

최종 보고서

(2023학년도 1학기)

| | |
|-----------------|---|
| 과제명 | 네이버 쇼핑 API를 통한 최저가 검색 Server |
| 제출일자 | 2023. 06. 08 |
| 연구참여자 (담당분야) | 팀장: 허진환(보고서 작성, 가상머신 환경 설정 및 웹서버 구축/DB 관리) 팀원: 김태현(회의록 작성, Main/UI/Additional Function Programming) 팀원: 정재호(자료조사, Ranking Algorithm Function/UI Programming) |
| 수업명/조번호 | 웹서버프로그래밍(7-9)/7조 |

목 차

| | |
|--------------------|---|
| 1. 설계과제 제목 | 1 |
| 2. 연구목적 | 1 |
| 3. 설계과제의 필요성 | 1 |
| 4. 설계과제의 목표 | 1 |
| 5. 설계과정 | 1 |
| 6. 제작 | 1 |
| 7. 시험 | 1 |
| 8. 평가 | 1 |
| 9. 추진체계 | 1 |
| 10. 설계 추진일정 | 1 |
| 11. 결론 | 1 |

부록

가. 프로그램 소스코드 (커멘트 포함)

1. 설계과제의 제목 :

네이버 쇼핑몰 API를 통한 최저가 검색 Server

2. 연구목적

- 1) Linux Cent OS 상에 Node.js 웹서버 생성 및 웹서버 구조 이해
- 2) 익스프레스 프레임워크 활용
- 3) Javascript 모듈/패키지 및 다양한 함수 활용
- 4) 클라이언트 - Server간 동작 이해
- 5) 네이버 쇼핑 API 데이터 이해 및 활용
- 6) MySQL을 통한 DBMS 구조 및 데이터베이스를 통한 데이터 관리 이해

3. 설계과제의 필요성

시간에 쫓겨사는 현대인들에게 편리한 쇼핑 경험을 제공하는 인터넷 쇼핑몰은 고객들이 가정이나 사무실에서 인터넷에 접속하여 제품을 검색하고 주문할 수 있게 한다. 인터넷 쇼핑몰은 시간과 장소에 구애받지 않고 24시간 언제든지 쇼핑이 가능하며, 전 세계 어디에서든 접속할 수 있다. 또한, 인터넷 쇼핑몰은 수많은 상품과 다양한 카테고리들을 제공하는데, 제한된 공간 제약이 없으므로 오프라인 쇼핑몰보다 훨씬 더 많은 상품을 보유하고 다양한 선택지를 제공하게 되므로 소비자의 수요가 늘고 있다. 또한, 소비자들에게 편리한 UI로 가격을 비교하고, 검색할 수 있게 하여 제품에 대한 리뷰나 별점 등을 확인하여 합리적인 소비가 가능하게 하도록 한다. 이러한 측면들에서 온라인쇼핑은 더욱 현대인들에게 매력적으로 다가오며, 수요가 늘어나고 있다.

4. 설계과제의 목표

- 웹서버 생성 : Linux Cent OS에 Node.js의 익스플로러 프레임워크 활용
- 로그인 기능 : 회원가입 시 데이터베이스 내 users 테이블에 user 정보 저장 및 password 암호화 후 저장 기능
- 상품 가격순 정렬 기능 : 낮은 가격순 / 높은 가격순 별 정렬 기능
- 네이버 쇼핑 API 데이터 처리 : MySQL에서 특성별로 테이블링 ; 카테고리별 저장
- 상품 정보 출력 기능 : 상품명, 가격, 이미지, 판매처명 출력 기능
- 상품 리다이렉션 기능 : 상품 클릭 시 네이버 쇼핑으로 리다이렉션 기능
- user 데이터 저장 : MySQL에서 관리, user password hash암호화 거친 후 후 저장
- 상품별 댓글 입력 : 댓글 입력 후 데이터베이스 내 comments 테이블에 저장 후 출력
- 상품별 평점 입력 : 평점 입력 후 데이터 베이스 내 ratings 테이블에 저장 후 출력

표 1. 본 프로젝트의 현실적 제한요소 항목

| 현실적 제한 요소들 | 내 용 (Content) |
|------------|--|
| 경제 | <ul style="list-style-type: none"> - 오픈소스 서버를 이용하여 비용 절감 - 온라인 상에 무료로 배포하는 오픈 소스 API를 활용한 경제성 고려 - 클라이언트-Server 간의 통신 간 CPU, 메모리 사용량을 최소화 하는 합리적인 코드 작성 |
| 편리 | <ul style="list-style-type: none"> - 여러 사이트에서 일일이 검색할 필요 없이 한 군데에서 다양한 가격대를 가지는 제품을 한눈에 볼 수 있는 가시성 |
| 윤리 | <ul style="list-style-type: none"> - 개발제품이 불법 또는 도덕적 윤리적 타락화에 사용되지 않을 것. |
| 사회 | <ul style="list-style-type: none"> - 과소비, 충동구매 등 건전하지 않은 소비습관에 대한 고민 |

5. 설계과정

5.1 설계 기초이론

5.1.1 가상머신 Cent OS 구축 및 Server 설정

비용절감 및 시스템효율, 보안성, 유연성 측면에서 이점을 가진 Linux OS를 이용한다. Linux Cent OS는 오픈소스이므로 비용절감이 가능하고, 실무에서도 가장 많이 쓰이는 OS이다. 또한, Linux는 적은 리소스를 사용해 시스템을 효율적으로 운영할 수 있고, Linux는 root계정 이외의 접근이 제한되고, 파일 구조가 복잡하므로 보안성 측면에서 이점을 가진다. 마지막으로, 오픈 소스이기 때문에 필요에 따라 관리자가 운영체제를 직접 수정하고 서비스를 필요한 것만 설치/실행시킬 수 있어 유연성이 뛰어나다.

