파이썬 프로그래밍

(Web Programming)

- **Python으로 웹서버 구축하기**

1) flask로 서버를 만들기 위해서 서버를 관리할 python 파일(app.py)과 아래 폴더를 생성한다.

- **app.py**
- **templates 폴더(html 문서)**
- **static 폴더(css문서, 이미지 등등)**

2) 패키지 설치하기

```
pip install flask
```

3) app.py에 서버 구축하기

```
[data]-[python]-[web]-[ex01] app.py
```

```
from flask import Flask

app = Flask(__name__, template_folder='templates')

@app.route('/hello')
def hello():
    return 'Hello World'

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

4) 터미널로 나와서 아래 명령을 실행한 후 브라우저에서 http://localhost:5000/hello로 접속해 'Hello World'가 출력되는지 확인한다.

```
python app.py
```

5) app.py에 서버를 통해 HTML 문서 실행하기

```
[data]-[python]-[web]-[ex01] app.py
```

```
from flask import Flask, render_template

app = Flask(__name__, template_folder='templates')

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/hello')
def hello():
    return 'Hello World'

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

```
[data]-[python]-[web]-[ex01]-[templates] index.html
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>나의 웹페이지</h1>
</body>
</html>
```

- **홈페이지 Layout 설정하기**

1) header.html 파일을 작성한다.

[data]-[python]-[web]-[ex01]-[templates] header.html

```
<div>
    <img src="http://via.placeholder.com/960x150" width="100%"/>
</div>
```

2) footer.html 파일을 작성한다.

[data]-[python]-[web]-[ex01]-[templates] footer.html

```
<div class="text-center">
    <hr/>
    <h3>Copyright 2024. ICIA 홍길동 All rights reserved.</h3>
</div>
```

3) 글꼴 변경을 위한 style.css 파일을 작성한다.

[data]-[python]-[web]-[ex01]-[static]-[css] style.css

```
@import url('https://fonts.googleapis.com/css2?family=Sunflower:wght@300&display=swap');
* {
    font-family: "Sunflower", sans-serif;
    font-weight: 300;
    font-style: normal;
}
```

4) base.html 파일을 작성한다.

[data]-[python]-[web]-[ex01]-[templates] base.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" ...>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" ...></script>
    <link rel="stylesheet" href="/static/css/style.css"/>
    <title>{{title}}</title>
</head>
<body>
    <div class="container">
        {%include 'header.html'%}
        {%block main_area%}
        {%endblock%}
        {%include 'footer.html'%}
    </div>
</body>
</html>
```

[data]-[python]-[web]-[ex01]-[templates] index.html

```
{%extends 'base.html'%}
{%block main_area%}
  <div class="my-5 text-center">
    <h1>{{title}}</h1>
  </div>
{%endblock%}
```

- **게시판 BluePrint 작성 후 app.py 파일에 등록**

1) menu.html 파일을 작성한다.

[data]-[python]-[web]-[ex01]-[templates] menu.html

```
<div class="mt-5">
    <a href="/" class="me-3">Home</a>
    <a href="/bbs" class="me-3">게시판</a>
    <a href="/user/login" style="float:right;">로그인</a>
    <hr/>
</div>
```

2) base.html 파일에 menu.html 파일을 삽입한다.

[data]-[python]-[web]-[ex01]-[templates] base.html

```
<body>
    <div class="container">
        {%include 'header.html'%}
        {%include 'menu.html'%}
        {%block main_area%}
        {%endblock%}
        {%include 'footer.html'%}
    </div>
</body>
```

3) bbs Bluepirnt 컴포넌트를 작성하고 app.py 파일에 작성한다.

[data]-[python]-[web]-[ex01]-[routes] bbs.py

```
from flask import Blueprint, render_template

bp = Blueprint('bbs', __name__, url_prefix='/bbs')

@bp.route('/')
def list():
   return render_template('bbs/list.html', title='게시판')
```

[data]-[python]-[web]-[ex01] app.py

```
from flask import Flask, render_template
from routes import bbs

app = Flask(__name__, template_folder='templates')
app.register_blueprint(bbs.bp)
...
if __name__ == '__main__':
  app.run(port=5000, debug=True)
```

4) 게시글 목록 출력을 위한 템플릿 파일을 작성한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] list.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
    </div>
{%endblock%}
```

- **게시글 목록 출력 프로그램**

1) 게시판 작성을 위한 사용자, 게시판 테이블을 생성하고 샘플 데이터를 입력한다.

```
create table users(
    uid varchar(20) not null primary key,
    upass varchar(20) not null,
    uname varchar(20) not null
);
```

```
insert into users values('blue', 'pass', '김블루');
insert into users values('green', 'pass', '최그린');
insert into users values('sky', 'pass', '스카이');
insert into users values('red', 'pass', '이레드');
```

```
create table bbs(
    bid int auto_increment primary key,
    title varchar(1000) not null,
    contents text,
    writer varchar(10) not null,
    regDate datetime default now(),
    foreign key(writer) references users(uid)
);
```

```
insert into bbs(title, writer)
    values('필자가 생각하는 플라스크란?','blue');
insert into bbs(title, writer)
    values('플라스크는 마이크로 웹 프레임워크다','sky');
insert into bbs(title, writer)
    values('간결하다는 것은 대체 무슨 뜻일까?','red');
insert into bbs(title, writer)
    values('확장성 있는 설계란 대체 무슨 뜻일까?','green');
insert into bbs(title, writer)
    values('플라스크는 자유로운 프레임워크다','blue');
```

2) 데이터베이스 접속을 위한 컴포넌트를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] config.json

```
{
    "host": "localhost",
    "user": "web",
    "password": "pass",
    "db":"webdb"
}
```

[data]-[python]-[web]-[ex01]-[dao] db.py

```
import pymysql
import json

file =open('./dao/config.json', 'r', encoding='utf-8')
config = json.loads(file.read())

connection = pymysql.connect(
    host=config['host'],
    user=config['user'],
    password=config['password'],
    db=config['db'],
    charset='utf8',
    cursorclass=pymysql.cursors.DictCursor)
```

[data]-[python]-[web]-[ex01]-[dao] bbsDAO.py

```
from dao import db

def list():
    try:
        with db.connection.cursor() as cursor:
        sql = "select *, date_format(regDate, '%Y-%m-%d %T') fmtdate from bbs order by bid desc"
        cursor.execute(sql)
        rows = cursor.fetchall()
        return rows
    except Exception as err:
        print('목록오류:', err)
    finally:
        cursor.close()
```

[data]-[python]-[web]-[ex01]-[templates] base.html

```
<head>
    ...
    <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
    <title>{{title}}</title>
</head>
```

