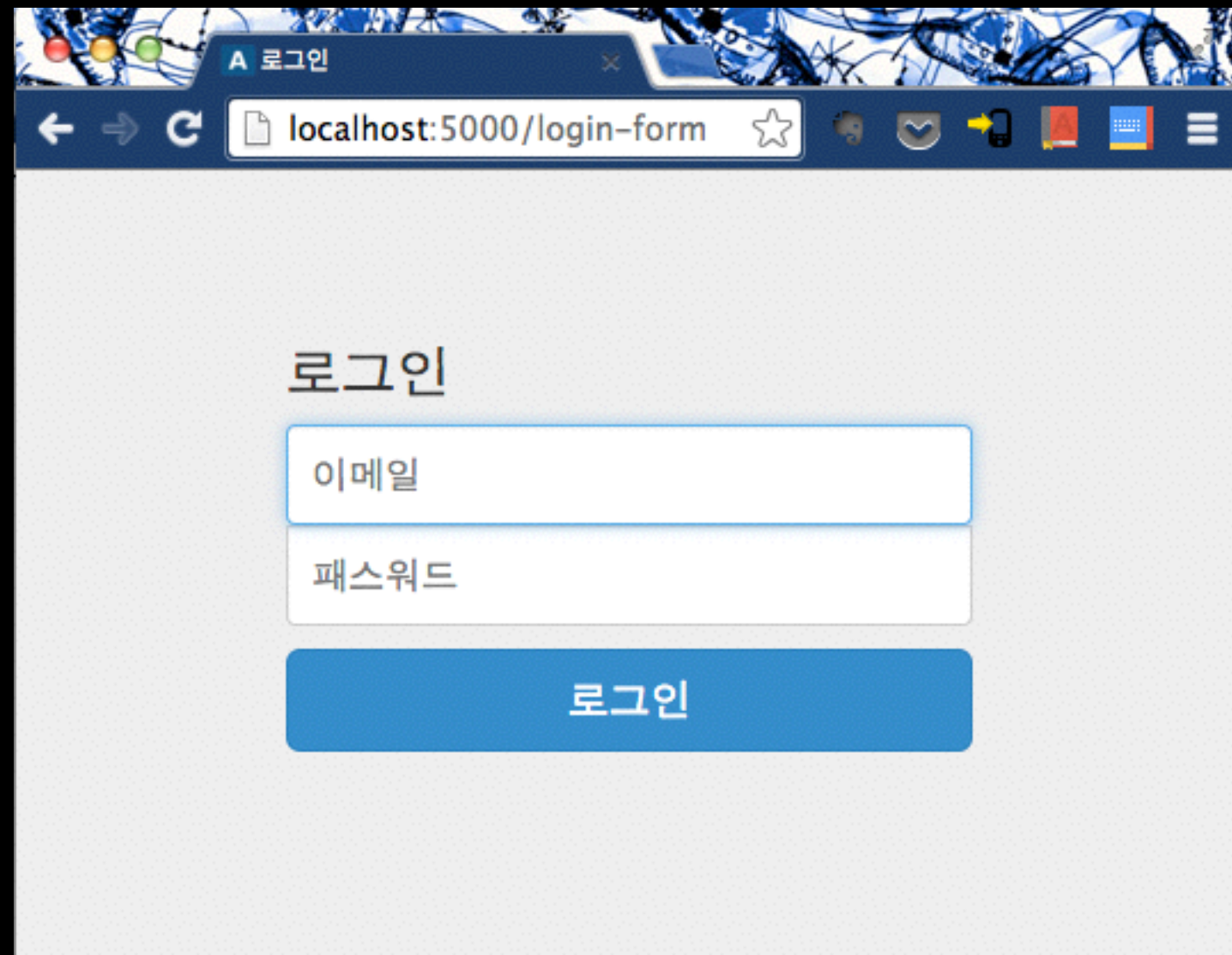
구동 2

<http://localhost:5000/profile/1>



구동 3

<http://localhost:5000/login-form>



A screenshot of a web browser window. The address bar shows the URL `localhost:5000/login-form`. The page title is '로그인' (Login). The main content area contains a login form with the following elements:

- A title '로그인' (Login) in a large, dark font.
- An input field labeled '이메일' (Email) with a light blue border.
- An input field labeled '패스워드' (Password) with a light blue border.
- A blue button labeled '로그인' (Login) in white text.

Q&A

정말!!
수고하셨습니다. :D

- ABCD 그룹

<https://www.facebook.com/groups/562787713823026/>

- 한성일

<https://www.facebook.com/jamie.han.16>
iheart79@gmail.com