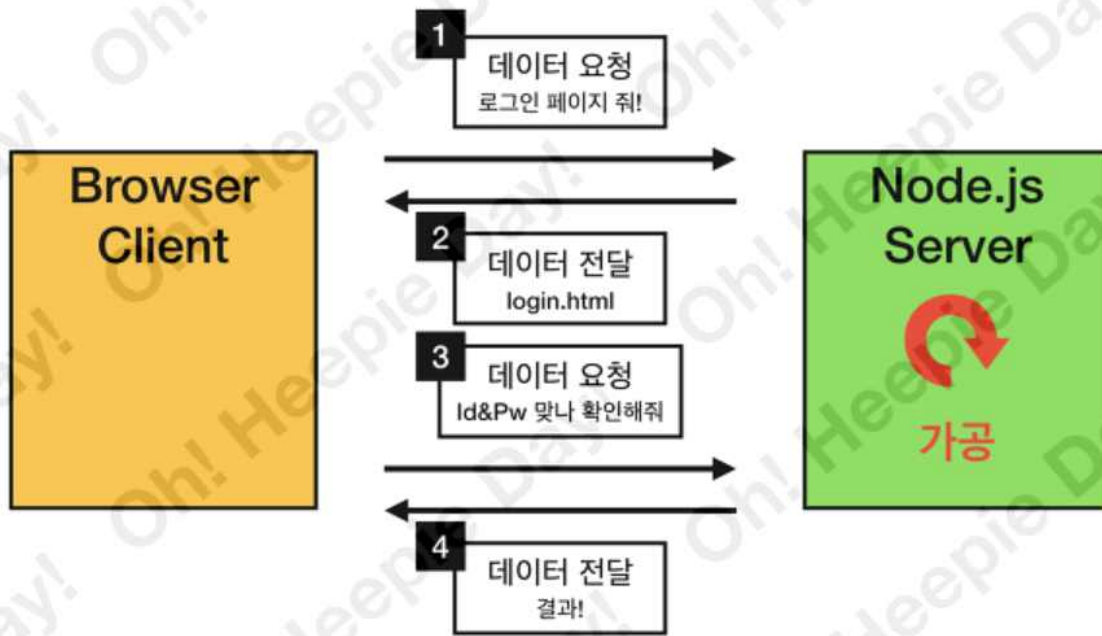
위와 같은 이유로 가상머신(Virtual Box)를 이용하여 Linux CentOS Server를 구축하고, 작업에 용이하도록 포트포워딩하여 원격접속한다. Linux 명령어를 이용하여 node와 MySQL, 필요로 하는 npm패키지들을 설치한다.

5.1.2 Server-Client 모델 및 네트워크 기초

서비스를 제공하는 Server와 Server에 서비스를 요청(request)하는 Client로 구성되어 있으며, Client의 요청(request)과 이에 Server가 요청(request)을 처리하고 응답(response)하는 형식으로 동작한다. 즉, Client의 요청에 의해서만 통신이 시작된다. Server는 Client의 요청을 항상 대기하기 위해 켜져 있어야 한다.

Server-Client 간 통신에서 포트번호는 서버 내에서 프로세스를 구분하는 번호로, 설

정한 웹서버 포트에 따라 접속이 가능하다. 일반적으로 http 서버는 80번 포트를 사용하나(Well-known Port), 보안을 강화하기 위해 3000번 포트를 임의 지정하여 사용하였다. 또한, Host OS에서 Guest OS 내 MySQL 서버로 Workbench 원격접속을 위해 3306 포트를 사용하며, 원격접속(Putty 이용, ssh)을 위해 22번 포트를 사용한다. 이 때, Cent OS 방화벽 포트를 개방한다.

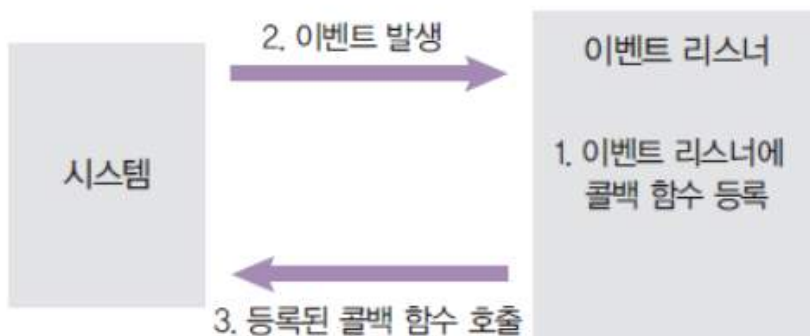


5.1.3 Node.js 개념 및 동작

Node는 Javascript Runtime으로 js로 만든 프로그램을 실행할 수 있게 해주는 가상 머신이다. Node는 libuv를 내부적으로 포함하고 있어 Node의 특징인 이벤트 기반, 논블로킹 I/O 모델을 구현한 라이브러리를 내장한다.

Node.js의 특징인 이벤트 기반은 이벤트가 발생할 때 미리 지정해둔 작업(콜백 함수)을 실행시키는 것으로, 이벤트 리스너를 통해 이벤트를 등록하면 해당 이벤트 발생(예) 마우스 클릭, 네트워크 요청 등)에 따라 콜백 함수가 호출되는 방식이다.