3) 게시글 목록 출력을 위한 템플릿을 작성한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] list.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        <div id="div_list"></div>
        {%raw%}
        <script id="temp_list" type="x-handlebars-template">
            <table class="table">
                {{#each .}}
                    <tr>
                        <td>{{bid}}</td>
                        <td>{{title}}</td>
                        <td>{{writer}}</td>
                        <td>{{fmtdate}}</td>
                    </tr>
                {{/each}}
            </table>
        </script>
        {%endraw%}
    </div>
    <script>
        getList();
        function getList(){
            $.ajax({
                type:"get",
                url:"/bbs/list.json",
                success:function(data){
                    const temp=Handlebars.compile($("#temp_list").html());
                    $("#div_list").html(temp(data));
                }
              })
        }
    </script>
{%endblock%}
```

- **게시글 등록 프로그램**

1) 게시글 등록 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dba] bbsDAO.py

```
from dao import db

def list():
    try:
        with db.connection.cursor() as cursor:
            ...
    finally:
        cursor.close()

def insert(bbs):
    try:
        with db.connection.cursor() as cursor:
            sql ="insert into bbs(title, contents, writer) values(%s,%s,%s)"
            cursor.execute(sql, (bbs.get('title'), bbs.get('contents'), bbs.get('writer')))
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('등록오류:', err)
        return 'fail'
    finally:
        cursor.close()
```

2) 게시글 등록 라우터를 등록한다.

[data]-[python]-[web]-[ex01]-[routes] bbs.py

```
from flask import Blueprint, render_template, request
from dao import bbsDAO
import json

bp = Blueprint('bbs', __name__, url_prefix='/bbs')
...
@bp.route('/insert')
def insert():
    return render_template('bbs/insert.html', title='글쓰기')

@bp.route('/insert', methods=['POST'])
def bbsPost():
    req = json.loads(request.get_data())
    result = bbsDAO.insert(req)
    return result
```

3) 게시글 등록 페이지를 작성한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] list.html

```
...
<div class="my-5 text-center">
    <h1>{{title}}</h1>
    <div class="text-end">
        <a href="/bbs/insert">글쓰기</a>
    </div>
    <hr/>
<div id="div_list"></div>
...
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] insert.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        <div class="row justify-content-center">
            <form name="frm" method="post" class="col-xs-12 col-md-10 col-lg-8">
                <input name="title" placeholder="제목을 입력하세요." class="form-control mb-2"/>
                <textarea name="contents" placeholder="내용을 입력하세요." rows="10" class="form-control"></textarea>
                <div class="text-center my-3">
                    <button class="btn btn-outline-primary">게시글등록</button>
                    <button class="btn btn-outline-secondary" type="reset">등록취소</button>
                </div>
            </form>
        </div>
    </div>
    <script>
        $(frm).on("submit", function(e){
            e.preventDefault();
            const title=$(frm.title).val();
            const contents=$(frm.contents).val();
            if(title == "") {
                alert("제목을 입력하세요.");
                $(frm.title).focus();
                return;
            }
            $.ajax({
                type:"post",
                url:"/bbs/insert",
                data:JSON.stringify({title, contents, writer:'blue'}),
                success:function(data){
                    if(data == 'success'){
                        location.href='/bbs/'
                    }
                }
            });
        });
    </script>
{%endblock%}
```

- **게시글 정보 프로그램**

1) 게시글 정보 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] bbsDAO.py

```
...
def read(bid):
    try:
        with db.connection.cursor() as cursor:
            sql = "select * from bbs where bid=%s"
            cursor.execute(sql, bid)
            row = cursor.fetchone()
            return row
        except Exception as err:
            print('읽기오류:', err)
    finally:
        cursor.close()
```

2) 게시글 정보 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] bbs.py

```
from flask import Blueprint, render_template, request
from dao import bbsDAO
import json

bp = Blueprint('bbs', __name__, url_prefix='/bbs')
...
@bp.route('/<int:bid>')
def read(bid):
    bbs = bbsDAO.read(bid)
    return render_template('bbs/read.html', title='게시글 정보', bbs=bbs)
```

3) 게시글 정보 페이지를 작성한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] list.html

```
...
    <script id="temp_list" type="x-handlebars-template">
        <table class="table">
            {{#each .}}
            <tr>
                <td>{{bid}}</td>
                <td><a href="/bbs/{{bid}}">{{title}}</a></td>
                <td>{{writer}}</td>
                <td>{{fmtdate}}</td>
            </tr>
            {{/each}}
        </table>
    </script>
...
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] read.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        <div class="row justify-content-center">
            <div class="col-xs-12 col-md-10 col-lg-8">
                <div class="text-end">
                    <a href="/bbs/update/{{bbs.bid}}">정보수정</a>
                </div>
                <div class="card text-start">
                    <div class="card-body">
                        <h5>{{bbs.title}}</h5>
                        <hr/>
                        <div style="white-space:pre-wrap">{{bbs.contents}}</div>
                    </div>
                    <div class="text-muted card-footer">
                        Posted on {{bbs.regDate}} by {{bbs.writer}}
                    </div>
                </div>
            </div>
        </div>
    </div>
{%endblock%}
```

- **게시글 수정 프로그램**

1) 게시글 수정를 위한 라이터와 페이지를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] bbs.py

```
...
@bp.route('/update/<int:bid>')
def update(bid):
    bbs = bbsDAO.read(bid)
    return render_template('bbs/update.html', title='게시글수정', bbs=bbs)
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] update.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        <div class="row justify-content-center">
            <form name="frm" method="post" class="col-xs-12 col-md-10 col-lg-8">
                <input name="title" value="{{bbs.title}}" class="form-control mb-2"/>
                <textarea name="contents" rows="10" class="form-control">{{bbs.contents}}</textarea>
                <div class="text-center my-3">
                    <button class="btn btn-outline-primary">게시글수정</button>
                    <button class="btn btn-outline-secondary" type="reset">등록취소</button>
                </div>
            </form>
        </div>
    </div>
{%endblock%}
```

2) 게시글 수정을 위한 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] bbsDAO.py

```
...
def update(bbs):
    try:
        with db.connection.cursor() as cursor:
            sql = "update bbs set title=%s, contents=%s where bid=%s"
            cursor.execute(sql, (bbs.get('title'), bbs.get('contents'), bbs.get('bid')))
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('수정오류:', err)
        return 'fail'
    finally:
       cursor.close()
```

3) 게시글 수정을 위한 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] bbs.py

