

[경기도형 대학생 취업브리지 사업]

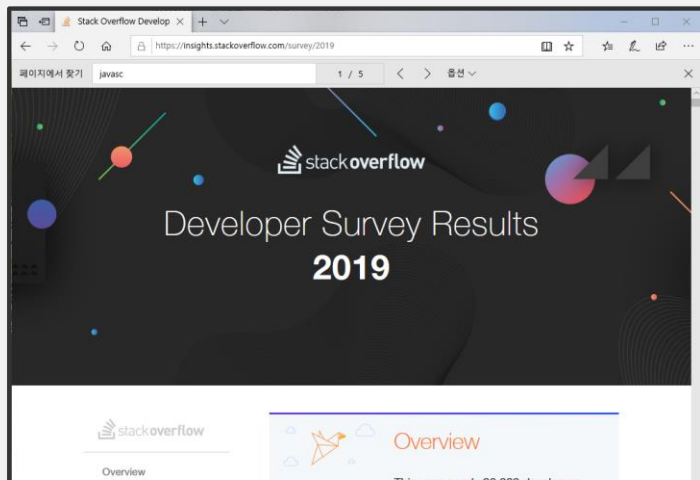
웹 모바일및 빅데이터 응용과정

1주차: 주차별 강좌 제목

Node.JS기반 백엔드 데모 서비스 개발

기술

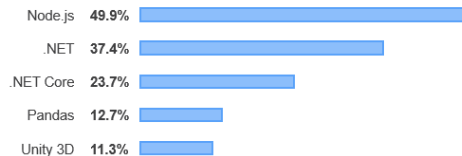
<https://insights.stackoverflow.com/survey/2019>



Other Frameworks, Libraries, and Tools

All Respondents

Professional Developers



Programming, Scripting, and Markup Languages

All Respondents

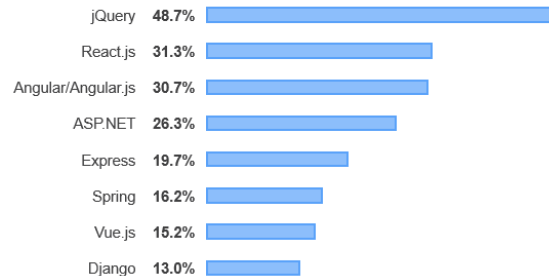
Professional Developers



Web Frameworks

All Respondents

Professional Developers



이번강좌에서는

Node.JS 기반 백엔드 서비스개발은 쇼핑몰 판매 서비스를 주제로 웹사이트를 개발하는 과정을 프로젝트 형식으로 살펴보며, 각 단계마다 필요한 기술과 해결을 위한 접근방식을 소개합니다.

특히 이 강의를 통해 **Node.JS, express** 웹프레임워크, **ejs** 미들웨어, **Bootstrap 4.x** 기술을 활용하는 기술을 배우며, 소스코드 템플릿과 샘플을 직접 다루어 보다 빠른 개발능력을 키우는데 혁신적인 학습방법을 제공합니다.



■ 목차

- 01. 서비스 소개(쇼핑몰 웹사이트 구축)
- 02. 개발환경 구축 (Node.js/Express, Bootstrap, Atom)
- 03. 개발사례 분석(샘플기능/코드)
- 04. 프로젝트 미션(서비스 완성)

■ 서비스 소개

■ 목차

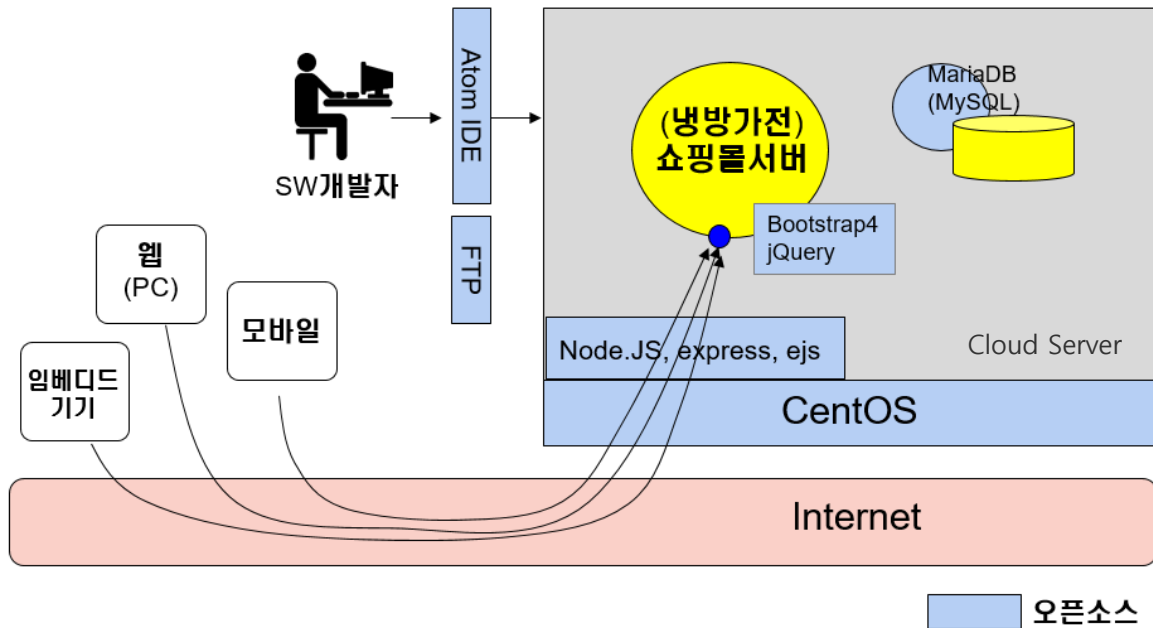
■ 01. 서비스 소개(쇼핑몰 웹사이트 구축)

■ 02. 개발환경 구축 (Node.js/Express, Bootstrap, Atom)

내용	상세내용
데모서비스 설계	<ul style="list-style-type: none"> - 데모서비스(쇼핑몰) 설계방법 (화면, 기능, DB, REST API) - MVC모델 개발방법 (디렉터리구조, Naming)
개발환경구축 (오픈소스)	<ul style="list-style-type: none"> - <u>백엔드</u> 개발환경: Node.js, express, <u>ejs</u>모듈 등 - <u>프론트엔드</u> 개발환경: Bootstrap4.3, <u>jQuery</u> - 개발 IDE: Atom - (자체)클라우드서버, DB: <u>MariaDB(MySQL)</u>
데모서비스 구현사례	<ul style="list-style-type: none"> - 쇼핑몰 서비스 코드 다운로드 / 코드설치 - 쇼핑몰 서비스 코드분석 (쇼핑몰) <ul style="list-style-type: none"> > 사용자: 메인화면, 회원가입, 로그인/로그아웃 > 관리자: 로그인/로그아웃, (관리자전용) 메뉴
데모서비스 추가개발 (수행과제)	<ul style="list-style-type: none"> - 쇼핑몰 서비스 추가기능 개발과제 수행 <ul style="list-style-type: none"> 추가1) 사용자: 회원가입수정(필드추가: 핸드폰, 주소, 포인트) 추가2) 관리자: 사용자관리(회원리스트, 정보수정, <u>회원삭제</u>) 추가3) 관리자: 제품등록관리(제품등록, 제품변경, 삭제) 추가4) 사용자: 제품검색("선풍기"검색, "<u>에어콘</u>"검색)