▼ 그림 1-4 이벤트 기반



이벤트 기반에서 이벤트 루프는 이벤트 발생 시 호출할 콜백 함수를 관리하고, 콜백 함수의 실행 순서를 (노드 종료 시까지) 결정하여 이벤트를 처리 한다. 태스크 큐는 이벤트 발생 후 호출되어야 할 콜백 함수가 대기하는 공간으로, 이벤트 루프가 정한 순서대로 대기한다. 이벤트 리스너들은 백그라운드에서 대기한다.

5.1.4 Express 구조와 pug 템플릿 엔진

Express는 Node.js 개발을 위한 웹개발 프레임워크로, express 모듈로부터 제공되는 Application은 Node.js의 http 모듈을 기반으로 동작하는 웹 애플리케이션으로, 서버의 핵심 역할을 수행한다.

Express는 미들웨어를 사용하여 요청(request)과 응답(response) 사이에서 처리할 작업을 정의한다. 미들웨어는 Express 애플리케이션에 연속적으로 적용되며, 요청을 처리하고 응답을 반환하는 중간 단계로 작동하며, app.use() 메서드를 사용하여 추가하고, 다양한 작업(라우팅, 로깅, 에러 처리 등)을 수행한다.

Express는 미들웨어의 일종인 라우터를 사용하여 특정 URL에 대한 요청을 처리합니다. 라우터는 요청 경로와 요청 방법(GET, POST, PUT, DELETE 등)에 따라 적절한 핸들러 함수를 호출하여 해당 요청을 처리한다. 라우터는 app.get(), app.post(), app.put(), app.delete() 등의 메서드를 사용하여 요청 경로와 요청 방법에 따라 적절하게 라우팅하여 서버-클라이언트 간 원활한 request, response를 제공한다.

const pug = require('pug'); app.set('view engine', 'pug');를 통해 view엔진을 pug 템플링 엔진으로 사용한다고 정의한다. 이후 view 폴더 안에 .pug파일이 생성되어 동적인 HTML을 구성할 수 있으며, res.render()를 통해 렌더링한다.

express.static() 미들웨어로 외부에 공유할 정적 파일을 클라이언트에 전달할 수 있으며, 서버 구조 파악이 힘들게 되어 보안성이 증대한다.

Express는 위와 같은 방식으로 request-response 사이클을 관리하고, 미들웨어와 라우터를 통해 application의 동작을 처리한다.

5.1.5 MySQL

서버 메모리에 데이터 저장 시 서버 재시작으로 인한 데이터 휘발 문제로 영구적으로 데이터를 저장할 공간인 데이터베이스를 사용하게 되었으며, MySQL은 관계형 데이터베이스(Relational Database Management System)로, 데이터를 테이블 형태로 저장, 관리, 검색한다. 본 프로젝트에서 사용할 Guest PC인 CentOS 가상머신은 물리 디스크로 구성되어 있어 데이터가 디스크에 기록되게 되고, root 사용자 이외에 oracle12 사용자를 추가해 원격(Host PC)에서 데이터베이스 서버에 3306포트로 접속 가능하게 하였다.

스키마(Schema)는 논리적인 구조를 정의하는 데이터베이스의 단위로, CREATE SCHEMA 쿼리문을 이용하여 생성 시 이 안에 데이터베이스 테이블을 생성할 수 있다. 테이블(Table)은 데이터 특성을 기준으로 구조화하여 데이터를 저장하는 단위로, 테이블의 열은 데이터 속성을 나타내고, 행은 실제 데이터를 저장한다.

5.2.1 소프트웨어 기능블록도 (개념설계)