```
...
@bp.route('/update', methods=['POST'])
def updatePost():
    req = json.loads(request.get_data())
    result = bbsDAO.update(req)
    return result
```

4) 게시글 수정 페이지에 수정을 위한 내용을 추가한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] update.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        ..
    </div>
    <script>
        $(frm).on("submit", function(e){
            e.preventDefault();
            const title=$(frm.title).val();
            const contents=$(frm.contents).val();
            const bid="{{bbs.bid}}";
            if(title == "") {
                alert("제목을 입력하세요.");
                $(frm.title).focus();
                return;
            }
            if(!confirm(`${bid}번 변경된 정보를 수정하실래요?`)) return;
            $.ajax({
                type:"post",
                url:"/bbs/update",
                data:JSON.stringify({title, contents, bid}),
                success:function(data){
                    if(data == 'success'){
                        location.href=`/bbs/${bid}`
                    }
                }
            });
        });
    </script>
{%endblock%}
```

- **게시글 삭제 프로그램**

1) 게시글 삭제를 위한 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] bbsDAO.py

```
...
def delete(bid):
    try:
        with db.connection.cursor() as cursor:
            sql = "delete from bbs where bid=%s"
            cursor.execute(sql, bid)
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('삭제오류:', err)
        return 'fail'
    finally:
        cursor.close()
```

2) 게시글 삭제를 위한 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] bbs.py

```python
...
@bp.route('/delete/<int:bid>', methods=['POST'])
def deletePost(bid):
    result = bbsDAO.delete(bid)
    return result
```

3) 게시글 정보 페이지에 게시글 삭제를 위한 내용을 추가한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] read.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        <div class="row justify-content-center">
            <div class="col-xs-12 col-md-10 col-lg-8">
                <div class="text-end">
                    <a href="/bbs/update/{{bbs.bid}}" class="me-2">수정</a>
                    <a href="#" id="delete">삭제</a>
                </div>
                <div class="card text-start">
                    <div class="card-body">
                        <h5>{{bbs.title}}</h5>
                        <hr/>
                        <div style="white-space:pre-wrap">{{bbs.contents}}</div>
                    </div>
                    <div class="text-muted card-footer">
                        Posted on {{bbs.regDate}} by {{bbs.writer}}
                    </div>
                </div>
            </div>
        </div>
    </div>
    <script>
        const bid="{{bbs.bid}}";
        $("#delete").on("click", function(e){
            e.preventDefault();
           if(!confirm(`${bid}번 게시글을 삭제하실래요?`)) return;
           $.ajax({
                type:"post",
                url:`/bbs/delete/${bid}`,
                success:function(data){
                    if(data == 'success'){
                        location.href='/bbs/';
                    }
                }
            });
        });
    </script>
{%endblock%}
```

- **게시글 목록 페이징**

1) 페이징을 위한 게시글 목록 라우터를 수정한다.

[data]-[python]-[web]-[ex01]-[routes] bbs.py

```
from flask import Blueprint, render_template, request
from dao import bbsDAO
import json

bp = Blueprint('bbs', __name__, url_prefix='/bbs')

@bp.route('/')
def list():
  return render_template('bbs/list.html', title='게시판')
...

@bp.route('/list.json')
def listJson():
    args = request.args
    row = bbsDAO.total()
    rows = bbsDAO.list(args)
    data = {'total':row.get('total'), 'list':rows}
    return data
```

2) 페이징 처리를 위한 게시글 목록 DAO를 수정하고 게시글 수를 구하는 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] bbsDAO.py

```
...
def list(args):
    page = int(args['page'])
    size = int(args['size'])
    start = (page - 1) * size
    try:
        with db.connection.cursor() as cursor:
            sql = "select *, date_format(regDate, '%%Y-%%m-%%d %%T') as fmtdate from bbs order by bid desc limit %s, %s"
            cursor.execute(sql, (start, size))
            rows = cursor.fetchall()
            return rows
    except Exception as err:
        print('목록오류:', err)
    finally:
        cursor.close()

def total():
    try:
        with db.connection.cursor() as cursor:
            sql = "select count(*) total from bbs"
            cursor.execute(sql)
            return cursor.fetchone()
    except Exception as err:
        print('Total오류', err)
    finally:
        cursor.close()
```

3) base.html에 페이징을 위한 CDN URL를 추가하고 목록 페이지를 수정한다.

[data]-[python]-[web]-[ex01]-[templates] base.html

```
<head>
    ...
    <script src="https://cdnjs.cloudflare.com/ajax/libs/twbs-pagination/1.4.2/jquery.twbsPagination.min.js"></script>
</head>
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] list.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        ...
        <div id="div_list"></div>
        <div id="pagination" class="pagination justify-content-center mt-5"></div>
        {%raw%}
        <script id="temp_list" type="x-handlebars-template">
            <table class="table">
                {{#each list}}
                    ...
                {{/each}}
              </table>
        </script>
        {%endraw%}
    </div>
    <script>
        let page=1;
        let size=5;
        let totalPages=1;
        getList();
        function getList(){
            $.ajax({
                type:"get",
                url:`/bbs/list.json?page=${page}&size=${size}`,
                success:function(data){
                    const temp=Handlebars.compile($("#temp_list").html());
                    $("#div_list").html(temp(data));
                    //전체 댓글수가 사이즈 보다 작은 경우에는 페이지네이션을 숨긴다.
                    if(data.total > size) $("#pagination").show();
                    else $("#pagination").hide();
                    let curTotalPages=data.total == 0 ? 1: Math.ceil(data.total/size);
                    if(totalPages != curTotalPages) {
                        totalPages=curTotalPages;
                        $("#pagination").twbsPagination("changeTotalPages", totalPages, page);
                    }
                }
            });
        }
        $('#pagination').twbsPagination({
            totalPages:totalPages,
            visiblePages: 5,
            startPage : 1,
            initiateStartPageClick: false,
            first:'<<',
            prev :'<',
            next :'>',
            last :'>>',
            onPageClick: function (event, clickedPage) {
                if(page != clickedPage) {
                    page=clickedPage;
                    getList();
                }
            }
        });
</script>
{%endblock%}
```

- **로그인/로그아웃 프로그램**

1) 로그인 작업을 위한 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] userDAO.py

```
from dao import db

def read(uid):
    try:
        with db.connection.cursor() as cursor:
            sql = "select * from users where uid=%s"
            cursor.execute(sql, uid)
            row = cursor.fetchone()
            return row
    except Exception as err:
        print('로그인오류:', err)
    finally:
        cursor.close()
```

2) 사용자 라우터를 생성한 후 app.py에 등록한다.

[data]-[python]-[web]-[ex01]-[routes] user.py

```
from flask import Blueprint, render_template, request
import json
from dao import userDAO

bp = Blueprint('user', __name__, url_prefix='/user')

@bp.route('/login')
def login():
    return render_template('user/login.html', title='로그인')
```

[data]-[python]-[web]-[ex01] app.py