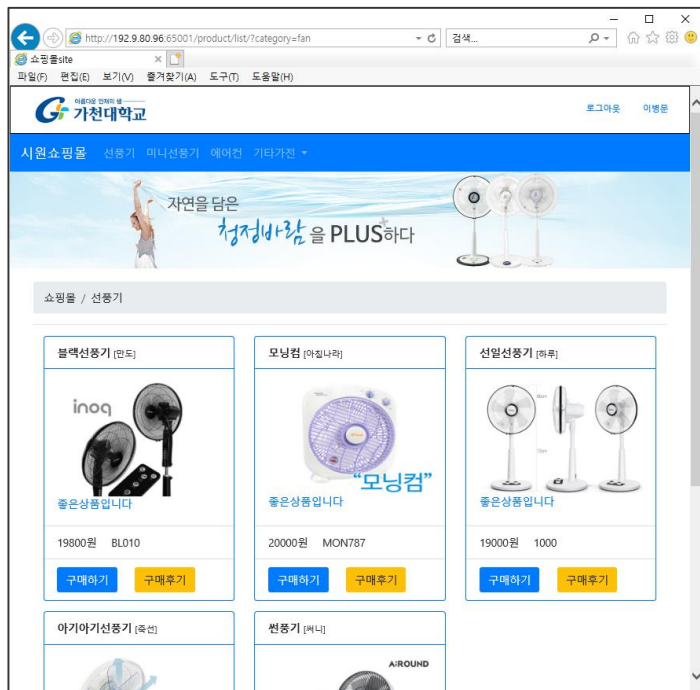
■ 서비스 개요

- 오픈소스 기반의 (데모용 냉방가전) 쇼핑몰 서비스 개발

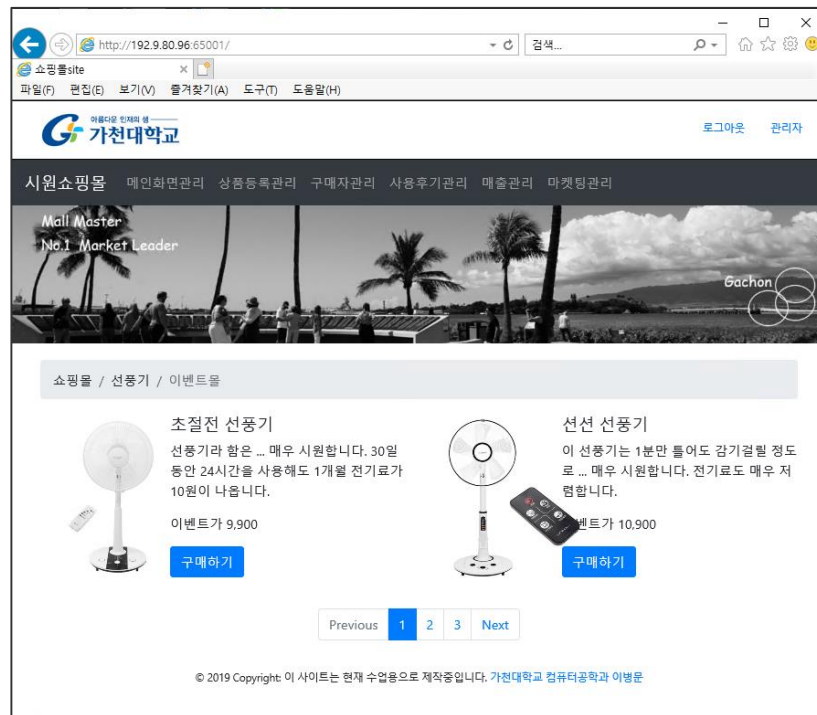


■ 서비스 개발 프로세스

- 오픈소스 기반의 (데모용 냉방가전) 쇼핑몰 서비스 개발

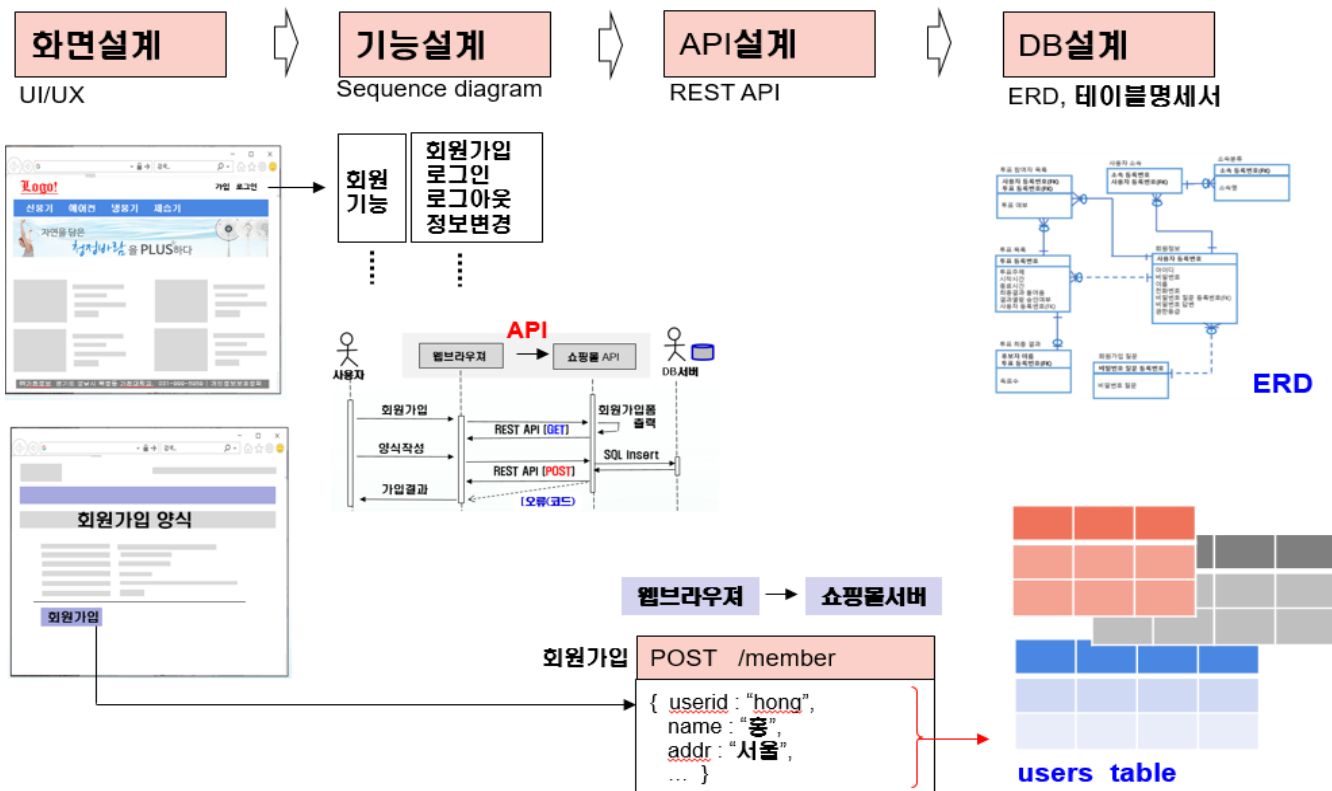


구매자서비스

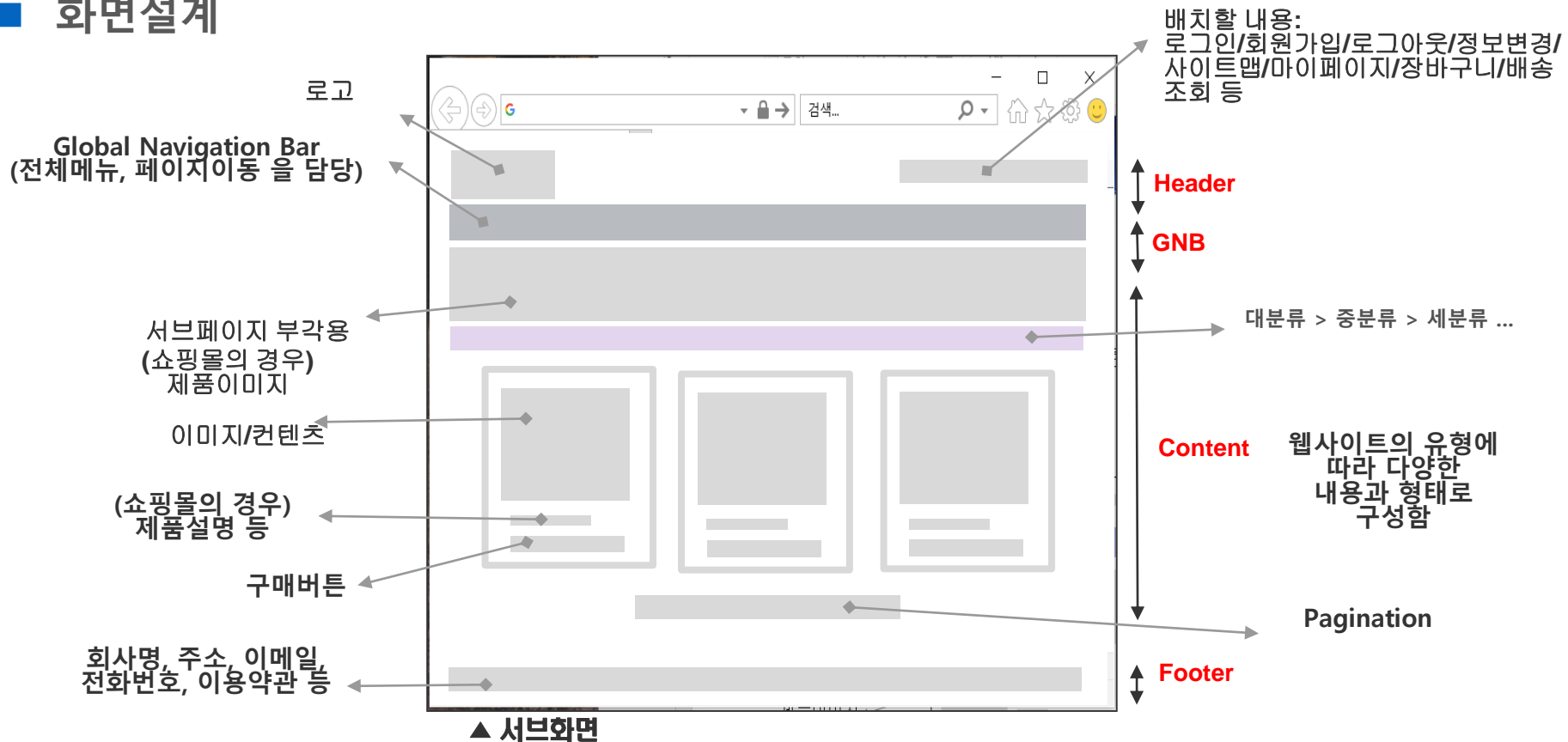


관리자서비스

■ 서비스 설계

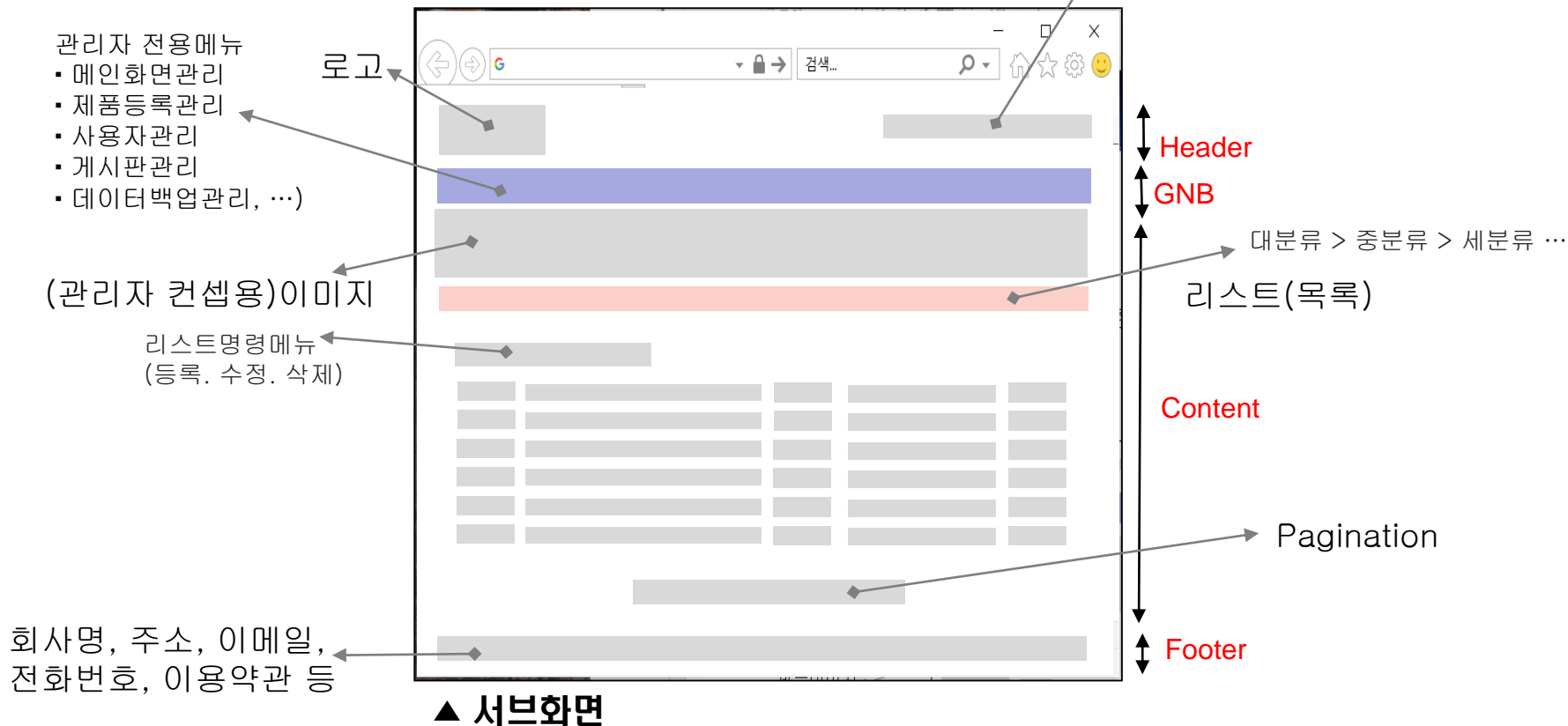


■ 화면설계



■ 화면설계

로그아웃 / 암호변경/마이페이지 등



■ 기능설계

- 사용자별 기능을 정의한다 (아래예시)

○ 사용자 기능

역할이 서로 다른 사용자
(예, 구매자 vs 판매자, ...)

주요기능	요구기능	사용자1	사용자2
메인 화면	헤더	O	O
	GNB	O	O
	Content	O	O
	Footer	O	O
회원 기능	회원가입	X	O
	로그인	X	O
	로그아웃	X	O
	정보변경	O	X