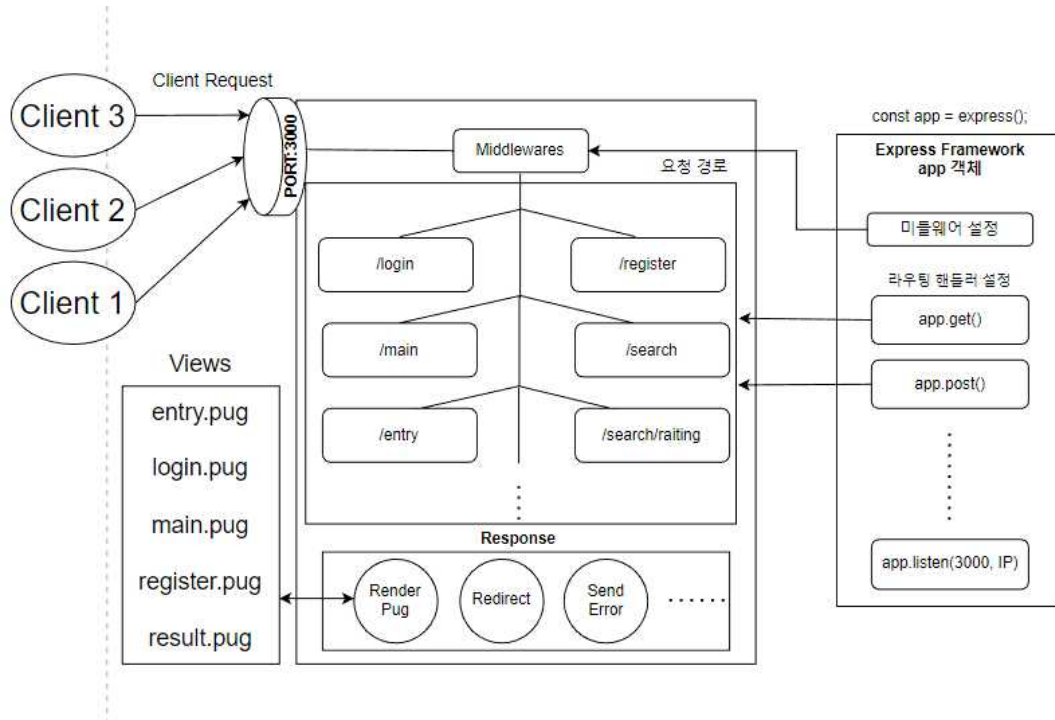


그림 1. 웹서버 구성도

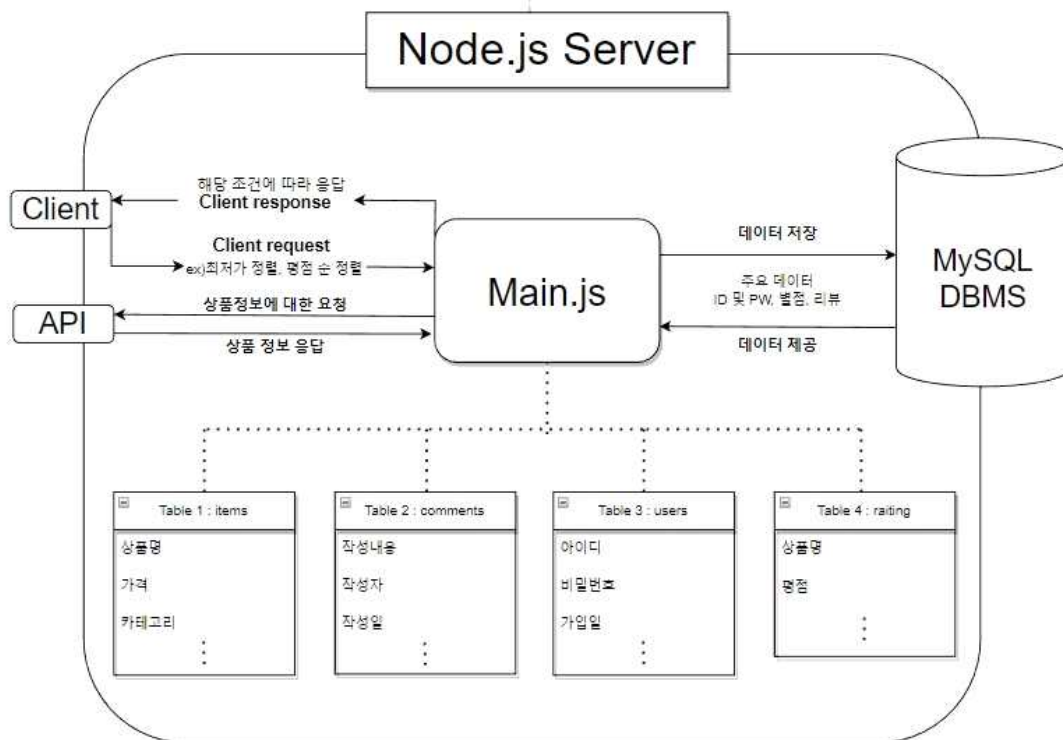
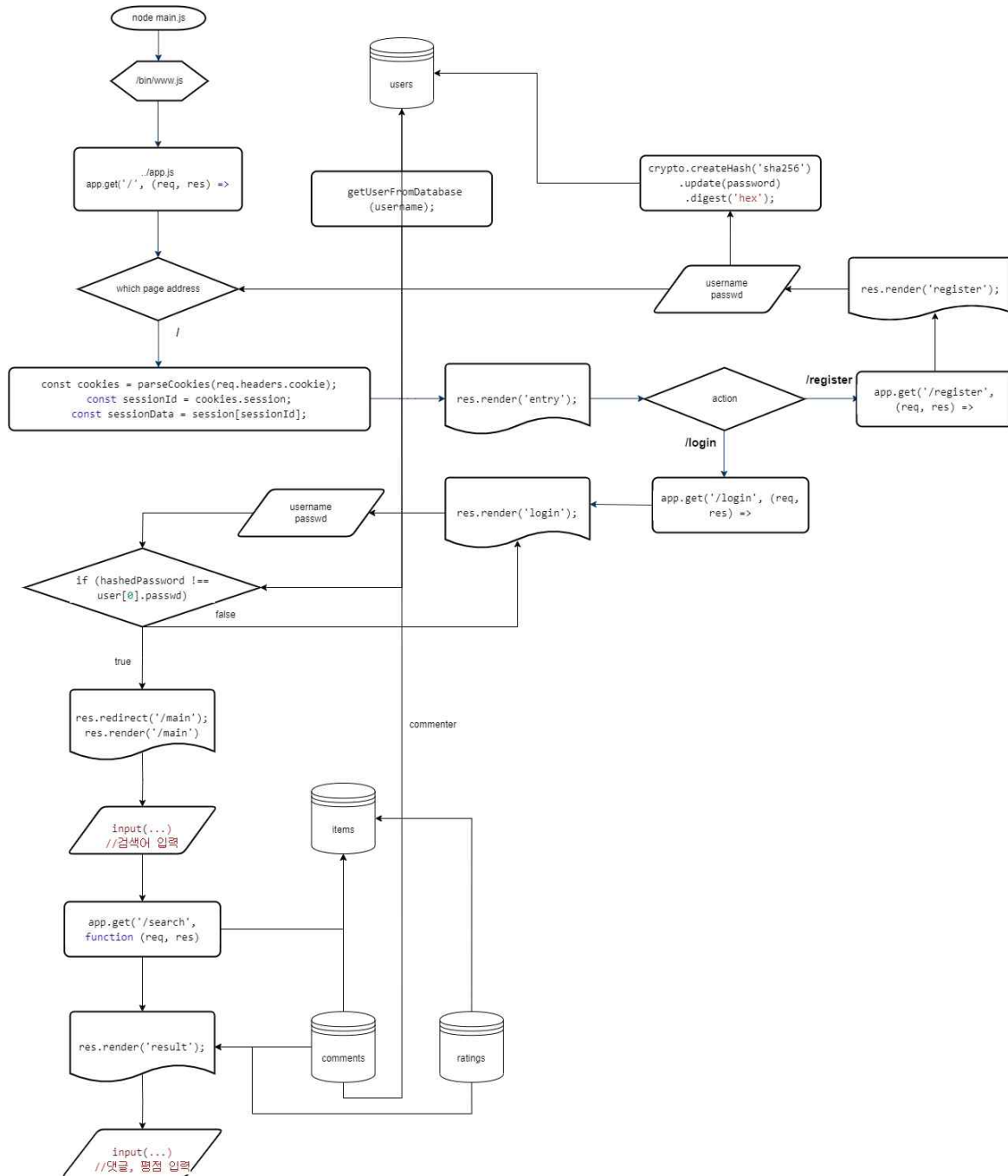