```
from flask import Flask, render_template
from routes import bbs, user

app = Flask(__name__, template_folder='templates')
app.register_blueprint(bbs.bp)
app.register_blueprint(user.bp)
...
```

[data]-[python]-[web]-[ex01]-[template]-[user] login.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        <div class="row justify-content-center px-5">
            <div class="col-md-6 col-lg-5 card">
                <form class="card-body" name="frm">
                    <input name="uid" class="form-control mb-2" placeholder="아이디" value="red">
                    <input name="upass" type="password" class="form-control mb-2" placeholder="비밀번호" value="pass">
                    <button class="btn btn-primary w-100">로그인</button>
                </form>
            </div>
        </div>
    </div>
{%endblock%}
```

3) 로그인 체크를 위한 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] user.py

```
from flask import Blueprint, render_template, request
import json
from dao import userDAO

bp = Blueprint('user', __name__, url_prefix='/user', template_folder='../templates/user')

@bp.route('/login')
def login():
    return render_template('login.html', title='로그인')

@bp.route('/login', methods=['POST'])
def loginPost():
    req = json.loads(request.get_data())
    uid = req.get('uid')
    upass = req.get('upass')
    user = userDAO.read(uid)
    result = 0
    if user:
        if user.get('upass') == upass:
            result = 1
        else:
            result = 2
    return str(result)
```

[data]-[python]-[web]-[ex01]-[templates]-[user] login.html

```
{%extends 'base.html'%}
{%block main_area%}
  <div class="my-5 text-center">
      ...
  </div>
  <script>
      $(frm).on("submit", function(e){
          e.preventDefault();
          const uid = $(frm.uid).val();
          const upass = $(frm.upass).val();
          if(uid == '' || upass== ''){
              alert('아이디 또는 비밀번호를 입력하세요.');
              return;
          }
          $.ajax({
              type:"post",
              url:"/user/login",
              data:JSON.stringify({uid, upass}),
              success:function(data){
                  if(data == 0){
                      alert("아이디가 존재하지 않습니다.");
                  }else if(data == 2){
                      alert("비밀번호가 일치하지 않습니다.");
                  }else if(data == 1){
                      sessionStorage.setItem("uid", uid)
                      location.href="/";
                  }
              }
          });
      });
  </script>
{%endblock%}
```

4) 로그인/로그아웃을 위한 menu.html 파일을 수정한다.

[data]-[python]-[web]-[ex01]-[templates] menu.html

```
<div class="mt-5">
    <a href="/" class="me-3">Home</a>
    <a href="/bbs" class="me-3">게시판</a>
    <a href="/user/login" id="login" style="float:right;">로그인</a>
    <span id="span_logout" style="float:right;">
        <a href="#" id="mypage" class="me-2">마이페이지</a>
        <a href="#" id="logout">로그아웃</a>
    </span>
    <hr/>
</div>
<script>
    const uid=sessionStorage.getItem("uid");
    if(uid){
        $("#span_logout").show();
        $("#login").hide();
        $("#mypage").html(`${uid}님`)
    }else {
       $("#span_logout").hide();
       $("#login").show();
    }

    $("#logout").on("click", function(e){
        e.preventDefault();
        if(!confirm("정말로 로그아웃하실래요?")) return;
        sessionStorage.clear();
        location.href="/";
    });
</script>
```

- **회원가입 프로그램**

1) 회원가입을 위한 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] user.py

```
from flask import Blueprint, render_template, request
import json
from dao import userDAO

bp = Blueprint('user', __name__, url_prefix='/user')

@bp.route('/login')
def login():
  return render_template('user/login.html', title='로그인')
...
@bp.route('/insert')
def insert():
    return render_template('user/insert.html', title='회원가입')

@bp.route('/insert', methods=['POST'])
def insertPost():
    req = json.loads(request.get_data())
    result = userDAO.insert(req)
    return result
```

2) 회원가입을 위한 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] userDAO.py

```
...
def insert(user):
    try:
        with db.connection.cursor() as cursor:
            sql = "insert into users(uid, upass, uname) values(%s, %s, %s)"
            cursor.execute(sql, (user.get('uid'), user.get('upass'), user.get('uname')))
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('회원가입:', err)
        return 'fail'
    finally:
        cursor.close()
```

3) 회원가입을 위한 html 파일을 작성한다.

[data]-[python]-[web]-[ex01]-[templates]-[user] insert.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        <div class="row justify-content-center px-5">
            <div class="col-md-6 col-lg-5 card">
                <form class="card-body" name="frm">
                    <input name="uid" class="form-control mb-2" placeholder="아이디" value="red">
                    <input name="upass" type="password" class="form-control mb-2" placeholder="비밀번호" value="pass">
                    <input name="uname" class="form-control mb-2" placeholder="이름">
                    <button class="btn btn-primary w-100">회원가입</button>
                </form>
            </div>
        </div>
    </div>
    <script>
        $(frm).on("submit", function(e){
            e.preventDefault();
            const uid = $(frm.uid).val();
            const upass = $(frm.upass).val();
            const uname = $(frm.uname).val();
            if(uid == '' || upass == '' || uname == ''){
                alert("회원 등록을 위한 모든 정보를 입력하세요.");
                return;
            }
            if(!confirm("정말로 회원가입을 하실래요?")) return;
            $.ajax({
                type:"post",
                url:"/user/insert",
                data:JSON.stringify({uid, upass, uname}),
                success:function(data){
                    if(data == 'success'){
                        location.href='/user/login';
                    }
                }
            });
        });
    </script>
{%endblock%}
```

4) 로그인 상태에 따른 프로그램 기능 변경

[data]-[python]-[web]-[ex01]-[templates]-[bbs] list.html

```
<a href="/bbs/insert" id="insert">글쓰기</a>
...
<script>
    $("#insert").on("click", function(e){
        e.preventDefault();
        const target = $(this).attr("href");
        if(uid){
            location.href=target;
        }else{
            sessionStorage.setItem("target", target);
            location.href='/user/login';
        }
    });
</script>
```

[data]-[python]-[web]-[ex01]-[templates]-[user] login.html

```
$.ajax({
    ...
    success:function(data){
          ...
        }else if(data == 1){
            sessionStorage.setItem("uid", uid)
            const target = sessionStorage.getItem("target");
            if(target){
              location.href=target;
            }else{
              location.href="/";
            }
        }
    }
});
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] read.html

```
<div class="text-end" id="div_update">
    <a href="/bbs/update/{{bbs.bid}}" class="me-2">수정</a>
    <a href="#" id="delete">삭제</a>
</div>
...
<script>
    const writer = "{{bbs.writer}}";
    if(uid == writer){
        $("#div_update").show();
    }else{
        $("#div_update").hide();
    }
</script>
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] insert.html

```
$.ajax({
    type:"post",
    url:"/bbs/insert",
    data:JSON.stringify({title, contents, writer:uid}),
    success:function(data){
      if(data == 'success'){
            location.href='/bbs/'
        }
    }
});
```

- **마이페이지 프로그램**

1) 사용자 테이블에 아래와 같이 칼럼들을 추가한다.