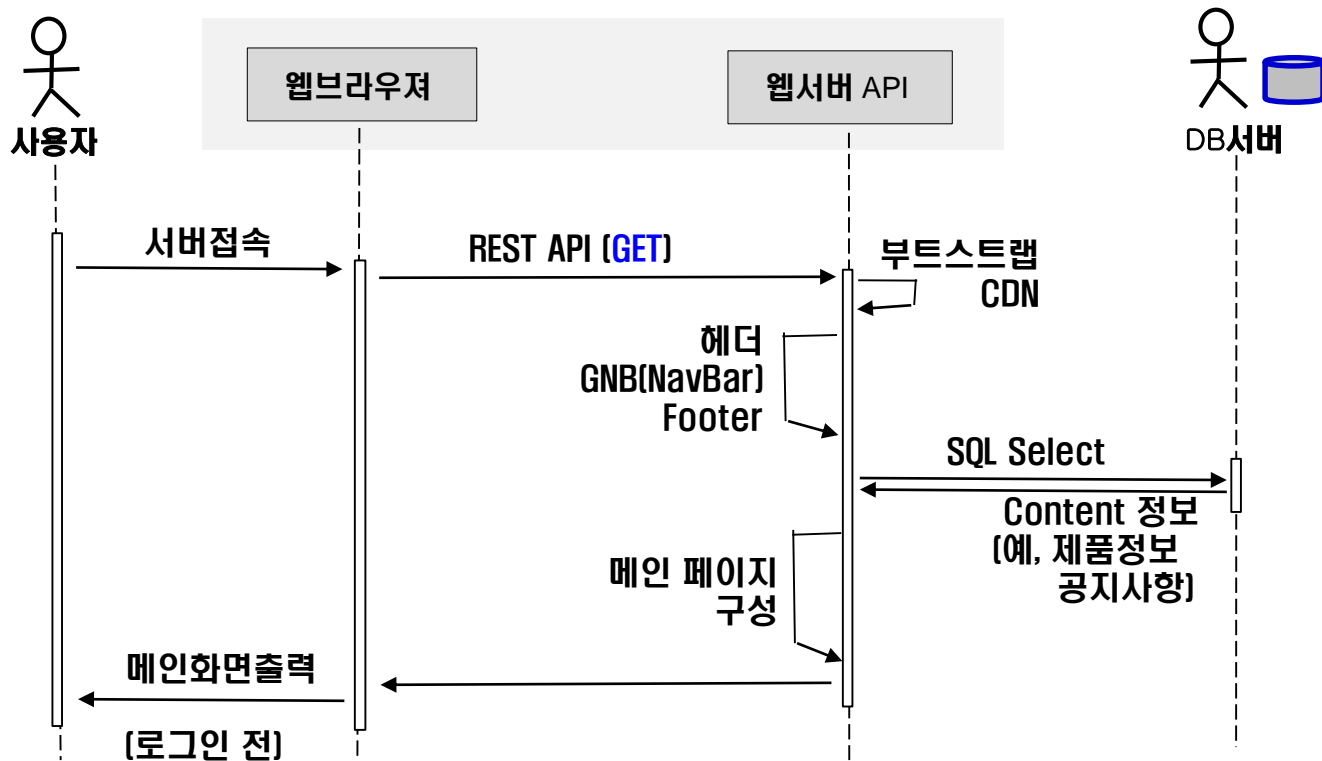
○ 관리자 기능

주요기능	요구기능
메인 화면	헤더
	GNB
	Content
	Footer
회원 관리	회원리스트
	회원검색
	회원보기
	회원삭제

같은 방식으로 모든기능을 정의한다

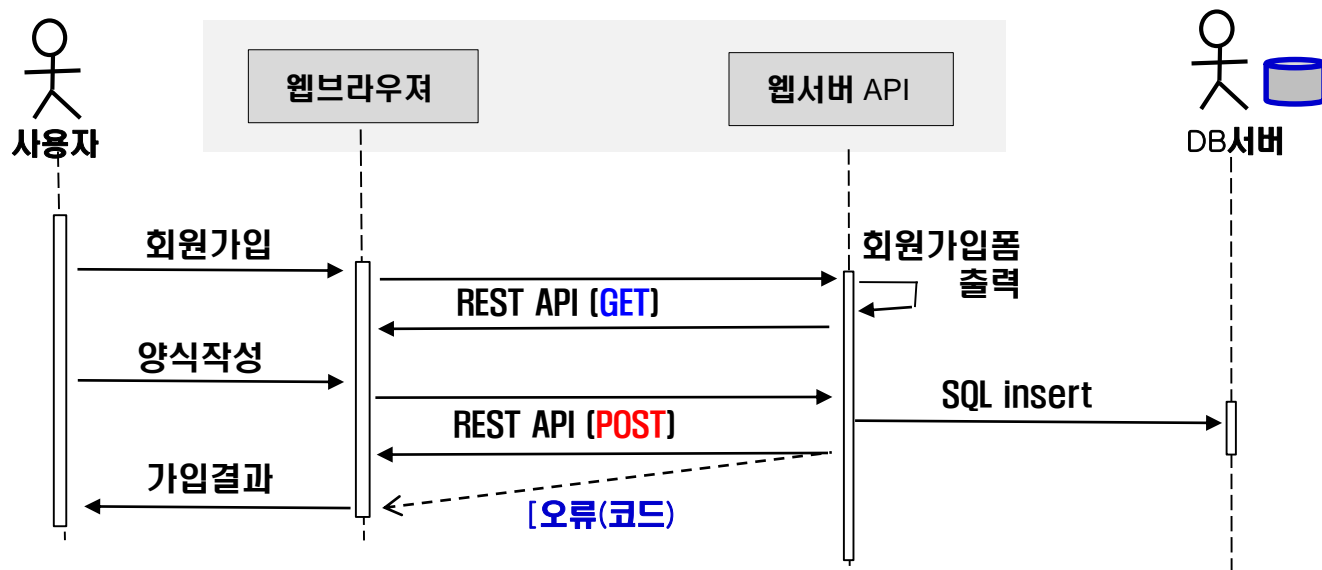
■ 기능설계

■ 메인화면접속



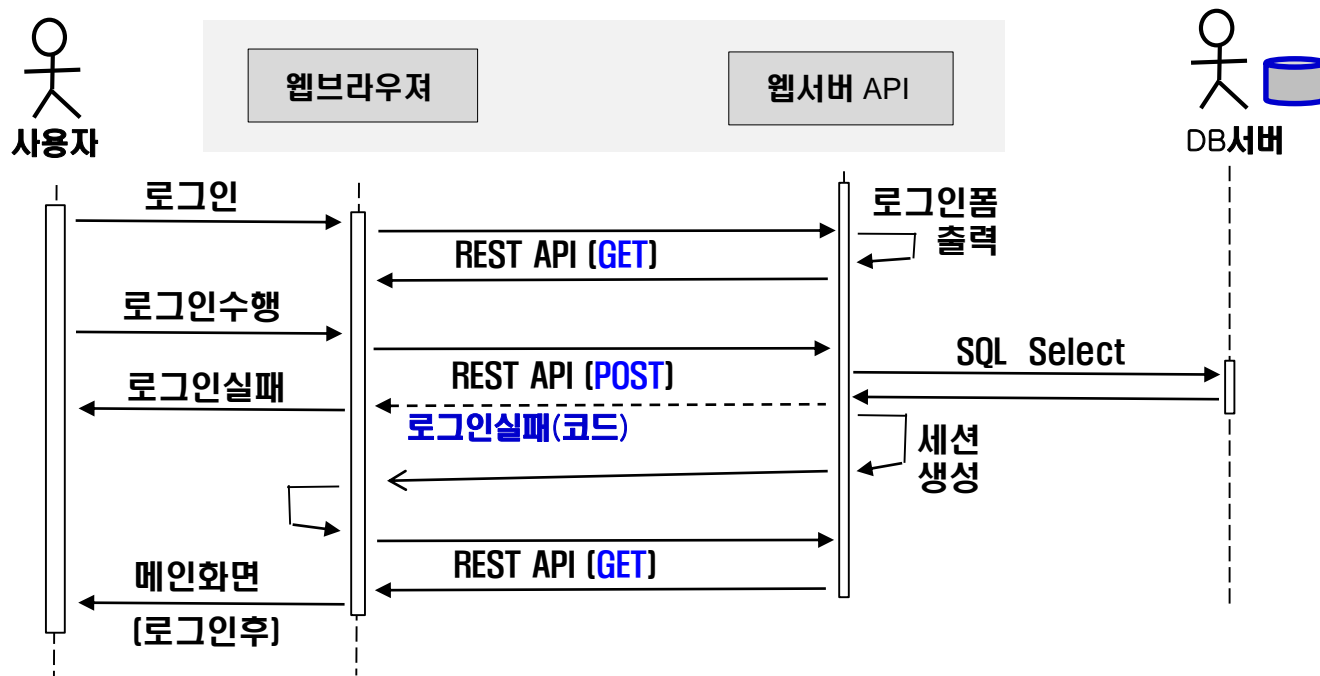
■ 기능설계

- 회원기능 (회원가입기능) (사용자)



■ 기능설계

- 회원기능 (로그인기능)

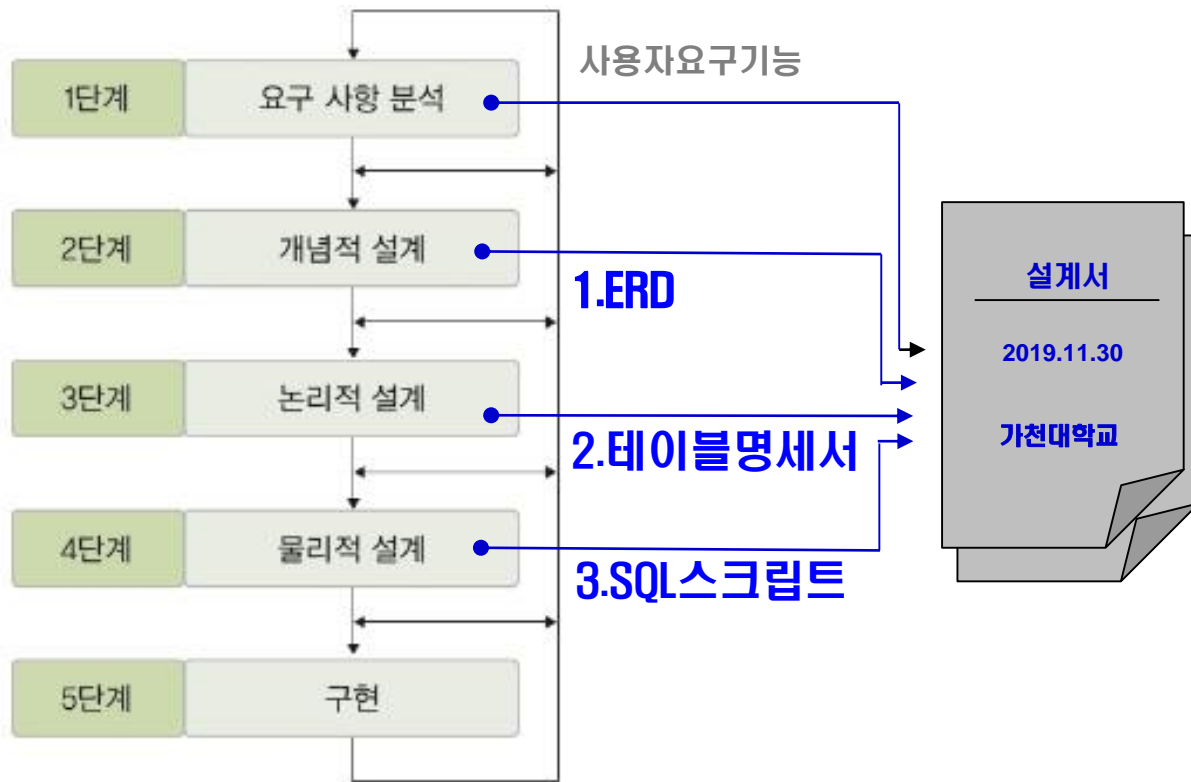


■ API설계 (REST API)

- POST
- GET
- PUT
- DELETE

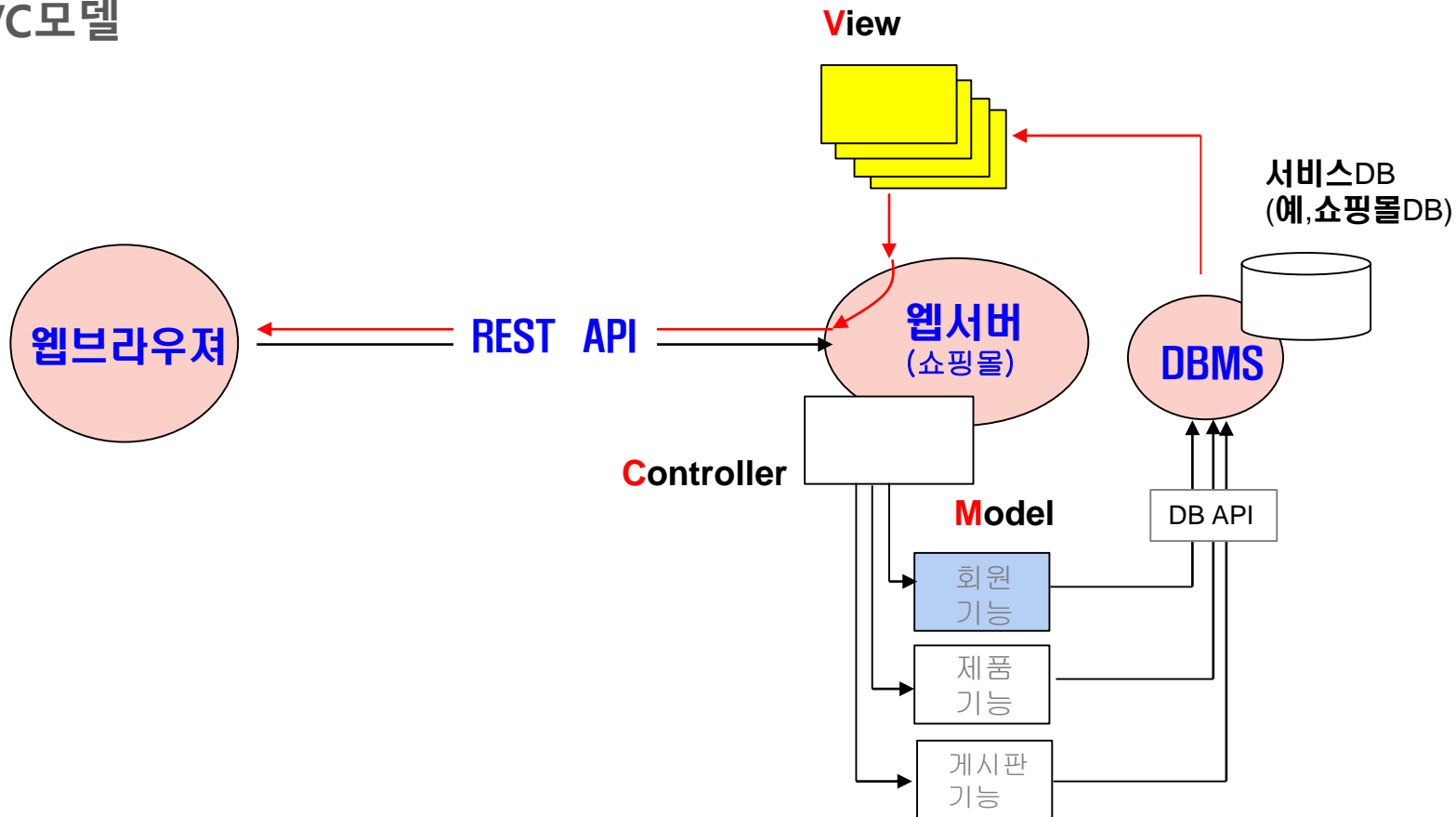
POST /users/reg				
<div> <div>웹브라우저</div> <div>→ 회원가입 (IN)</div> <div>← 가입결과 (OUT)</div> <div>쇼핑몰 서버</div> </div>				
Parameter				
속성	전송방향		Type	Description
	IN	OUT		
<u>userid</u>	O		char	사용자 <u>아이디</u>
<u>pw1</u>	O		char	비밀번호
<u>uname</u>	O		char	이름
status		O	int	200:가입성공 489:아이디틀림 490:암호틀림 491:등록회원아님
<div>전송방향 : IN</div> <pre>{ "userid" : "201912345", "pw1" : "passwd1234", "uname" : "마마무", }</pre> <div>전송방향 : OUT</div> <pre>{ "status" : 200 }</pre>				

■ DB설계



데이터베이스 설계의 과정

■ MVC모델



■ 서비스 개발

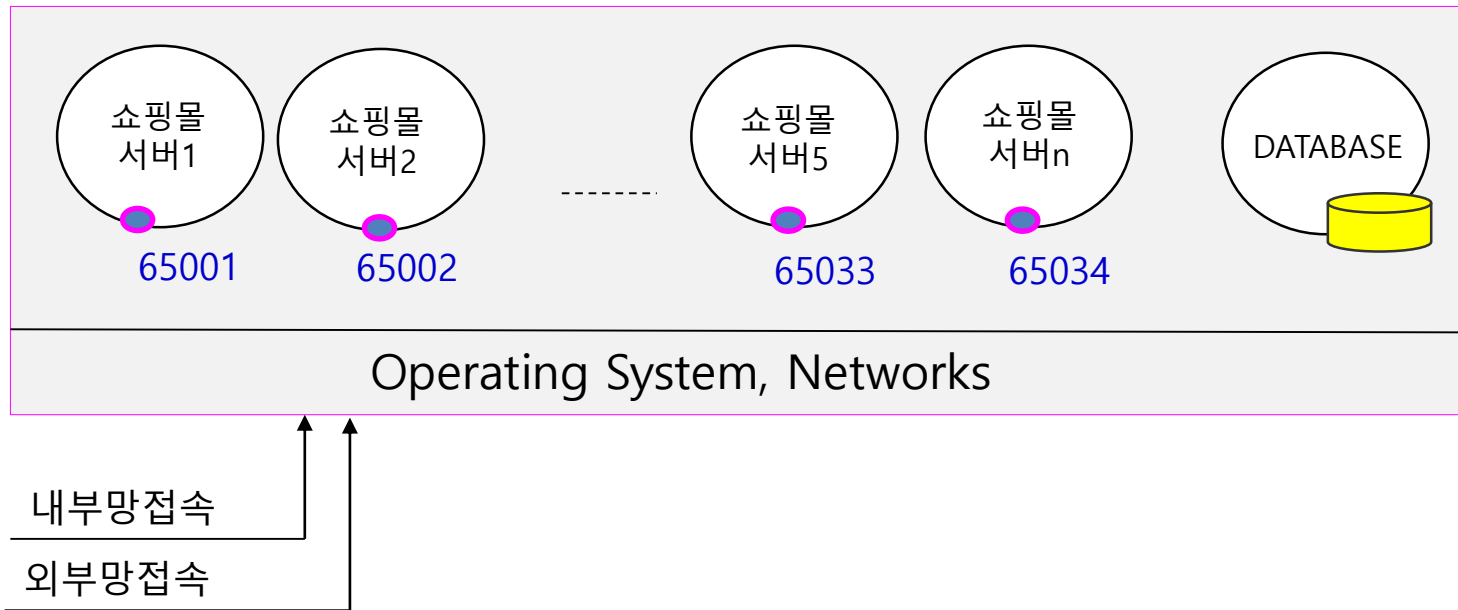
■ 목차

- 01. 서비스 소개(쇼핑몰 웹사이트 구축)
- 02. 개발환경 구축 (Node.js/Express, Bootstrap, jQuery, Atom)
- 03. 개발사례 분석(샘플기능/코드)
- 04. 프로젝트 미션(서비스 완성)