그림 2. MySQL DBMS에 적재된 데이터 테이블링 및 데이터 반환 구성도

5.2.2 기능적 요구사항 명세서

| ID | 요구사항 | 내용 | 설명 | 우선순위 |
|------|------------|--------------------------------|---|------|
| p_01 | 웹서버 생성 | Node.js 웹Server를 생성한다. | 가상 머신 이용, Linux Cent OS 상에 Node.js http 모듈, 익스프레스 프레임워크를 사용 | 1 |
| p_02 | DBMS 설치 | 데이터를 관리할 DBMS를 준비한다. | MySQL 설치 후 데이터들을 관리할 테이블을 생성 | 2 |
| p_03 | 외부 쇼핑몰 API | 다양한 외부 쇼핑몰 데이터 API를 활용한다. | 외부 쇼핑몰 API로 상품 데이터 MySQL에 적재 | 3 |
| p_04 | 사용자 평점 시스템 | 사용자가 각 상품에 대한 별점을 남길 수 있도록 한다. | 사용자가 각 상품에 대한 별점을 남기면 이를 MySQL 별점 테이블에 적재 | 4 |
| p_05 | 사용자 댓글 시스템 | 사용자가 각 상품에 대한 댓글을 남길 수 있도록 한다. | 사용자가 각 상품에 대한 댓글을 남기면 이를 MySQL 댓글 테이블에 적재 | 5 |
| p_06 | 편리한 UI 설계 | 사용자들이 이용하기 편하도록 UI설계 | 상품을 카테고리별로 분류하여 DBMS 테이블 별로 출력 | 6 |

5.2.3 전체 순서도 (상세설계)



6. 제작(Implementation)

6.1 제작과정

6.1.1 쿠키 파싱 함수

```

23 // 쿠키 파싱 함수
24 const parseCookies = (cookie = '') => cookie
25   .split(';')
26   .map(v => v.split('='))
27   .reduce((acc, [k, v]) => {
28     acc[k.trim()] = decodeURIComponent(v);
29     return acc;
30   }, {});

```

문자열 형태로 전달된 쿠키(cookie) 값을 파싱하여 객체 형태로 반환한다.

6.1.2 DB 연결 설정 및 연결

```

const connection = mysql.createConnection({
  host: 'localhost', // 데이터베이스 호스트
  user: 'root', // 데이터베이스 사용자 이름
  password: 'Mysql123!@#', // 데이터베이스 비밀번호
  database: 'nodejs_shoppingmall' // 데이터베이스 이름
});
connection.connect(); // 데이터베이스 연결

```

데이터베이스 연결에 필요한 정보 입력 후 웹서버를 MySQL서버에 연결한다.

6.1.3. main

```

46 // 메인 페이지
47 app.get('/main', function (req, res) {
48   console.log("IP: " + req.ip + " / 접속");
49   const cookies = parseCookies(req.headers.cookie);
50   const sessionId = cookies.session;
51   const sessionData = session[sessionId];
52
53   if (!sessionData) {
54     res.redirect('/'); // 세션 데이터가 없으면 로그인 페이지로 리다이렉션
55     return;
56   }
57
58   res.render('main', {
59     title: 'Main',
60     username: sessionData.username
61   });
62 });

```

/main에 대한 요청이 들어올 시 쿠키를 받아오고 세션데이터가 없으면 로그인 페이지로 리다이렉션한다. 이후 main.pug를 렌더링한다.

6.1.4. search

/search에 대한 요청이 들어올 시 쿠키의 query중 검색어를 따로 떼어낸다. API 요청에 필요한 옵션인 options객체에 정보 입력 후 이를 사용해 네이버 API를 통하여 해당 상품 정보를 받아온다. DB서버에서는 각각의 상품에 대한 평점, 리뷰를 받아오고 사용자가 원하는 조건(정확도, 가격 높은순, 가격 낮은순)에 따라 배열을 정리하고 result.pug를 렌더링한다. 이후 검색 정보와 클라이언트의 주소를 콘솔에 출력한다.