```
alter table users add column photo varchar(200);
alter table users add column phone varchar(20);
alter table users add column address1 varchar(1000);
alter table users add column address2 varchar(1000);
```

2) 사용자 정보 출력을 위한 라우터와 페이지를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] user.py

```
@bp.route('/read/<uid>')
def read(uid):
    user = userDAO.read(uid)
    for key, value in user.items():
        if not value:
            user[key] = ''

    return render_template('user/read.html', title='마이페이지', user=user)
```

[data]-[python]-[web]-[ex01]-[templates]-[user] read.html

```
{%extends 'base.html'%}
{%block main_area%}
    <div class="my-5 text-center">
        <h1>{{title}}</h1>
        <div class="row px-5">
            <div class="col card">
                <form name="frm" class="card-body">
                    <div class="input-group mb-2">
                        <div class="input-group-text px-5">이름</div>
                        <input value="{{user.uname}}" name="uname" class="form-control">
                    </div>
                    <div class="input-group mb-2">
                        <div class="input-group-text px-5">전화</div>
                        <input value="{{user.phone}}" name="phone" class="form-control">
                    </div>
                    <div class="input-group mb-2">
                        <div class="input-group-text px-5">주소</div>
                        <input value="{{user.address1}}" name="address1" class="form-control">
                        <button type="button" class="btn btn-primary" id="btn_search">주소검색</button>
                    </div>
                    <div class="input-group mb-2">
                        <input value="{{user.address2}}" name="address2" class="form-control" placeholder="상세주소">
                    </div>
                    <div class="mt-3">
                        <button class="btn btn-outline-success">정보수정</button>
                        <button type="reset" class="btn btn-outline-secondary">수정취소</button>
                    </div>
                </form>
            </div>
        </div>
    </div>
{%endblock%}
```

[data]-[python]-[web]-[ex01]-[templates] base.html

```
<head>
   ...
   <script src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
</head>
```

2) 사용자 정보수정을 위한 DAO와 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] userDAO.py

```
...
def update(user):
    try:
        with db.connection.cursor() as cursor:
            sql = "update users set uname=%s,phone=%s,address1=%s,address2=%s where uid=%s"
            cursor.execute(sql,(user.get('uname'),user.get('phone'),user.get('address1'),user.get('address2'),user.get('uid')))
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('정보수정:', err)
        return 'fail'
    finally:
        cursor.close()
```

[data]-[python]-[web]-[ex01]-[routes] user.py

```
...
@bp.route('/update', methods=['POST'])
def update():
    req = json.loads(request.get_data())
    result = userDAO.update(req)
    return result
```

3) 사용자 정보수정을 위한 read.html 페이지에 아래 내용을 추가한다.

[data]-[python]-[web]-[ex01]-[templates]-[user] read.html

```
<script>
    $(frm).on("submit", function(e){
        e.preventDefault();
        if(!confirm("변경된 정보를 수정하실래요?")) return;
        const uname=$(frm.uname).val();
        const phone=$(frm.phone).val();
        const address1=$(frm.address1).val();
        const address2=$(frm.address2).val();
        $.ajax({
            type:"post",
            url:"/user/update",
            data:JSON.stringify({uid, uname, phone, address1, address2}),
            success:function(data){
                if(data == 'success'){
                    alert("정보가 변경되었습니다.");
                    location.href=`/user/read/${uid}`;
                }
            }
        });
    });

    $("#btn_search").on("click", function(){
        new daum.Postcode({
            oncomplete: function(data){
                const address = data.buildingName ? `${data.address}(${data.buildingName})`:data.address;
                $(frm.address1).val(address);
            }
        }).open();
    });
</script>
```

- **비밀번호 변경 프로그램**

1) 비밀번호 변경을 위한 모달창을 작성한다.

[data]-[python]-[web]-[ex01]-[templates]-[user] passModal.html

```
<div style="top:30%"
    class="modal fade" id="modal_password" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1 “
    aria-labelledby="staticBackdropLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="staticBackdropLabel">비밀번호변경</h1>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <form name="frm1">
                    <div class="input-group mb-2">
                        <div class="input-group-text title">현재비밀번호</div>
                        <input class="form-control" name="upass" type="password">
                    </div>
                    <div class="input-group mb-2">
                        <div class="input-group-text title">새 비밀번호</div>
                        <input class="form-control" name="npass" type="password">
                    </div>
                    <div class="input-group mb-2">
                        <div class="input-group-text title">비밀번호확인</div>
                        <input class="form-control" name="cpass" type="password">
                    </div>
                </form>
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">닫기</button>
                <button type="button" class="btn btn-primary" id="btn_save">저장</button>
            </div>
        </div>
    </div>
</div>
```

[data]-[python]-[web]-[ex01]-[templates]-[user] read.html

```
...
<form name="frm" class="card-body">
    <div class="input-group mb-2">
        <div class="input-group-text px-5">이름</div>
        <input value="{{user.uname}}" name="uname" class="form-control">
        <button class="btn btn-primary" id="btn_password" type="button">비밀번호</button>
    </div>
    ...
</form>
{%include 'user/passModal.html'%}
...
<script>
    $("#btn_password").on("click", function(){
        $(frm1.upass).val("");
        $(frm1.npass).val("");
        $(frm1.cpass).val("");
        $("#modal_password").modal("show");
    });
</script>
```

---

2) 비밀번호 변경을 위한 DAO와 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] userDAO.py

```
def updatePass(user):
    try:
        with db.connection.cursor() as cursor:
            sql = "update users set upass=%s where uid=%s"
            cursor.execute(sql, (user.get('upass'), user.get('uid')))
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('비밀번호수정:', err)
        return 'fail'
    finally:
        cursor.close()
```

[data]-[python]-[web]-[ex01]-[routes] user.py

```
@bp.route('/update/pass', methods=['POST'])
def updatePass():
    req = json.loads(request.get_data())
    result = userDAO.updatePass(req)
    return result
```

[data]-[python]-[web]-[ex01]-[templates]-[user] passModa.html