■ 개발환경

- 클라이언트/서버 환경 구조

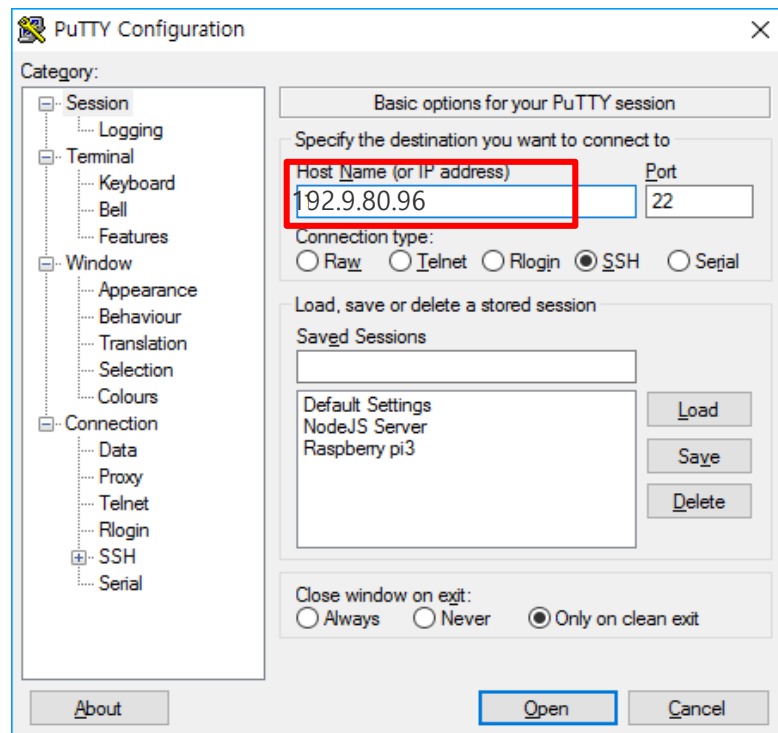
클라우드서버



■ 개발환경

■ 클라우드서버접속

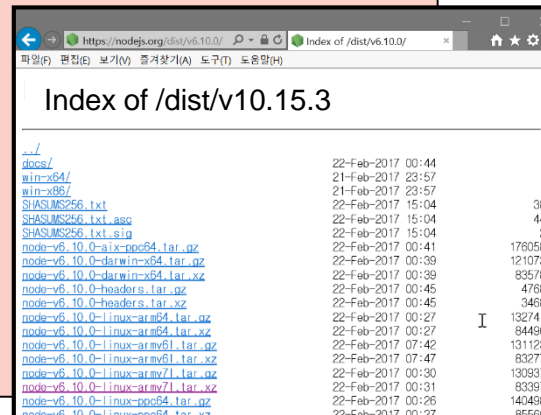
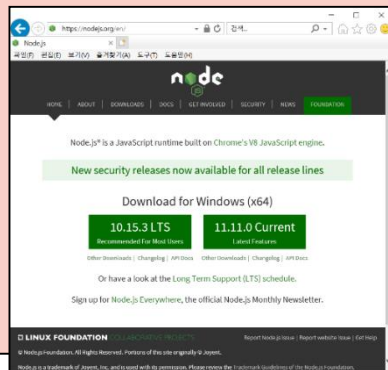
- 내부 IP : 192.9.80.96
- 외부 IP : 203.249.127.60
- 방화벽 오픈 포트 :
 - ssh(22)
 - ftp(21)
 - 개인별(전용계정 및 포트)



■ Node.js/Express 설치

- Linux 환경에서 설치

```
$ su -  
# wget https://nodejs.org/dist/v10.15.3/node-v10.15.3-linux-x64.tar.xz  
# tar -xvf node-v10.15.3-linux-x64.tar.xz  
# cd node-v10.15.3-linux-x64  
# cp -R * /usr  
# exit  
$ node -v  
v10.15.3  
$ npm -v  
6.4.1  
$
```



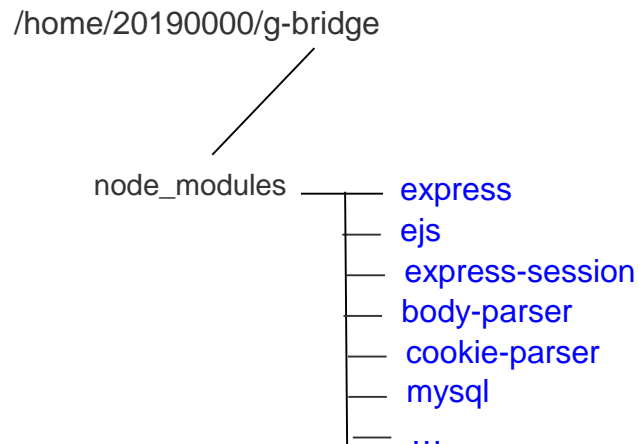
■ 외부모듈 확장설치

- Linux 환경에서 설치

```
$ cd g-bridge
$ pwd
/home/20190000/g-bridge

$ npm install ejs
$ npm install express
$ npm install cookie-parser
$ npm install body-parser
$ npm install express-session
$ npm install mysql
....
```

Local설치 디렉터리 위치



■ 외부모듈 확장설치

- 소스환경에서 설치

```
gbuser99@prjsvr:~  
login as: gbuser99  
gbuser99@192.9.80.96's password:  
Last login: Fri Sep  6 06:40:55 2019 from 116.121.119.217  
[gbuser99@prjsvr ~]$ ls  
[gbuser99@prjsvr ~]$ ftp 192.9.80.96  
Connected to 192.9.80.96 (192.9.80.96).  
220 (vsFTPD 3.0.2)  
Name (192.9.80.96:gbuser99): anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp> cd pub  
250 Directory successfully changed.  
ftp> ls  
227 Entering Passive Mode (192,9,80,96,160,144).  
150 Here comes the directory listing.  
-r--r--r--   1 0      0          1075200 Sep 05 21:23 g-bridgel.0.tar  
-r--r--r--   1 0      0          1986560 Sep 05 21:23 g-bridge2.0.tar  
-r--r--r--   1 0      0           10240 Jun 01 05:45 node-db-sample.tar  
-r--r--r--   1 0      0         11274240 Sep 05 21:35 node_modules.tar  
226 Directory send OK.  
ftp> mget g* node_mo*  
mget g-bridgel.0.tar? y  
227 Entering Passive Mode (192,9,80,96,171,65).  
150 Opening BINARY mode data connection for g-bridgel.0.tar (1075200 bytes).  
226 Transfer complete.  
1075200 bytes received in 0.00276 secs (389283.10 Kbytes/sec)  
mget g-bridge2.0.tar? y  
227 Entering Passive Mode (192,9,80,96,101,105).  
150 Opening BINARY mode data connection for g-bridge2.0.tar (1986560 bytes).  
226 Transfer complete.  
1986560 bytes received in 0.00199 secs (998773.25 Kbytes/sec)  
mget node_modules.tar? y  
227 Entering Passive Mode (192,9,80,96,135,81).  
150 Opening BINARY mode data connection for node_modules.tar (11274240 bytes).  
226 Transfer complete.  
11274240 bytes received in 0.0357 secs (316044.06 Kbytes/sec)  
ftp>
```

■ 외부모듈 확장설치

- 소스환경에서 설치

```
gbuser99@prjsvr:~  
[gbuser99@prjsvr ~]$ tar xvf g-bridge1.0.tar  
g-bridge/  
g-bridge/public/  
g-bridge/public/images/  
g-bridge/public/images/logo.jpg  
g-bridge/public/images/banner1.png  
g-bridge/public/images/product/  
g-bridge/public/images/product/2.jpg  
g-bridge/public/images/product/1.jpg  
g-bridge/public/images/admin_banner.png  
g-bridge/public/javascripts/  
g-bridge/public/stylesheets/  
g-bridge/public/stylesheets/style.css  
g-bridge/routes/  
g-bridge/routes/mainui.js  
g-bridge/routes/adminprod.js  
g-bridge/routes/users.js  
g-bridge/views/  
g-bridge/views/error.ejs  
g-bridge/views/content.ejs  
g-bridge/views/header.ejs  
g-bridge/views/navbar.ejs  
g-bridge/views/reg_form.ejs  
g-bridge/views/login_form.ejs  
g-bridge/views/alert.ejs  
g-bridge/views/footer.ejs  
g-bridge/views/adminbar.ejs  
g-bridge/views/adminproduct.ejs  
g-bridge/views/admincontent.ejs  
g-bridge/uploads/  
g-bridge/uploads/prod/  
g-bridge/server.js  
[gbuser99@prjsvr ~]$ tar xvf g-bridge2.0.tar
```

■ 외부모듈 확장설치

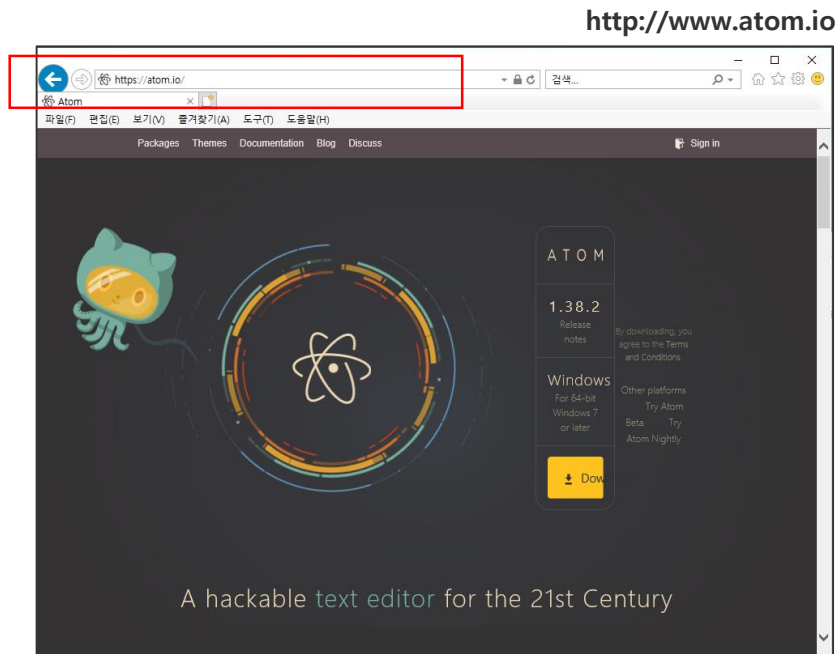
- 소스환경에서 설치

```
gbuser99@prjsvr:~  
[gbuser99@prjsvr ~]$ tar xvf node_modules.tar  
node_modules/  
node_modules/bytes/  
node_modules/bytes/package.json  
node_modules/bytes/LICENSE  
node_modules/bytes/index.js  
node_modules/bytes/Readme.md  
node_modules/content-type/  
node_modules/content-type/package.json  
node_modules/content-type/README.md  
node_modules/content-type/LICENSE  
node_modules/content-type/index.js  
node_modules/content-type/HISTORY.md  
node_modules/depd/  
node_modules/depd/package.json  
node_modules/depd/LICENSE  
node_modules/depd/index.js  
node_modules/depd/History.md  
node_modules/depd/Readme.md  
node_modules/depd/lib/  
node_modules/depd/lib/browser/  
node_modules/depd/lib/browser/index.js  
node_modules/depd/lib/compat/
```

```
gbuser99@prjsvr:~/g-bridge  
[gbuser99@prjsvr ~]$ ls  
g-bridge g-bridge1.0.tar g-bridge2.0 g-bridge2.0.tar node_modules node_modules.tar  
[gbuser99@prjsvr ~]$ cd g-bridge  
[gbuser99@prjsvr g-bridge]$ ls  
public routes server.js uploads views  
[gbuser99@prjsvr g-bridge]$ node server.js  
서버실행 : http://203.249.127.60:65012  
서버실행 : http://192.9.96.96:65012
```


■ Atom 에디터 설치

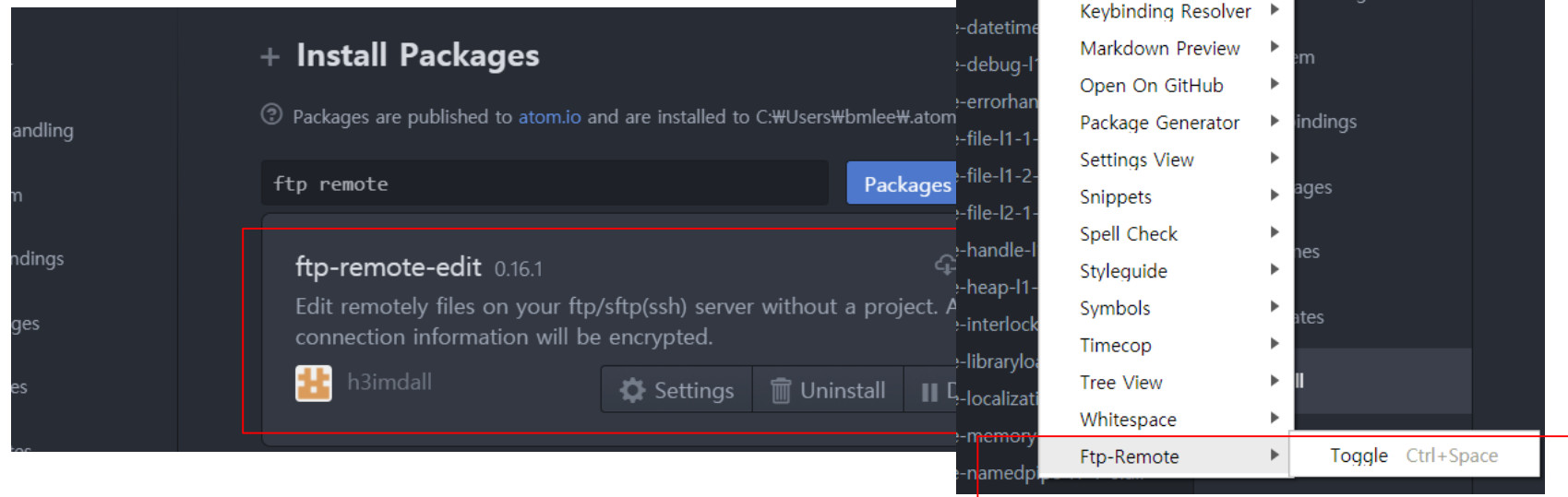
- Windows 10 환경에서 설치



■ Atom 에디터 설치

- Atom 추가패키지 설치

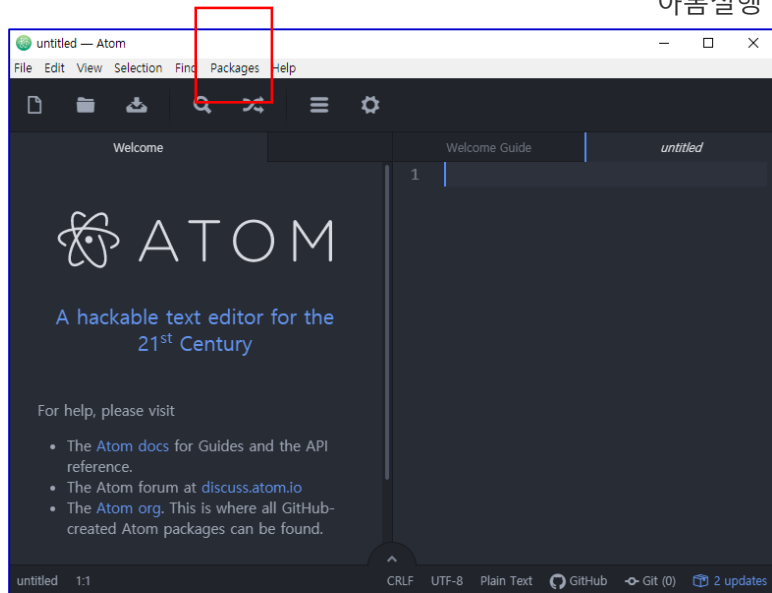
FTP-remote-edit, Bracket Matcher, Github 등



