

가천대학교 2024학년도 2학기 과목명 : 웹데이터베이스(61355002)		중간고사	교수 : 왕보현
학번		이름	

1. 웹 환경에서 client(브라우저의 웹페이지)가 웹 서버에 정보를 보내는 방법 세 가지를 쓰시오.(3점)
get, post, Ajax

2. 현재 실습 시간에 작성하는 home.ejs와 topic.js 그리고 db.js를 3 tier 구조 측면에서 설명하시오. (3점)

*Model: db.js
 View: home.ejs
 Controller: topic.js*

*home.ejs 는 프레젠테이션 계층
 topic.js 는 애플리케이션 계층
 db.js 는 데이터 계층*

3. 현재 실습 시간에 수행하는 프로그램 환경과 같은 환경에서 (b)의 ejs를 이용하여 (a)의 main.js를 실행시키면 (c)의 실행 결과를 얻을 수 있다. ① ~ ⑤까지 빈 칸에 알맞은 코드를 작성하시오. 5, 3과 같은 상수를 쓰지 말고 변수를 최대한 활용하고 ejs에 context의 모든 변수를 반드시 활용하시오.(5점)

(a) main.js

```
app.get('/', (req,res)=>{
  var a = 5;
  var b = 3;

  var context = {name : "input ",
    grammer: `<input type="text" name = "aaa">`,
    grammer2 : `<input type="text">`,
    answer : ①
  };

  res.render('homeW3-1',context,(err,html)=>{
    res.end(html)
  });
});
```

(c) 실행결과

← → ↻ ① 127.0.0.1:3000

HTML 태그 문법을 배워 봅시다.

input 태그 문법은 다음과 같습니다.

<input type="text" name = "aaa">

위의 input 태그를 실행한 결과 입니다.

(b) homeW3-1.ejs

```
<h4>HTML 태그 문법을 배워 봅시다. </h4>
<p>②</p>
<pre>
  ③
</pre>
<p>위의 input 태그를 실행한 결과 입니다. </p>
<pre>
  ④
</pre>
<p>answer : ⑤</p>
```

answer : 5 + 3의 결과는 8 입니다.

*① : a+b
 ② : <% = name %>
 ③ : <% = grammer %>
 ④ : <% = grammer2 %>
 ⑤ : <% = answer %>*

가천대학교 2024학년도 2학기 과목명 : 웹데이터베이스(61355002) 중간고사 교수 : 왕보현			
학번		이름	

4. (a)는 home.ejs 코드 일부이다. (a) home.ejs

home.ejs를 사용하는 main.js의 코드 일부가 (b)와 같을 때 (a)의 form문의 제출 버튼 클릭 요청을 처리하기 위해 (b)의 중간생략 부분에 추가할 코드를 작성하시오. (a)의 요청을 처리할 모듈은 topic.home(req,res)이다. /another/5는 semantic url이며 5를 저장하는 변수는 num이다. 5를 읽어 콘솔에 출력하는 코드도 포함하시오. (3점)

```
<form action="/another/5" method="post">
  <div>제목 : <input type="text" name="title"> </div>
  <div>날짜 : <input type="text" name="date" len="8"></div>
  <div><input type="submit" value="제출"></div>
</form>
```

(b) main.ejs

```
const express=require('express');
const app = express();

app.set('views',__dirname + '/views');
app.set('view engine','ejs')

topic = require("../lib/topicT5")

중간생략

app.get('/favicon.ico', (req,res)=>res.w
app.listen(3000, ()=>console.log('Example
```

*app.post('/another/:num', (req,res) => {
 const num = req.params.num;
 console.log(num);
 topic.home(req,res);
});*

5. (b)와 같은 이름과 값을 갖고 브라우저를 닫아도 3일간 유지되는 영구 쿠키를 설정하는 코드를 (a)의 빈칸에 쓰시오.(3점)

(a) 127.0.0.1:3000/요청을 처리하는 메소드 일부

```
res.writeHead(200, {
  'Set-Cookie': ['yummy_cookie=choco',
                 'tasty_cookie=strawberry',
                 ]
});
```

(b)

Name	Value
Permanent	cookies

*Permanent = cookies; Max-Age = 60*60*24*3*

6. 요청된 id와 같은 값을 갖는 하나의 레코드를 topic 테이블로부터 검색하는 sql 문이다. 검색된 컬럼들(title, descript, dated)을 nodejs 변수 t, d, dt에 각각 저장하는 코드를 작성하시오. (3점)

```
db.query('select title, descript, dated from topic where id = ${id}',(error2, topic)=>{
```

var t = topic.title;

*var d = topic.descript;
var dt = topic.dated;*

7. 다음 코드를 실행하고 브라우저에서 "127.0.0.1:3000/"을 첫번째 요청 했을 때와 두 번째 요청 했을 때 콘솔 창에 뜨는 결과를 작성하시오. 단, 마지막 줄의 Cookie Test는 고려하지 마세요.

```
const express = require('express') ;
const app = express() ;
var cookie = require('cookie');
app.get('/',function(req, res){
  res.writeHead(200, {
    'Set-Cookie': ['cookie1=seoul', 'cookie2=seongnam']
  });
  console.log(req.headers.cookie)
  if(req.headers.cookie !== undefined){
    var cookies = cookie.parse(req.headers.cookie)
    console.log(cookies)
  }
  res.end('Cookie!!!');
});
app.listen(3000, () => console.log('Cookie Test'))
```

1) 첫번째 요청했을 때 콘솔 결과 (1점)

undefined

2) 두번째 요청했을 때 콘솔 결과 (2점)

*cookie = seoul ; cookie2 = seongnam
{ cookie: 'seoul', cookie2: 'seongnam' }*