6.1.5. register

```

158 // 회원 가입 처리
159 app.post('/register', (req, res) => {
160   const body = [];
161   req
162     .on('data', (chunk) => {
163       body.push(chunk);
164     })
165     .on('end', () => {
166       const data = Buffer
167         .concat(body)
168         .toString();
169       const {username, password} = qs.parse(data);
170
171       getUserFromDatabase(username)
172         .then((user) => {
173           if (user.length > 0) {
174             res.send(
175               "<script>alert('이미 존재하는 사용자 이름입니다.');" + window.location.href + '/' + "</script>"
176             );
177             return;
178           }
179
180           // 회원 정보 저장
181           const hashedPassword = crypto
182             .createHash('sha256')
183             .update(password)
184             .digest('hex');
185           const newUser = {
186             username,
187             password: hashedPassword
188           };
189
190           saveUserToDatabase(newUser)
191             .then(() => {
192               res.send(
193                 "<script>alert('회원 가입이 완료되었습니다.');" + window.location.href + '/' + "</script>"
194               );
195             })
196             .catch((error) => {
197               console.error(error);

```

/register에 대한 요청이 들어올 시 입력받은 data 문자열에 qs.parse() 함수를 사용하여 username과 password를 떼어낸다. 이후 getUserFromDatabase(username) 함수를 사용하여 데이터베이스에서 동일한 사용자 이름을 가진 사용자를 조회하고 존재하는 경우 이미 존재하는 사용자임을 알리는 경고창을 보낸 후 메인페이지로 리다이렉션한다. 존재하지 않을 경우 비밀번호를 해시함수를 통해 암호화 한 후 아이디와 함께 DB에 저장한다. 이후 메인페이지로 리다이렉션한다.

6.1.6. comment

```

212 // 댓글 추가 처리
213 app.get('/search/comment', (req, res) => {
214     const item = req.query.itemName; // 상품 이름을 파라미터로 받음
215     const comment = req.query.comment; // 클라이언트에서 전송된 댓글 내용
216     const commenter = user_ID;
217     // 댓글을 데이터베이스 또는 다른 저장소에 추가
218     saveCommentToDatabase(item, comment, commenter); // 데이터베이스에 댓글 저장하는 함수 호출
219
220     // 댓글 목록을 클라이언트로 전송
221     const redirectURL = '/search?query=' + encodeURIComponent(onSearch);
222     res.redirect(redirectURL);
223 });

```

/search/comment에 대한 요청이 들어올 시 query에서 상품 이름과 댓글 내용을 추출하여 이를 작성자의 ID와 함께 저장하고 saveCommentToDB함수를 호출해서 DB에 이에 대한 정보를 기록, 이후 상품 검색 결과가 있는 페이지로 리다이렉션한다.

6.1.7. rating

```

225 // 평점 추가 처리
226 app.get('/search/rating', (req, res) => {
227     const item = req.query.itemName; // 상품 이름을 파라미터로 받음
228     const rating = req.query.rating; // 클라이언트에서 전송된 평점
229
230     saveRatingToDatabase(item, rating); // 데이터베이스에 평점 저장하는 함수 호출
231
232     const redirectURL = '/search?query=' + encodeURIComponent(onSearch);
233     res.redirect(redirectURL);
234 });

```

/search/rating에 대한 요청이 들어올 시 query에서 상품 이름과 별점을 추출한다. 이후 saveRatingToDB함수를 호출해서 DB에 이에 대한 정보를 기록, 이후 상품 검색 결과가 있는 페이지로 리다이렉션한다.

6.1.8. login

```

236 // 로그인 처리
237 app.post('/login', (req, res) => {
238   const body = [];
239   req
240     .on('data', (chunk) => {
241       body.push(chunk);
242     })
243     .on('end', () => {
244       const data = Buffer
245         .concat(body)
246         .toString();
247       const {username, password} = qs.parse(data);
248       user_ID = username;
249
250       // 회원 정보 저장
251       const hashedPassword = crypto.createHash('sha256').update(password).digest('hex');
252
253       getUserFromDatabase(username)
254         .then((user) => {
255           // 검색된 사용자 정보를 이용하여 작업 수행 로그인 실패 처리
256           if (user.length === 0 || hashedPassword !== user[0].passwd) {
257             res.send("<script>alert('로그인 실패'); window.location.href='/login';</script>");
258             return;
259           }
260
261           const expires = new Date();
262           expires.setMinutes(expires.getMinutes() + 10);
263           const uniqueInt = Date.now();
264           session[uniqueInt] = {
265             username,
266             expires
267           };
268
269           // 세션 정보를 쿠키에 저장
270           res.cookie('session', uniqueInt, {
271             expires: expires,
272             httpOnly: true,

```

/login에 대한 요청이 들어올 시 data 문자열에 qs.parse() 함수를 사용하여 username과 password를 떼어낸다. 이후 해시 알고리즘을 통해 사용자의 비밀번호를 해시화한다. getUserFromDB함수를 사용해서 DB에서 사용자를 찾고, 사용자가 검색되지 않거나 비밀번호가 일치하지 않는 경우 로그인실패 메시지를 보내고 로그인 페이지로 리다이렉션한다. 사용자가 인증되었을 시 세션을 생성하고 정보를 쿠키에 저장한다. 이후 메인페이지로 리다이렉션한다.