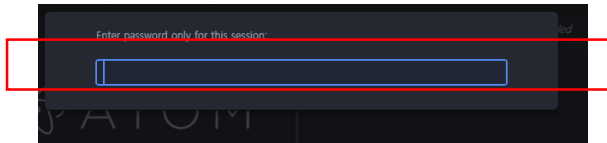
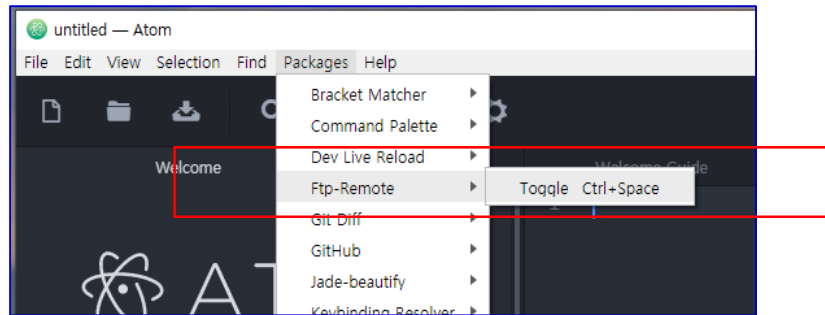
■ Atom 활용

■ 프로그래밍

아톰실행



FTP서버접속

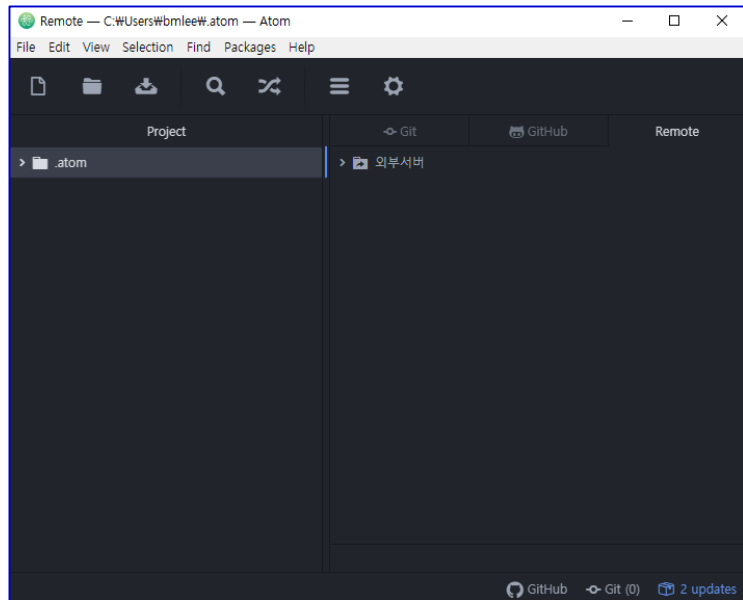


(처음에 Ftp-Remote 세션에) 등록했었던 암호입력

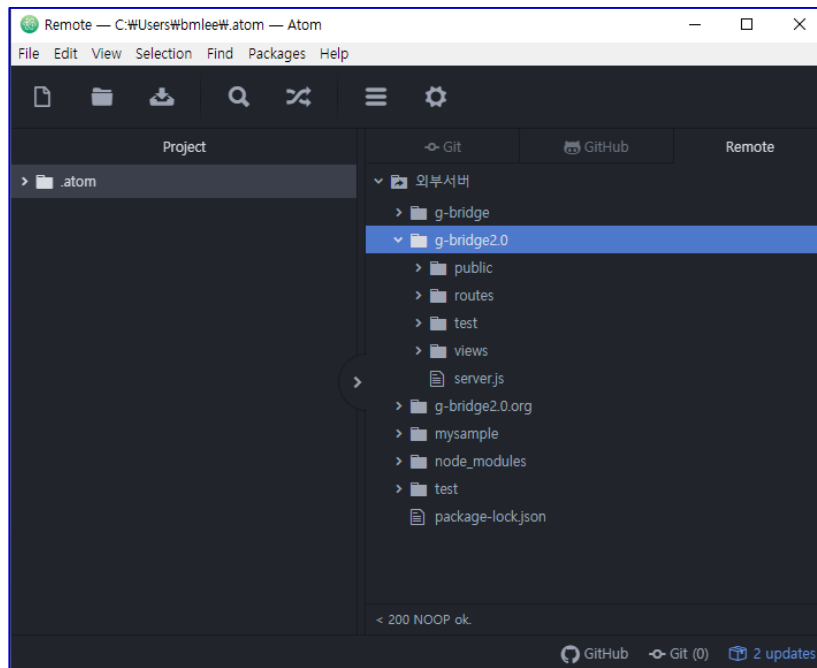
■ Atom 활용

■ 프로그래밍

FTP서버 접속세션

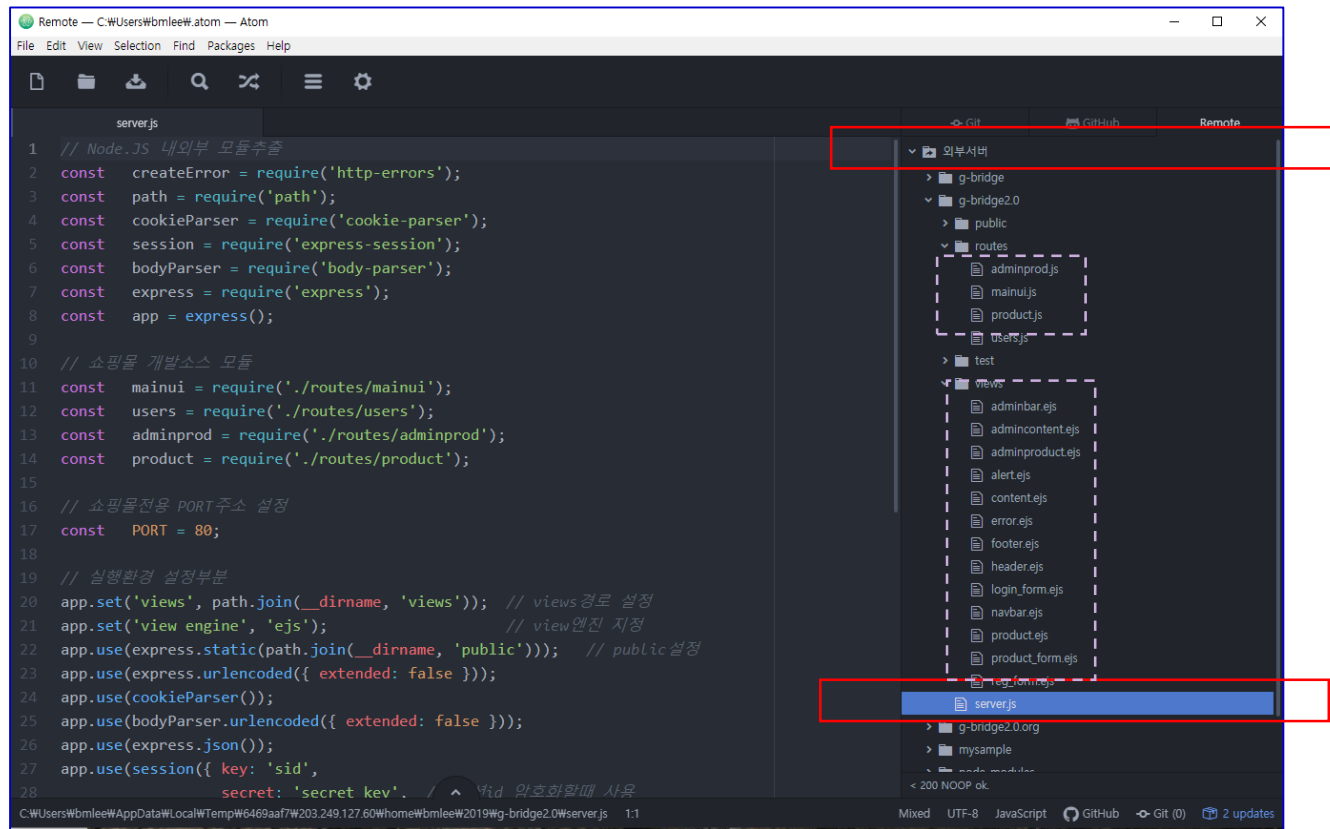


서버소스 디렉터리에 접속



■ Atom 활용

■ 프로그래밍



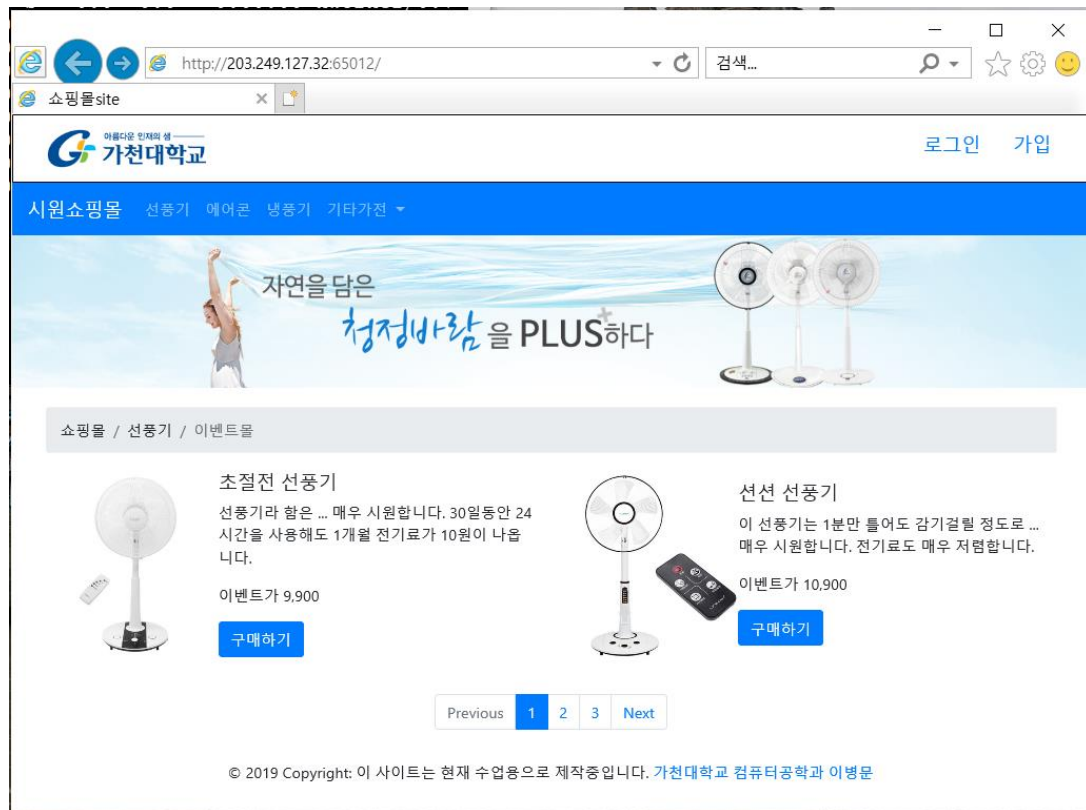
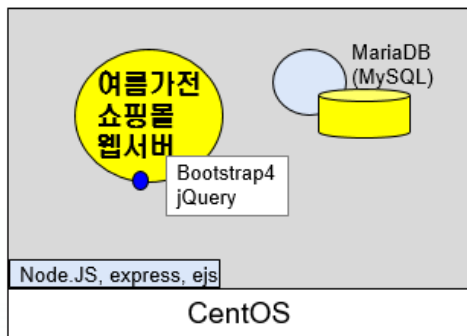
■ 서비스 소개

■ 목차

- 01. 서비스 소개(쇼핑몰 웹사이트 구축)
- 02. 개발환경 구축 (Node.js/Express, Bootstrap, Atom)
- 03. 개발사례 분석(샘플기능/코드)
- 04. 프로젝트 미션(서비스 완성)