```
<script>
    $("#btn_save").on("click", function(){
        const upass=$(frm1.upass).val();
        const npass=$(frm1.npass).val();
        const cpass=$(frm1.cpass).val();
        if(upass == '' || npass == '' || cpass == ''){
            alert("비밀번호들을 모두 입력하세요!"); return;
        }
        if(npass != cpass){
            alert("새 비밀번호와 비밀번호 확인이 일치하지 않습니다."); return;
        }
        $.ajax({
            type:"post",
            url:"/user/login",
            data:JSON.stringify({uid, upass}),
            success:function(data){
                if(data == 2){
                    alert("현재 비밀번호가 일치하지 않습니다.");
                }else if(data == 1){
                    if(upass == npass) {
                        alert("새로운 비밀번호가 현재 비밀번호와 일치합니다.") return;
                    }
                    if(!confirm("새로운 비밀번호로 변경하실래요?")) return;
                    $.ajax({
                        type:"post",
                        url:"/user/update/pass",
                        data:JSON.stringify({uid, upass:npass}),
                        success:function(data){
                            if(data == 'success'){
                                alert("비밀번호가 변경되었습니다.");
                                sessionStorage.clear();
                                location.href="/user/login";
                            }
                        }
                    });
                }
            }
        });
    });
</script>
```

- **사용자 사진 등록 프로그램**

1) 사용자 시진 변경을 위한 모달창을 작성한다.

[data]-[python]-[web]-[ex01]-[templates]-[user] photoModal.html

```
<div style="top:10%"
    class="modal fade" id="modal_photo" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1"
    aria-labelledby="staticBackdropLabel" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="staticBackdropLabel">사진변경</h1>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body px-5">
                <img id="photo" src="http://via.placeholder.com/100x100" width="100%" class="mb-3">
                <input id="file" type="file" class="form-control">
            </div>
            <div class="modal-footer">
                <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">닫기</button>
                <button type="button" class="btn btn-primary" id="btn_save_photo">저장</button>
            </div>
        </div>
    </div>
</div>
<script>
    $("#file").on("change", function(e){
        const file=e.target.files[0];
        $("#photo").attr("src", URL.createObjectURL(file));
    });
</script>
```

[data]-[python]-[web]-[ex01]-[templates]-[user] read.html

```
<style>
    .photo {
        width: 50px;
        border-radius: 50%;
        cursor: pointer;
    }
</style>
<div class="my-5 text-center">
    ...
    <h1>
        <img src="http://via.placeholder.com/50x50" class="photo" id="img_photo">
        {{title}}
    </h1>
    ...
    <form name="frm" class="card-body">
        ...
    </form>
    {%include 'user/passModal.html'%}
    {%include 'user/photoModal.html'%}
</div>
</script>
    $("#img_photo").on("click", function(){
        $("#photo").attr("src", "http://via.placeholder.com/100x100");
        $("#file").val("");
        $("#modal_photo").modal("show");
    })
</script>
```

---

2) 사용자 사진 업로드를 위한 라우터와 사진명 수정을 위한 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] user.py

```
from flask import Blueprint, render_template, request
import json
from dao import userDAO
import os

bp = Blueprint('user', __name__, url_prefix='/user')
...
@bp.route('/update/photo', methods=['POST'])
def updatePhoto():
    try:
        file = request.files['file']
        uid = request.form['uid']

        projectPath = os.getcwd()
        uploadPath = '/static/upload/'
        fileName= uid + ".jpg"
        uploadFile = projectPath + uploadPath + fileName

        if os.path.exists(uploadFile):
            os.remove(uploadFile)
        file.save(uploadFile)

        user={'uid':uid, 'photo':uploadPath + fileName}
        userDAO.updatePhoto(user)
        return 'success'
    except Exception as err:
        print('파일업로드:', err)
        return 'fail'
```

[data]-[python]-[web]-[ex01]-[dao] userDAO.py

```
from dao import db

def read(uid):
    try:
        with db.connection.cursor() as cursor:
            sql = "select * from users where uid=%s"
            cursor.execute(sql, uid)
            row = cursor.fetchone()
            return row
    except Exception as err:
        print('로그인오류:', err)
    finally:
        cursor.close()
...
def updatePhoto(user):
    try:
        with db.connection.cursor() as cursor:
            sql = "update users set photo=%s where uid=%s"
            cursor.execute(sql, (user.get('photo'), user.get('uid')))
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('사진수정:', err)
        return 'fail'
    finally:
        cursor.close()
```

3) 사용자 사진 변경을 위한 모달 html 파일에 아래 내용을 추가한다.

[data]-[python]-[web]-[ex01]-[templates]-[user] photoModal.html

```
<script>
    ...
    $("#btn_save_photo").on("click", function(){
        const file=$("#file")[0].files[0];

        if($("#file").val() == "") {
            alert("변경할 사진을 선택하세요.");
            return;
        }
        if(!confirm("변경된 사진을 저장하실래요?")) return;

        const formData = new FormData();
        formData.append("uid", sessionStorage.getItem("uid"));
        formData.append("file", file);

        $.ajax({
            type:"post",
            url:"/user/update/photo",
            data:formData,
            processData:false,
            contentType:false,
            success:function(data){
                if(data == 'success'){
                    alert("사진이 변경되었습니다.");
                    location.href=`/user/read/${sessionStorage.getItem('uid')}`
                }
            }
        });
    });
</script>
```

4) 사진 출력을 위한 read.html 파일을 수정한다.

[data]-[python]-[web]-[ex01]-[templates]-[user] read.html

```
...
<div class="my-5 text-center">
    <h1>
        {%if user.photo%}
            <img src="{{user.photo}}" class="photo" id="img_photo">
        {%else%}
            <img src="http://via.placeholder.com/50x50" class="photo" id="img_photo">
        {%endif%}
        {{title}}
    </h1>
</div>
<script>
    ...
    $("#img_photo").on("click", function(){
        const photo="{{user.photo}}";
        if(photo){
            $("#photo").attr("src", photo);
        }else{
            $("#photo").attr("src", "http://via.placeholder.com/100x100");
        }
        $("#file").val("");
        $("#modal_photo").modal("show");
    })
</script>
```

- **댓글 등록 프로그램**

1) 댓글 프로그램을 위한 테이블을 생성한다.