6.1.9. 페이지 처리 및 서버 실행

```
288 // 진입 페이지 처리
289 app.get('/', (req, res) => {
290   const cookies = parseCookies(req.headers.cookie);
291   const sessionId = cookies.session;
292   const sessionData = session[sessionId];
293
294   res.render('entry');
295 });
296
297 // 로그인 페이지 처리
298 app.get('/login', (req, res) => {
299   res.render('login');
300 });
301
302 // 회원 가입 페이지 처리
303 app.get('/register', (req, res) => {
304   res.render('register');
305 });
306
307 // 서버 실행
308 app.listen(3000, function () {
309   console.log('server is running');
310 });
```

/ 경로로 요청이 들어오면 쿠키로부터 세션ID를 추출하고 데이터를 가져온다. 이후 entry.pug를 렌더링한다.

/login 경로로 요청이 들어오면 login.pug를 렌더링한다.

/register 경로로 요청이 들어오면 register.pug를 렌더링한다.

이후 3000번 포트에서 서버를 실행하고 콘솔에 “서버 실행 중” 메시지를 출력한다.

6.1.10. database.js

```
const mysql = require('mysql');

const connection = mysql.createConnection({
  host: 'localhost', // centos 서버 아이피
  user: 'root', // oracle12
  password: 'Mysql123!@#', // Mysql123!@#
  database: 'nodejs_shoppingmall' // nodejs_shoppingmall
});

connection.connect();

function getUserFromDatabase(username) {
  return new Promise((resolve, reject) => {
    // 데이터베이스에서 아이템을 검색하기 위한 쿼리
    const query = `SELECT * FROM users WHERE username = ?`;

    // 쿼리 실행
    connection.query(query, [username], (error, results, fields) => {
      if (error) {
        reject(error);
        return;
      }

      // 검색된 데이터를 JavaScript 변수에 저장
      const user = results.map((row) => {
        return {username: row.username, nickname: row.nickname, comment: row.comment,
          created_at: row.created_at, passwd: row.passwd};
      });

      resolve(user);
    });
  });
}

function saveUserToDatabase(users) {
  return new Promise((resolve, reject) => {
    // 변수 값을 MySQL 데이터베이스에 저장
    const query = `INSERT INTO users (username, passwd) VALUES (?, ?)`;
    const values = [users.username, users.password];
    connection.query(query, values, (error, result) => {
      if (error) {
        reject(error);
        return;
      }
      console.log('user saved successfully');
      resolve();
    });
  });
}
```

웹서버 내에서 MySQL 서버에 연결하여 데이터를 적재하고, 조회하는 함수를 선언하여 사용자 요청에 따라 데이터베이스로부터의 데이터 교환을 수행하도록 한다.

6.2 제작시 문제점 및 개선사항

리눅스 파일구조를 보안성 있게 관리하고 싶었지만, 시간 관계상 그러지 못한 점이 문제가 될 수 있다. 이런 문제점을 보완하고자 기본 http포트 80번을 3000번의 특정 포트를 사용하도록 하여 보안성을 유지하였다.

6.3 지도교수 지도내용 및 지적사항 조치결과

| 일시 | 내용 | 피드백 |
|------------|--------------------------------------|--|
| 5/25 13:00 | MySQL 테이블링에 따른 데이터 실제 적재 과정 설계 방법 | Javascript 내에서 MySQL서버 접속 및 쿼리문 사용 가능 |

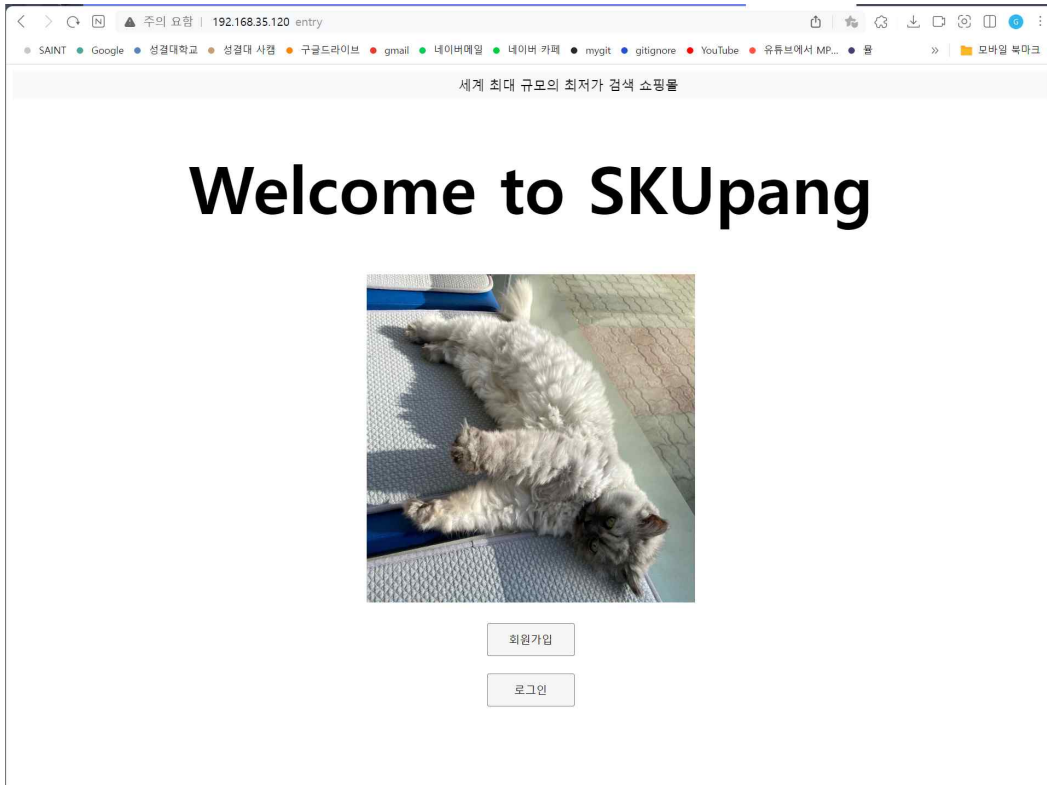
7. 시험

```
[root@localhost webserver_shoppingmall]# git pull origin master
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 5 (delta 4), reused 5 (delta 4), pack-reused 0
Unpacking objects: 100% (5/5), done.
From https://github.com/znantothelimit/webserver_shoppingmall
* branch      master      -> FETCH_HEAD
Updating 42ecd58..122f4b7
Fast-forward
 README.md      | 1 -
 webserver_shoppingmall/main.js | 2 +-
 2 files changed, 1 insertion(+), 2 deletions(-)
[root@localhost webserver_shoppingmall]#
```