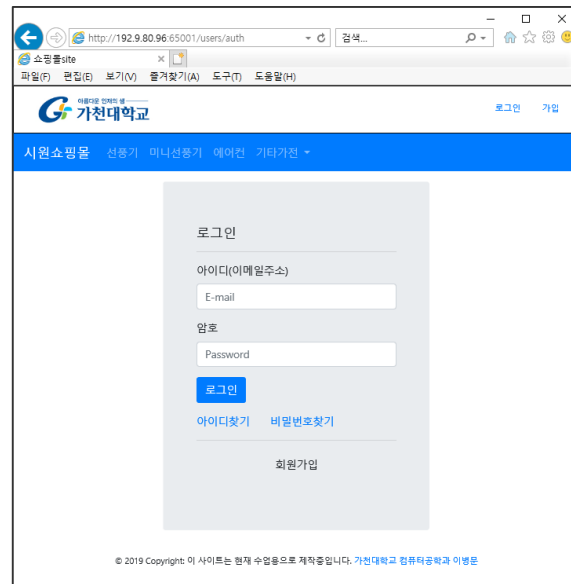
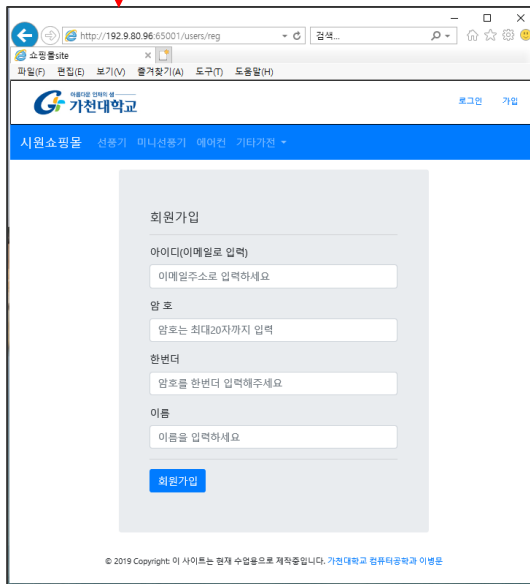
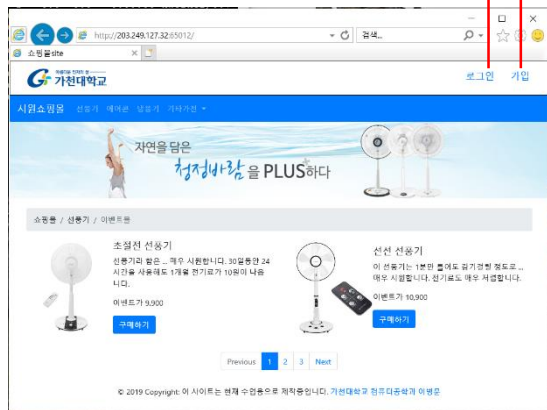
■ 개발서비스

- 여름가전 쇼핑몰 서비스



■ 개발서비스

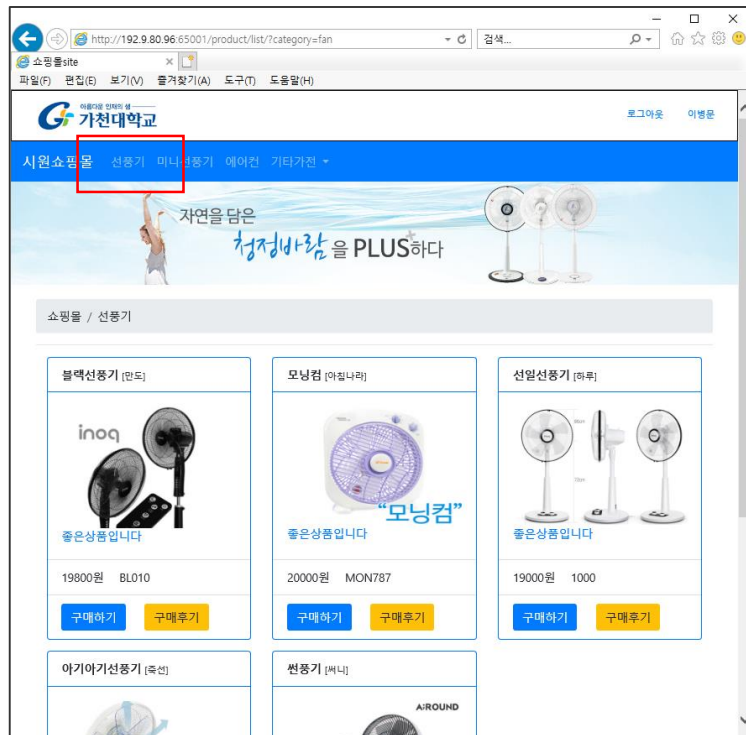
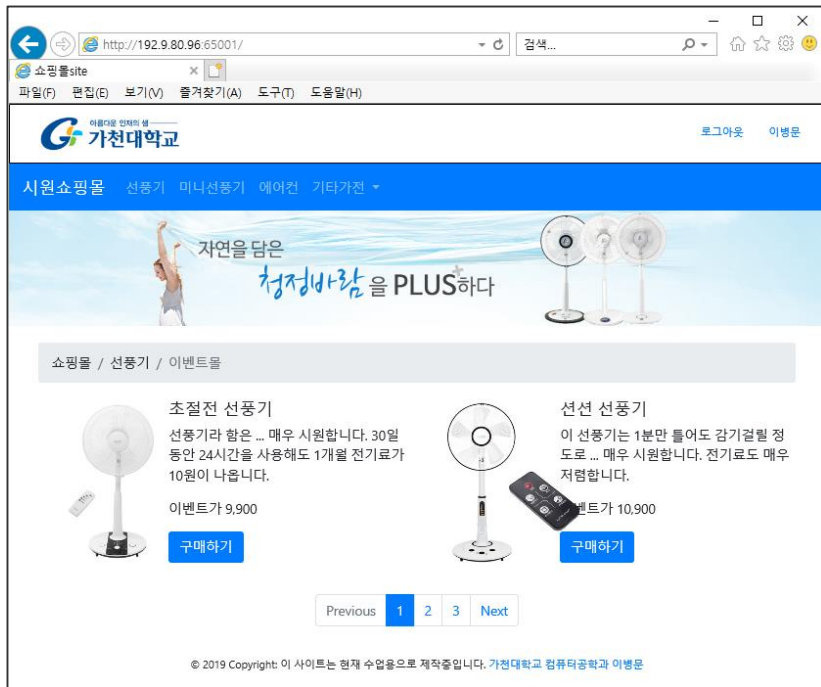
■ 여름가전 쇼핑몰 서비스



■ 개발서비스

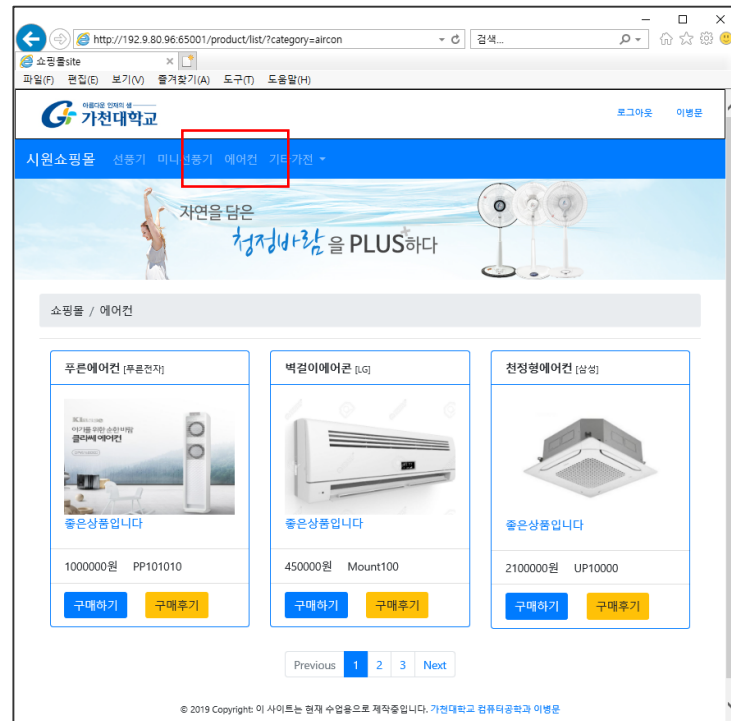
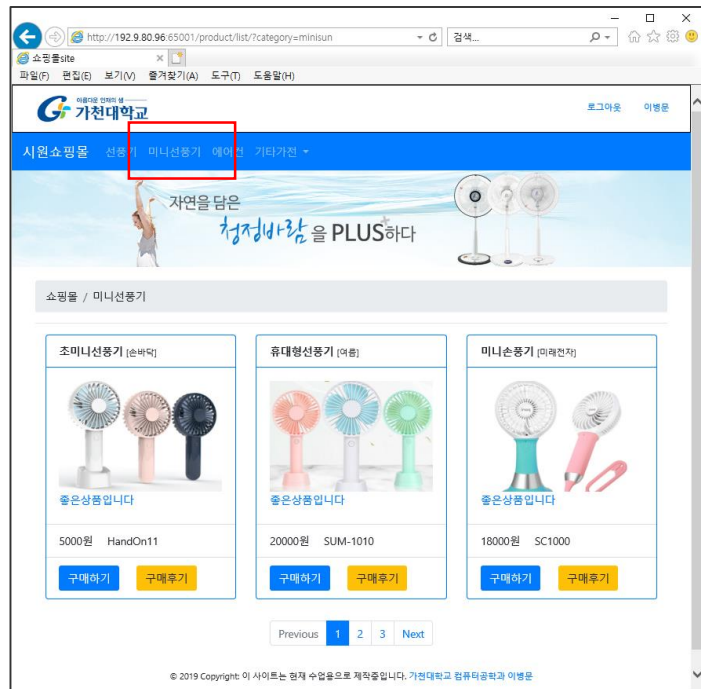
■ 여름가전 쇼핑몰 서비스

로그인후



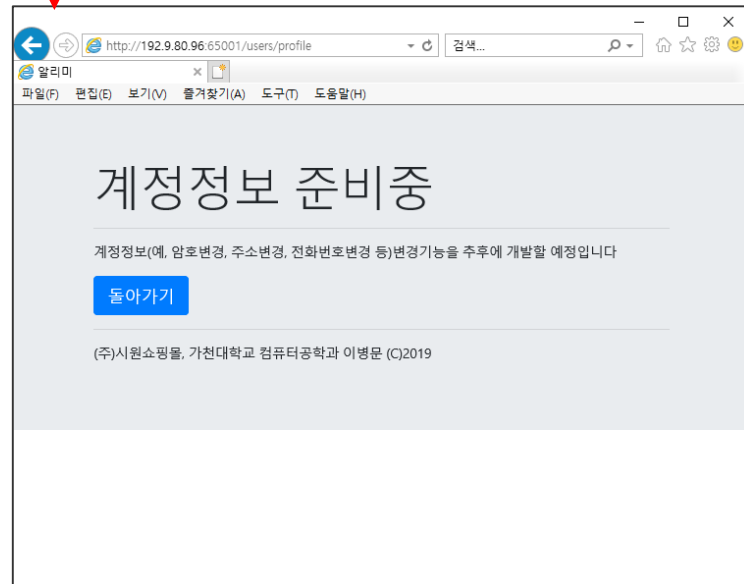
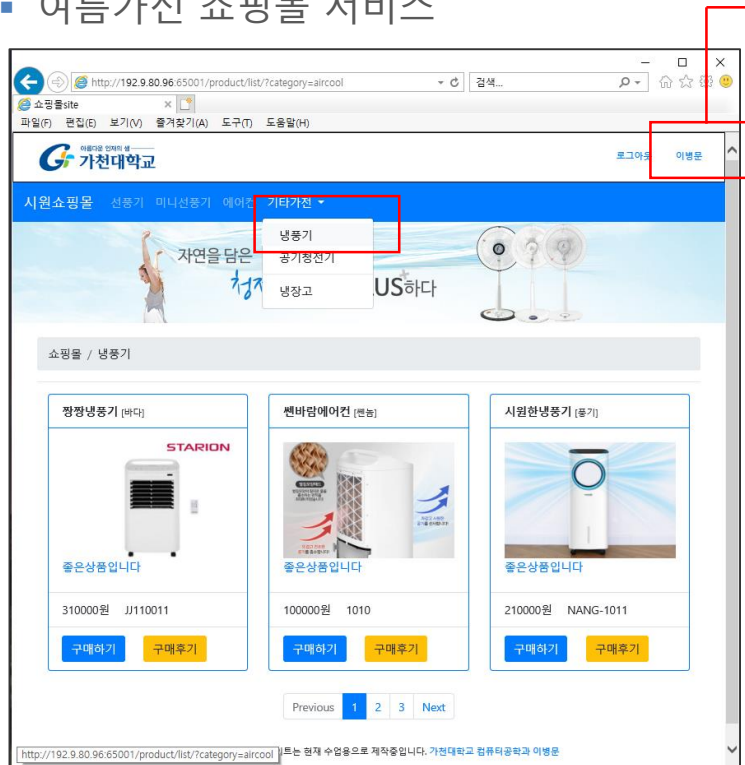
■ 개발서비스