```
create table reply(
    rid int auto_increment primary key,
    bid int not null,
    writer varchar(20) not null,
    contents text not null,
    regDate datetime default now(),
    foreign key(bid) references bbs(bid)
);
```

2) 댓글 등록을 위한 페이지를 작성한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] reply.html

```
<div class="my-5">
    <div id="login_after">
        <textarea id="contents" class="form-control" rows="5" placeholder="내용을 입력하세요."></textarea>
        <div class="text-end mt-2">
            <button class="btn btn-outline-primary px-5" id="btn_insert">글쓰기</button>
        </div>
    </div>
    <div id="login_before" class="text-end">
        <button class="btn btn-outline-primary px-5" id="btn_write">댓글작성</button>
    </div>
</div>
<script>
    $("#btn_write").on("click", function(){
        sessionStorage.setItem("target", `/bbs/${bid}`);
        location.href="/user/login";
    });

    if(uid){
        $("#login_after").show();
        $("#login_before").hide();
    }else{
        $("#login_after").hide();
        $("#login_before").show();
    }
</script>
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] read.html

```
{%extends 'base.html'%}
{%block main_area%}
<div class="my-5 text-center">
    <h1>{{title}}</h1>
    <div class="row justify-content-center">
        ...
    </div>
    {%include 'bbs/reply.html'%}
</div>
<script>
    const bid = "{{bbs.bid}}";
    ...
</script>
{%endblock%}
```

3) 댓글 등록을 위한 DAO와 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] replyDAO

```
from dao import db

def insert(reply):
    try:
        with db.connection.cursor() as cursor:
            sql = "inesrt into reply(bid, writer, contents) values(%s, %s, %s)"
            cursor.execute(sql, (reply.get('bid'), reply.get('writer'), reply.get('contents')))
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('등록오류:', err)
        return 'fail'
    finally:
        cursor.close()
```

[data]-[python]-[web]-[ex01]-[routes] reply.py

```
from flask import Blueprint, request
import json
from dao import replyDAO

bp = Blueprint('reply', __name__, url_prefix='/reply')

@bp.route('/insert', methods=['POST'])
def insert():
    reply = json.loads(request.get_data())
    result = replyDAO.insert(reply)
    return result
```

[data]-[python]-[web]-[ex01] app.py

```
from flask import Flask, render_template
from routes import bbs, user, reply
...
app.register_blueprint(reply.bp)
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] reply.html

```
<script>
    ...
    $("#btn_insert").on("click", function(){
        const contents=$("#contents").val();
        if(contents==""){
            alert("댓글 내용을 입력하세요!");
            $("#contents").focus();
            return;
        }
        $.ajax({
            type:"post",
            url:"/reply/insert",
            data:JSON.stringify({bid, writer:uid, contents}),
            success:function(data){
                if(data == 'success') {
                    $("#contents").val("");
                }
            }
        });
    });
</script>
```

- **댓글 목록 출력 프로그램**

1) 댓글 목록 출력을 위한 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] replyDAO.py

```
def list(args, bid):
    page = int(args['page'])
    size = int(args['size'])
    start = (page - 1) * size
    try:
        with db.connection.cursor() as cursor:
            sql = "select *, date_format(regDate,'%%Y-%%m-%%d %%T') as fmtdate \
                from reply \
                where bid=%s \
                order by rid desc \
                limit %s, %s"
            cursor.execute(sql, (bid, start, size))
            rows = cursor.fetchall()
            return rows
    except Exception as err:
        print('목록오류:', err)
    finally:
        cursor.close()

def total(bid):
    try:
        with db.connection.cursor() as cursor:
            sql = "select count(*) total from reply where bid=%s"
            cursor.execute(sql, bid)
            row = cursor.fetchone()
            return row
    except Exception as err:
        print('댓글수오류', err)
    finally:
     cursor.close()
```

2) 댓글 목록 출력을 위한 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] reply.py

```
...
@bp.route('/list.json/<int:bid>')
def list(bid):
    args = request.args
    rows =replyDAO.list(args, bid)
    row = replyDAO.total(bid)
    data = {'total':row.get('total'), 'list':rows}
    return data
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] read.html

```
...
<div class="card text-start">
    <div class="card-body">
        <h5>
            <span id="bid">{{bbs.bid}}</span> : {{bbs.title}}
        </h5>
    </div>
</div>
```

3) 댓글 목록 출력 기능을 추가한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] reply.html

```
<style>
    .div_contents {
        white-space: pre-wrap;
        cursor: pointer;
    }
</style>

<div class="my-5">
    <div id="login_after">
        <textarea id="contents" class="form-control" rows="5" placeholder="내용을 입력하세요."></textarea>
        <div class="text-end mt-2">
            <button class="btn btn-outline-primary px-5" id="btn_insert">글쓰기</button>
        </div>
    </div>
    <div id="login_before" class="text-end">
        <button class="btn btn-outline-primary px-5" id="btn_write">댓글작성</button>
    </div>
    <div class="my-5">
        <div id="div_list"></div>
    </div>

    {%raw%}
    <script id="temp_list" type="x-handlebars-template">
        {{#each list}}
            <div class="reply">
                <div class="row">
                    <div class="col my-3 text-start text-muted" style="font-size:12px">
                        <span>{{writer}}</span> | <span>{{fmtdate}}</span>
                    </div>
                    <div class="group1 col text-end" style="{{display writer}}">
                        <button class="update btn btn-outline-success btn-sm">수정</button>
                        <button class="delete btn btn-outline-danger btn-sm" rid="{{rid}}">삭제</button>
                    </div>
                    <div class="group2 col text-end" style="display:none;">
                        <button class="save btn btn-outline-primary btn-sm">저장</button>
                        <button class="cancel btn btn-outline-secondary btn-sm">취소</button>
                    </div>
                </div>
                <div class="div_contents text-start ellipsis">{{contents}}</div>
                <div class="txt_contents" style="display:none">
                     <textarea class="contents form-control" rows="5">{{contents}}</textarea>
                </div>
                <hr>
            </div>
        {{/each}}
    </script>
    <script>
        Handlebars.registerHelper("display", function(writer){
            if(uid!=writer) return "display:none"
        });
    </script>
    {%endraw%}
</div>
```

[data]-[python]-[web]-[ex01]-[templates]-[bbs] reply.html

```
<script>
    let page=1;
    let size=3;
    const bbs_id=$("#bid").html();
    ...
    $("#btn_insert").on("click", function(){
        ...
        $.ajax({
            ...
            success:function(data){
                $("#contents").val("");
                page=1;
                getList();
            }
        });
    });
    ...
    getList();
    function getList(){
        $.ajax({
            type:"get",
            url:`/reply/list.json/${bbs_id}`,
            data:{size, page},
            success:function(data){
                const temp=Handlebars.compile($("#temp_list").html());
                $("#div_list").html(temp(data));
            }
        });
    }

    //각행의 내용을 클릭한 경우
    $("#div_list").on("click", ".div_contents", function(){
        $(this).toggleClass("ellipsis");
    });

    //각행의 수정버튼을 클릭한 경우
    $("#div_list").on("click", ".update", function(){
        const reply=$(this).parent().parent().parent();
        reply.find(".group1").hide();
        reply.find(".group2").show();
        reply.find(".txt_contents").show();
        reply.find(".div_contents").hide();
    });

    //각행의 취소버튼을 클릭한 경우
    $("#div_list").on("click", ".cancel", function(){
        const reply=$(this).parent().parent().parent();
        reply.find(".group1").show();
        reply.find(".group2").hide();
        reply.find(".txt_contents").hide();
        reply.find(".div_contents").show();
    });

    //각행의 삭제버튼을 클릭한 경우
    $("#div_list").on("click", ".delete", function(){
        const rid=$(this).attr("rid");
        if(!confirm(`${rid}번 댓글을 삭제하실래요?`)) return;
    });
</script>
```

- **댓글 삭제 수정하기**

1) 댓글 삭제와 수정을 위한 DAO를 작성한다.