■ 여름가전 쇼핑몰 서비스



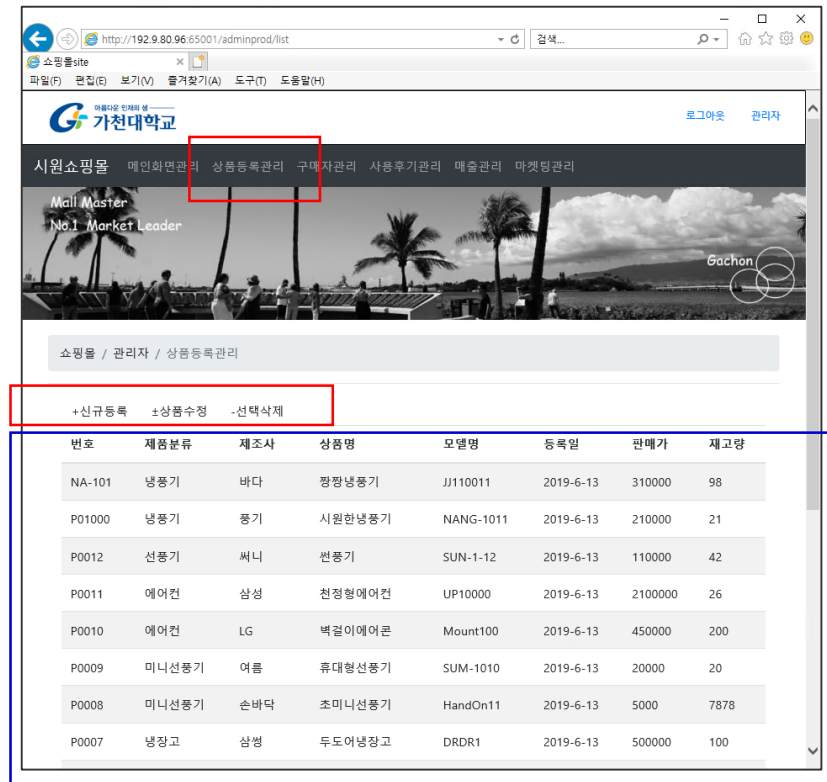
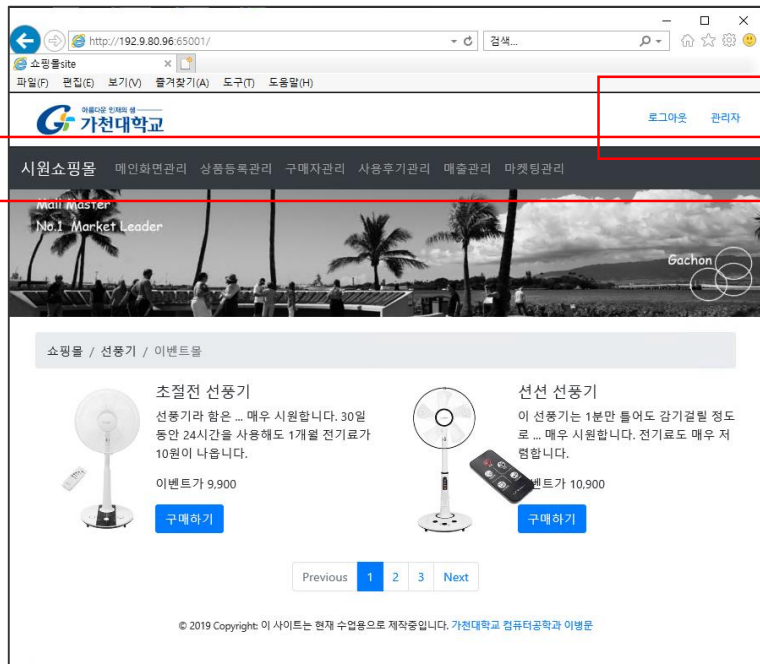
■ 개발서비스

■ 여름가전 쇼핑몰 서비스



■ 개발서비스

■ 여름가전 쇼핑몰 서비스(관리자)

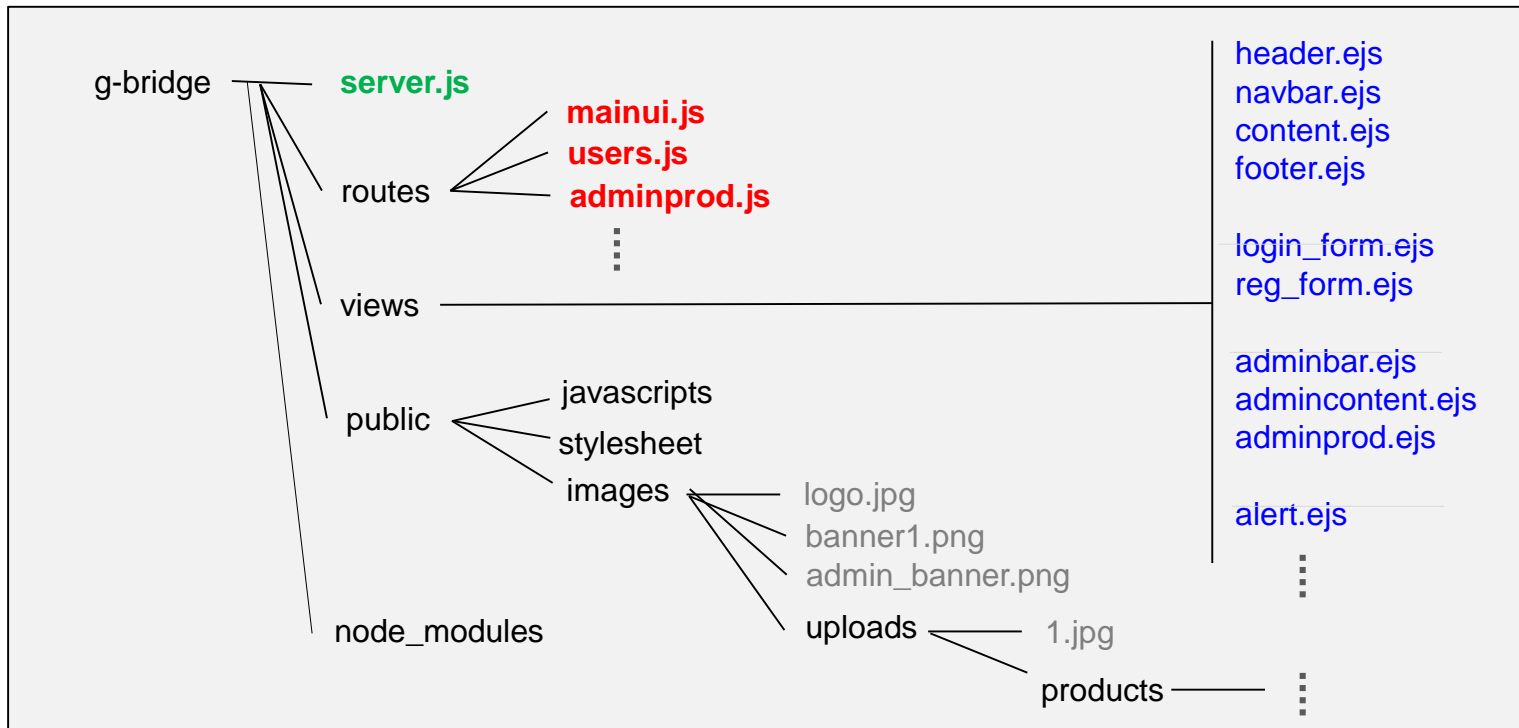


■ 소스디렉터리 구조

■ MVC모델구조

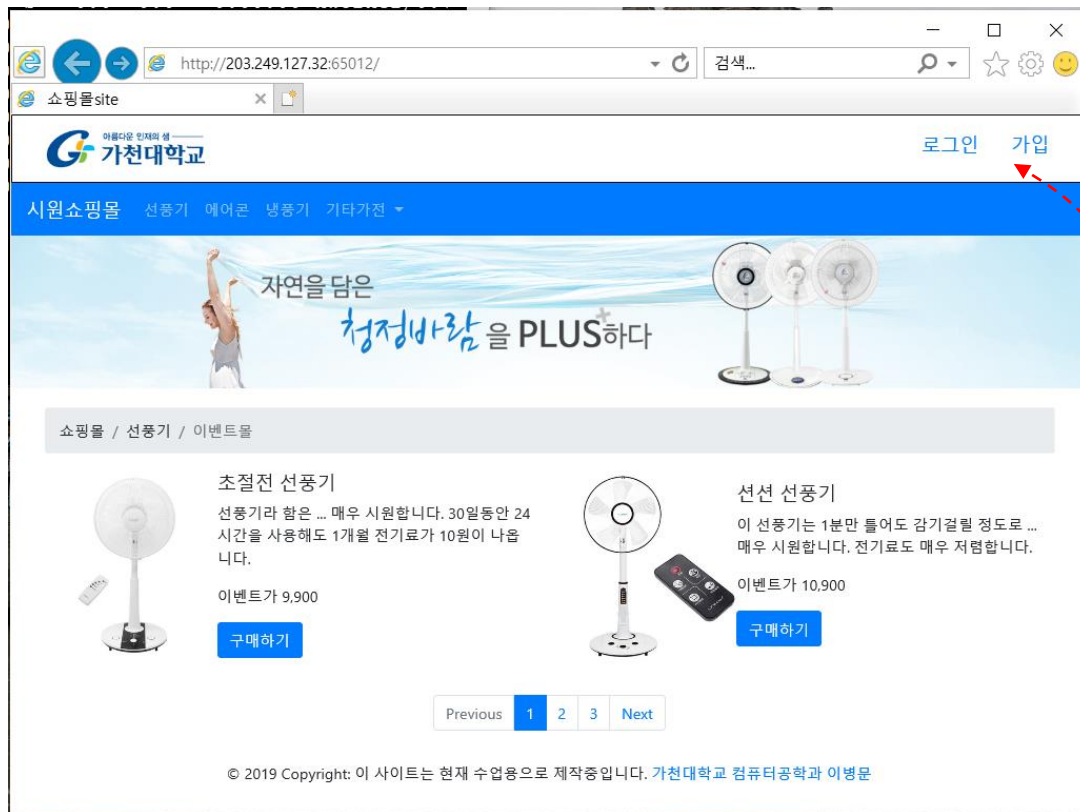
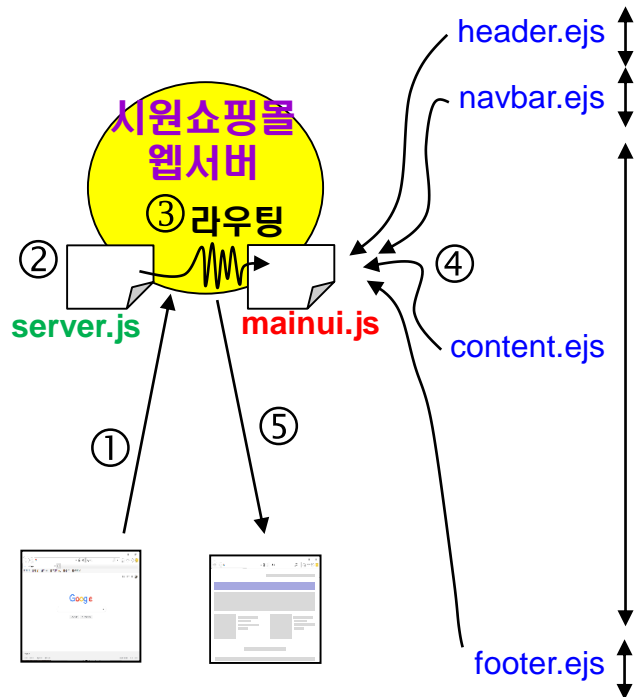
front-end 프레임워크 : bootstrap JQuery

back-end 프레임워크 : Node.JS, express, [ejs](#), mysql



■ 샘플소스 분석

- server.js / mainui.js



로그인전 / 로그인후 처리필요

■ 샘플소스 분석

```
const express = require('express');
const createError = require('http-errors');
const path = require('path');
const cookieParser = require('cookie-parser');
const session = require('express-session');
const bodyParser = require('body-parser');
```

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

```
const mainui = require('./routes/mainui');
const users = require('./routes/users');
const adminprod = require('./routes/adminprod');
const PORT = 65012;
```

// 실행환경 (초기화)설정

```
app.set('views', path.join(__dirname, 'views')); // views경로 설정
app.set('view engine', 'ejs'); // view엔진 지정
app.use(express.static(path.join(__dirname, 'public'))); // public 설정 (브라우저부분 설정)
app.use(express.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(express.json());
app.use(session({ key: 'sid',
                  secret: 'secret key', // 세션id 암호화할때 사용
                  resave: false, // 접속할때마다 id부여금지
                  saveUninitialized: true })); // 세션id사용전에는 발급금지
```

// URI와 핸들리를 매핑

```
app.use('/', mainui); // / 메인화면기능
app.use('/users', users); // '/users' 사용자기능
app.use('/adminprod', adminprod); // 'adminprod' 상품등록관리
```

// 쇼핑몰 웹서버를 실행합니다.

```
app.listen(PORT, function () {
  console.log('서버실행: http://203.249.127.32:' + PORT);
  console.log('서버실행: http://192.9.44.53:' + PORT);
});
```

server.js

■ 샘플소스 분석

mainui.js

// 메인화면 출력파일

```
const express = require('express');
const ejs = require('ejs');
const fs = require('fs');
const router = express.Router();
var loglevel = 1;
```

```
const GetMainUI = (req, res) => { // 메인화면을 출력합니다
  let htmlstream = '';
```

```
  logging(loglevel, 'router.get() invoked!');
```

```
  htmlstream = fs.readFileSync(__dirname + '/../views/header.ejs','utf8'); // 헤더부분
```

```
  if (req.session.auth && req.session.admin) { // 만약, 관리자가 로그인했다면
```

```
    htmlstream = htmlstream + fs.readFileSync(__dirname + '/../views/adminbar.ejs','utf8'); // 관리자메뉴
```

```
    htmlstream = htmlstream + fs.readFileSync(__dirname + '/../views/admincontent.ejs','utf8');
```

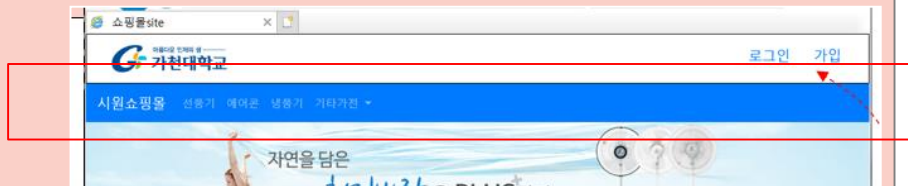
```
  } else {
```

```
    htmlstream = htmlstream + fs.readFileSync(__dirname + '/../views/navbar.ejs','utf8'); // 일반사용자메뉴
```

```
    htmlstream = htmlstream + fs.readFileSync(__dirname + '/../views/content.ejs','utf8'); // Content
```

```
  }
```

```
  htmlstream = htmlstream + fs.readFileSync(__dirname + '/../views/footer.ejs','utf8'); // Footer
```