[data]-[python]-[web]-[ex01]-[dao] replyDAO.py

```
from dao import db

def insert(reply):
    ...
def list(args, bid):
    ...
def total(bid):
    ...

def delete(rid):
    try:
        with db.connection.cursor() as cursor:
            sql = "delete from reply where rid=%s"
            cursor.execute(sql, rid)
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('댓글삭제:', err)
        return 'fail'
    finally:
        cursor.close()

def update(reply):
    try:
        with db.connection.cursor() as cursor:
            sql = "update reply set contents=%s where rid=%s"
            cursor.execute(sql, (reply.get('contents'), reply.get('rid')))
            db.connection.commit()
            return 'success'
    except Exception as err:
        print('댓글수정:', err)
        return 'fail'
    finally:
        cursor.close()
```

2) 댓글 삭제와 수정을 위한 라우터를 작성한다.

[data]-[python]-[web]-[ex01]-[routes] reply.py

```
from flask import Blueprint, request
import json
from dao import replyDAO

bp = Blueprint('reply', __name__, url_prefix='/reply')
...

@bp.route('/delete/<int:rid>', methods=['POST'])
def delete(rid):
    result = replyDAO.delete(rid)
    return result

@bp.route('/update', methods=['POST'])
def update():
    reply = json.loads(request.get_data())
    result = replyDAO.update(reply)
    return result
```

3) 댓글 삭제 수정 기능을 reply 페이지에 추가한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] replay.html

```
<script id="temp_list" type="x-handlebars-template">
    {{#each list}}
        ...
        <div class="group1 col text-end" style="{{display writer}}">
          <button class="update btn btn-outline-success btn-sm">수정</button>
          <button class="delete btn btn-outline-danger btn-sm"  rid={{rid}}>삭제</button>
        </div>
        <div class="group2 col text-end" style="display:none;">
          <button class="save btn btn-outline-primary btn-sm" rid="{{rid}}">저장</button>
          <button class="cancel btn btn-outline-secondary btn-sm">취소</button>
        </div>
      </div>
      <div class="div_contents text-start ellipsis">{{contents}}</div>
      <div class="txt_contents" style="display:none">
        <textarea class="contents form-control" rows="5">{{contents}}</textarea>
      </div>
      ...
    {{/each}}
</script>
<script>
    ...
    //각행의 삭제버튼을 클릭한 경우
    $("#div_list").on("click", ".delete", function(){
        const rid=$(this).attr("rid");
        if(!confirm(`${rid}번 댓글을 삭제하실래요?`)) return;
        $.ajax({
            type:"post",
            url:`/reply/delete/${rid}`,
            success:function(data){
                if(data=='success'){
                    getList();
                }
            }
        });
    });

    //각행의 수정버튼을 클릭한 경우
    $("#div_list").on("click", ".save", function(){
        const rid=$(this).attr("rid");
        const reply = $(this).parent().parent().parent();
        const contents = reply.find(".contents").val();
        if(contents == ""){
            alert("댓글 내용을 입력하세요.");
            return;
        }
        if(!confirm(`${rid}번 댓글 내용을 수정하실래요?`)) return;
        $.ajax({
            type:"post",
            url:"/reply/update",
            data:JSON.stringify({rid, contents}),
            success:function(data){
                if(data == 'success'){
                    getList();
                }
            }
        });
    });
</script>
```

4) 댓글 목록 페이징 기능 추가 한다.

[data]-[python]-[web]-[ex01]-[templates]-[bbs] replay.html

```
{%extends 'base.html'%}
{%block main_area%}
...
<div class="my-5">
    <div id="div_list"></div>
    <div id="pagination" class="pagination justify-content-center mt-5"></div>
</div>
...
<script>
    let size=3;
    let page=1;
    const bbs_id=$("#bid").html();
    let totalPages = 1;
    ....
    function getList(){
        $.ajax({
            type:"get",
            url:`/reply/list.json/${bbs_id}`,
            data:{size, page},
            success:function(data){
                const temp=Handlebars.compile($("#temp_list").html());
                $("#div_list").html(temp(data));
                //전체 댓글수가 사이즈보다 작은 경우에는 페이지네이션을 숨긴다.
                if(data.total > size) $("#pagination").show();
                else $("#pagination").hide();
                let curTotalPages=data.total == 0 ? 1: Math.ceil(data.total/size);
                if(totalPages != curTotalPages) {
                    totalPages = curTotalPages;
                    //현재 페이지가 전체 페이지수 보다 큰 경우 페이지를 1감소한다.
                    if(page > totalPages) {
                        page--;
                        getList();
                    }
                    $("#pagination").twbsPagination("changeTotalPages", totalPages, page);
                }
            }
        });
    }

    $('#pagination').twbsPagination({
        totalPages:totalPages,
        visiblePages: 5,
        startPage : 1,
        initiateStartPageClick: false,
        first:'<<',
        prev :'<',
        next :'>',
        last :'>>',
        onPageClick: function (event, clickedPage) {
            if(page != clickedPage) {
                page=clickedPage;
                getList();
            }
        }
    });
</script>
{%endblock%}
```

- **파일첨부 예제**

[templates] fileUpload.html

```
<html>
<div>
    <h1>회원가입</h1>
    <form name='frm' enctype='multipart/form-data' method='post'>
        <input name='uid'>
        <input name='uname'>
        <input name='file' type='file>
        <button>등록</button>
    </form>
</div>
<script>
    $(frm).on('submit', function(e){
        e.preventDefault();
        const uid = $(frm.uid).val()
        const uname = $(frm.uname).val()
        const file = $(frm.file)[0].files[0];
        const formData = new FormData();
        formData.append('file', file);
        formData.append('uid', uid);
        formData.append('uname', uname);
        $.ajax({
            type:'post',
            url:'/users/insert',
            data:formData,
            processData:false,
            contentType:false,
            success:function() {
              alert('첨부파일 업로드가 성공되었습니다.');
            }
        });
    });
</script>
</html>
```

---

app.py

```
from flask import Flask, render_template

app = Flask(__name__, template_folder='templates')

@app.route('/upload')
def upload():
    return render_template('fileUpload.html', title='파일업로드')

@bp.route('/upload', methods=['POST'])
def uploadPost():
    files=request.files.getlist('file')
    for file in files:
        file.save('static/images/' + file.filename)
    form=request.form
    print(form.get('uid'), form.get('uname'))
    return redirect('/')

if__name__ == '__main__':
    app.run(port=5000, debug=True)
```