■ 샘플소스 분석

mainui.js

```
res.writeHead(200, {'Content-Type':'text/html; charset=utf8'});
if (req.session.auth) { // true :로그인된 상태, false : 로그인안된 상태
  res.end(ejs.render(htmlstream, { 'title' : '쇼핑몰site',
                                'logurl': '/users/logout',
                                'loglabel': '로그아웃',
                                'regurl': '/users/profile',
                                'reglabel':req.session.who })); // 세션에 저장된 사용자명표시
}
else {
  res.end(ejs.render(htmlstream, { 'title' : '쇼핑몰site',
                                'logurl': '/users/auth',
                                'loglabel': '로그인',
                                'regurl': '/users/reg',
                                'reglabel':'가입' }));
}

};

const logging = (level, logmsg) => {
  if (level != 0) {
    console.log(level, logmsg)
    loglevel++;
  }
}
```

```
// '/' get 메소드의 핸들러를 정의
router.get('/', GetMainUI);
```

```
// 외부로 뱉니다.
module.exports = router
```

■ 샘플소스 분석

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title> <%= title %> </title>
  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"
    crossorigin="anonymous">

  <!-- 스크립트 코딩크 (CND)를 지정 -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-?????????XpG5KkN"
    crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-??0b4Q"
    crossorigin="anonymous"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-???07jRM"
    crossorigin="anonymous"></script>
</head>
<body>
  <!-- 헤더부분 -->
  <div class="container-fluid" style="border: solid 1px black; border-radius: 1px">
    <div class="row align-items-center">
      <div class="col-2"></div>
      <div class="col-8"> </div>
      <div class="col-1"> <a href='<%=logurl%>'><h5><%=loglabel%></h5></a></div>
      <div class="col-1"> <a href='<%=regurl%>'><h5><%=reglabel%></h5></a></div>
    </div>
  </div>
```

header.ejs

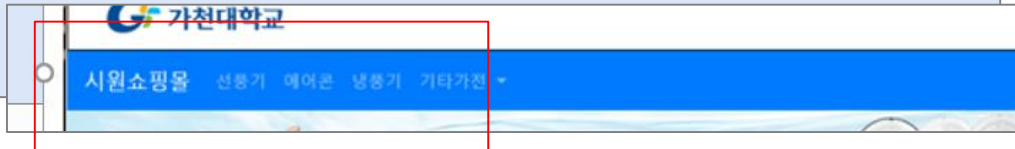
■ 샘플소스 분석

navbar.ejs

```
<!DOCTYPE html>
<!-- GNB(Global Navigation Bar) 부분 -->
<nav class="navbar navbar-expand-md navbar-dark bg-primary">

  <a class="navbar-brand" href="/">시원쇼핑몰</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
    aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>

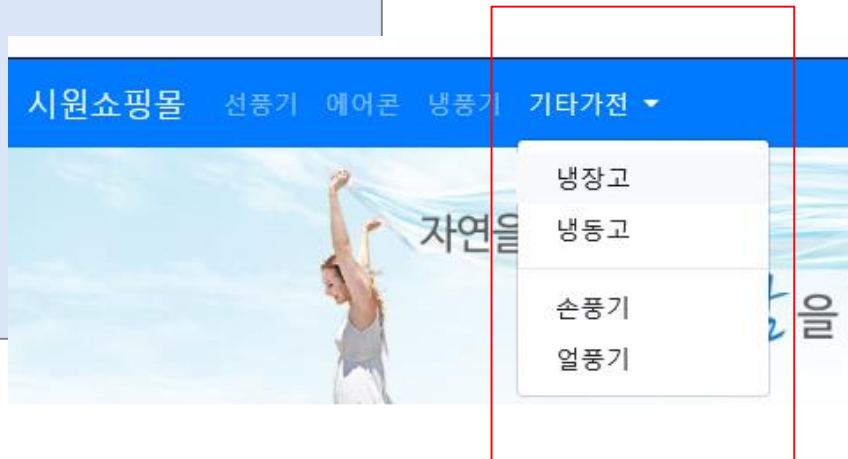
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item">
        <a class="nav-link" href="#">선풍기<span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">에어콘</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">냉풍기</a>
      </li>
```



■ 샘플소스 분석

```
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
    role="button" data-toggle="dropdown" aria-haspopup="true"
    aria-expanded="false">기타가전 </a>
  <div class="dropdown-menu" aria-labelledby="navbarDropdown">
    <a class="dropdown-item" href="#">냉장고</a>
    <a class="dropdown-item" href="#">냉동고</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item" href="#">손풍기</a>
    <a class="dropdown-item" href="#">열풍기</a>
  </div>
</li>
</ul>
</div>
</nav>
```

navbar.ejs



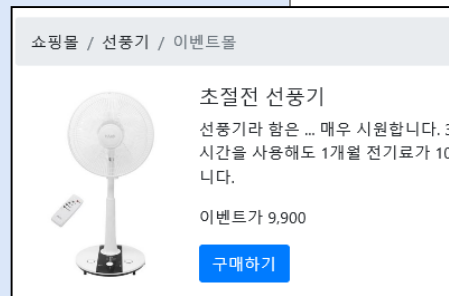
■ 샘플소스 분석

```

<!DOCTYPE html>
<!-- Content (배너이미지) 부분 -->
<div>  </div>

<!-- 컨텐츠(예, 쇼핑몰일 경우 판매제품리스트)헤더부분 -->
<div class="container-fluid" style="border: solid 20px white; border-radius: 1px">
  <div class="panel panel-default">
    <div class="panel-heading">
      <nav aria-label="breadcrumb">
        <ol class="breadcrumb">
          <li class="breadcrumb-item">쇼핑몰</li>
          <li class="breadcrumb-item">선풍기</li>
          <li class="breadcrumb-item active" aria-current="page">이벤트몰</li>
        </ol>
      </nav>
    </div>
    <div class="panel-body">
      <div class="row align-items-center">
        <div class="col-2"></div>
        <div class="col-4">
          <h5 class="mt-0">초절전 선풍기</h5>
          <p>선풍기라 함은 ... 매우 시원합니다. 30일동안 24시간을 사용해도 1개월 전기료가 10원이 나옵니다.</p>
          <p>이벤트가 9,900</p>
          <button type="button" class="btn btn-primary">구매하기</button>
        </div>
      </div>
    </div>
  </div>
</div>

```



■ 샘플소스 분석

```
<div class="col-2">
  
</div>
<div class="col-4">
  <h5 class="mt-0">선풍기</h5>
  <p class="xs">이 선풍기는 1분만 틀어도 감기걸릴 정도로 ... 매우 시원합니다. 전기료도 매우 저렴합니다.</p>
  <p>이벤트가 10,900</p>
  <button type="button" class="btn btn-primary">구매하기</button>
</div>
</div> </div> </div> </div>

<p>
<nav aria-label="...">
  <ul class="pagination justify-content-center">
    <li class="page-item disabled">
      <a class="page-link" href="#" tabindex="-1" aria-disabled="true">Previous</a>
    </li>
    <li class="page-item active" aria-current="page">
      <a class="page-link" href="#">1 <span class="sr-only">(current)</span></a>
    </li>
    <li class="page-item"> <a class="page-link" href="#">2</a> </li>
    <li class="page-item"> <a class="page-link" href="#">3</a> </li>
    <li class="page-item"> <a class="page-link" href="#">Next</a> </li>
  </ul>
</nav>
```

content.ejs

Previous 1 2 3 Next

■ 샘플소스 분석

footer.ejs

```
<!doctype html>
<!-- Footer -->
<p>
  <footer class="page-footer font-small blue">
    <div class="footer-copyright text-center py-3">© 2019 Copyright: 이 사이트는 현재 수업용으로
      제작중입니다.
      <a href="#"> 가천대학교 컴퓨터공학과 이병문 </a>
    </div>
  </footer>

</body>
</html>
```

© 2019 Copyright: 이 사이트는 현재 수업용으로 제작중입니다. 가천대학교 컴퓨터공학과 이병문

■ 샘플소스 분석

```
const bodyParser = require('body-parser');
const session = require('express-session');
const router = express.Router();

router.use(bodyParser.urlencoded({ extended: false }));

const db = mysql.createConnection({
  host: 'localhost',          // DB서버 IP주소
  port: 3306,                 // DB서버 Port주소
  user: 'bmlee',              // DB접속 아이디
  password: 'bmlee654321',    // DB암호
  database: 'bmlee'           //사용할 DB명
});

// ----- 회원가입기능 -----
// 회원가입 입력양식을 브라우저로 출력합니다.
const PrintRegistrationForm = (req, res) => {
  let htmlstream = "";

  htmlstream = fs.readFileSync(__dirname + '/../views/header.ejs', 'utf8');
  htmlstream = htmlstream + fs.readFileSync(__dirname + '/../views/navbar.ejs', 'utf8');
  htmlstream = htmlstream + fs.readFileSync(__dirname + '/../views/reg_form.ejs', 'utf8');
  htmlstream = htmlstream + fs.readFileSync(__dirname + '/../views/footer.ejs', 'utf8');

  res.writeHead(200, {'Content-Type': 'text/html'; charset: 'utf8'});
```

```
if (req.session.auth) { // true :로그인된 상태,
  res.end(ejs.render(htmlstream, { 'title': '쇼핑몰site',
    'logurl': '/users/logout',
    'loglabel': '로그아웃',
    'regurl': '/users/profile',
    'reglabel': req.session.who }));
}
else { // false : 로그인안된 상태
  res.end(ejs.render(htmlstream, { 'title': '쇼핑몰site',
    'logurl': '/users/auth',
    'loglabel': '로그인',
    'regurl': '/users/reg',
    'reglabel': '가입' }));
}

};
```

users.js

■ 샘플소스 분석

// 회원가입 양식에서 입력된 회원정보를 신규등록(DB에 저장)합니다.

```
const HandleRegistration = (req, res) => { // 회원가입
```

```
  let body = req.body;
```

```
  let htmlstream="";
```

// 임시로 확인하기 위해 콘솔에 출력해봅니다.

```
console.log(body.uid); console.log(body.pw1); console.log(body.uname);
```

```
if (body.uid == "" || body.pw1 == "") {
```

```
  console.log("데이터입력이 되지 않아 DB에 저장할 수 없습니다.");
```

```
  res.status(561).end('<meta charset="utf-8">데이터가 입력되지 않아 가입을 할 수 없습니다');
```

```
}
```

```
else {
```

```
  db.query('INSERT INTO users (uid, pass, name) VALUES (?, ?, ?)',
```

```
    [body.uid, body.pw1, body.uname], (error, results, fields) => {
```

```
    if (error) {
```

```
      htmlstream = fs.readFileSync(__dirname + '/../views/alert.ejs','utf8');
```

```
      res.status(562).end(ejs.render(htmlstream, { 'title': '알리미',
```

```
        'warn_title': '회원가입 오류',
```

```
        'warn_message': '이미 회원으로 등록되어 있습니다. 바로 로그인을 하시기 바랍니다.',
```

```
        'return_url': '/' }));
```

```
    } else {
```

```
      console.log("회원가입에 성공하였으며, DB에 신규회원으로 등록하였습니다.");
```

```
      res.redirect('/');
```

```
    }
```

```
  });
```

```
} };
```

users.js

// REST API의 URI와 핸들러를 매핑합니다.

```
router.get('/reg', PrintRegistrationForm); // 회원가입화면을 출력처리
```

```
router.post('/reg', HandleRegistration); // 회원가입내용을 DB에 등록처리
```

```
router.get('/', function(req, res) { res.send('respond with a resource 111'); });
```

users.js

[illegible]

■ 샘플소스 분석

// 로그인을 수행합니다. (사용자인증처리)

```
const HandleLogin = (req, res) => {  
  let body = req.body;  
  let userid, userpass, username;  
  let sql_str;  
  let htmlstream = "";  
  
  console.log(body.uid);  
  console.log(body.pass);  
  if (body.uid == "" || body.pass == "") {  
    console.log("아이디나 암호가 입력되지 않아서 로그인할 수 없습니다.");  
    res.status(562).end('<meta charset="utf-8">아이디나 암호가 입력되지 않아서 로그인할 수 없습니다.');
```

users.js

```
}  
}
```

■ 샘플소스 분석

users.js

```
else { // select 조회결과가 있는 경우 (즉, 등록사용자인 경우)
  results.forEach((item, index) => {
    userid = item.uid; userpass = item.pass; username = item.name;
    console.log("DB에서 로그인성공한 ID/암호:%s/%s", userid, userpass);
    if (body.uid == userid && body.pass == userpass) {
      req.session.auth = 99; // 임의로 수(99)로 로그인성공했다는 것을 설정함
      req.session.who = username; // 인증된 사용자명 확보 (로그인후 이름출력용)
      if (body.uid == 'admin') // 만약, 인증된 사용자가 관리자(admin)라면 이를 표시
        req.session.admin = true;
      res.redirect('/');
    }
  }); /* foreach */
} // else
} // else
});
}
}

// REST API의 URI와 핸들러를 매핑합니다.
// URI: http://xxxx/users/auth
router.get('/auth', PrintLoginForm); // 로그인 입력화면을 출력
router.post('/auth', HandleLogin); // 로그인 정보로 인증처리
```

■ 샘플소스 분석

users.js

```
// ----- 로그아웃기능 -----
const HandleLogout = (req, res) => {
  req.session.destroy(); // 세션을 제거하여 인증오작동 문제를 해결
  res.redirect("/");      // 로그아웃후 메인화면으로 재접속
}

// REST API의 URI와 핸들러를 매핑합니다.
router.get('/logout', HandleLogout); // 로그아웃 기능

// ----- 정보변경 기능을 개발합니다 -----
const PrintProfile = (req, res) => {
  let htmlstream = "";

  htmlstream = fs.readFileSync(__dirname + '/../views/alert.ejs','utf8');
  res.status(562).end(ejs.render(htmlstream, { 'title': '알리미',
    'warn_title':'계정정보 준비중',
    'warn_message':'계정정보(예, 암호변경, 주소변경, 전화번호변경 등)변경기능을 추후에 개발할 예정입니다',
    'return_url':'/' })));
}

router.get('/profile', PrintProfile); // 정보변경화면을 출력

module.exports = router;
```

■ 프로젝트 미션