root@localhost:var/www/webserver_shoppingmall/webserver_shoppingmall

```
[root@localhost webserver_shoppingmall]# ls -al
total 84
drwxr-xr-x.  7 root root  174 Jun  6 14:48 .
drwxr-xr-x.  5 root root  139 Jun  6 14:48 ..
drwxr-xr-x.  2 root root   17 Jun  6 03:10 bin
-rw-r--r--.  1 root root 7040 Jun  6 12:25 database.js
-rw-r--r--.  1 root root 10680 Jun  6 14:48 main.js
drwxr-xr-x. 138 root root 4096 Jun  6 07:34 node_modules
-rw-r--r--.  1 root root  427 Jun  6 03:11 package.json
-rw-r--r--.  1 root root 47202 Jun  6 07:34 package-lock.json
drwxr-xr-x.  3 root root   25 Jun  6 03:10 public
drwxr-xr-x.  2 root root   38 Jun  6 03:10 routes
-rw-r--r--.  1 root root  329 Jun  6 06:11 system.js
drwxr-xr-x.  2 root root   94 Jun  6 14:45 views
[root@localhost webserver_shoppingmall]# node main.js
http://192.168.35.120:3000/ app listening on port 3000!
```

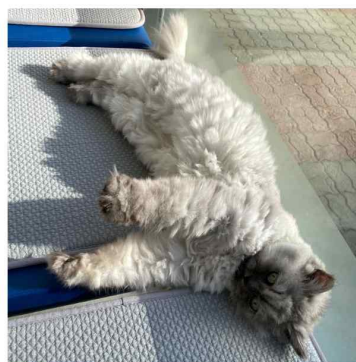
가상머신 Cent OS 내로 git clone -> git pull을 통해 코드를 가져온 다음, 웹서버를 실행시킴



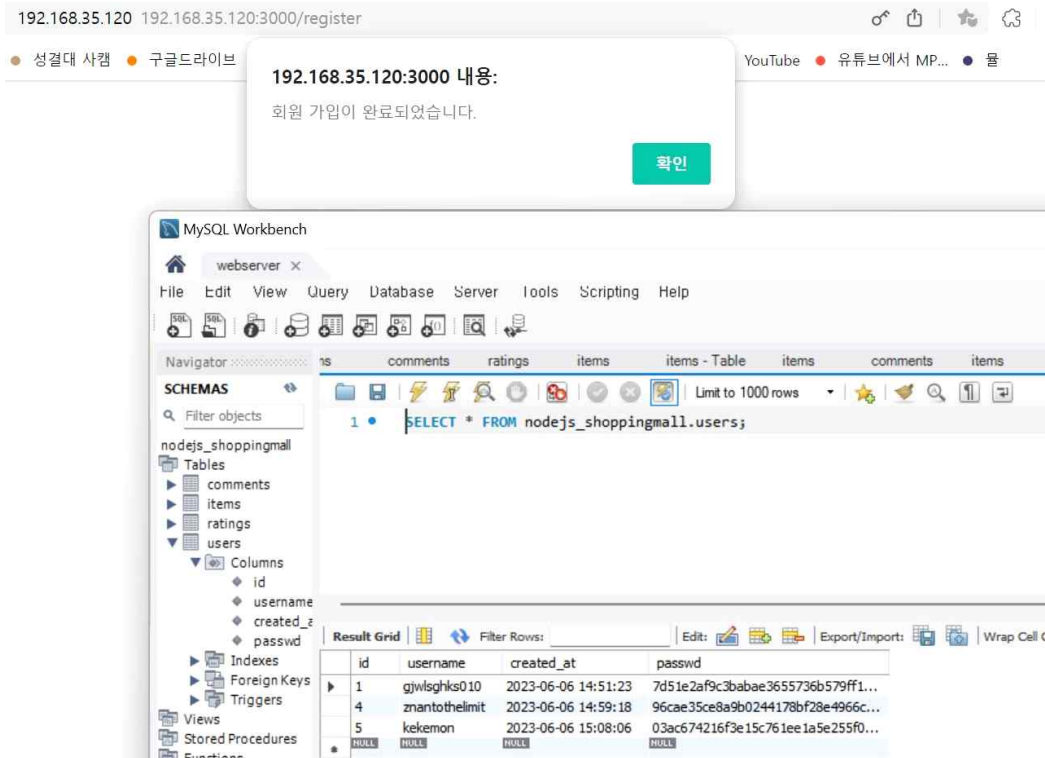
인덱스 경로로 진입했을 시 entry.pug로 랜더링되어 사용자에게 출력된다.



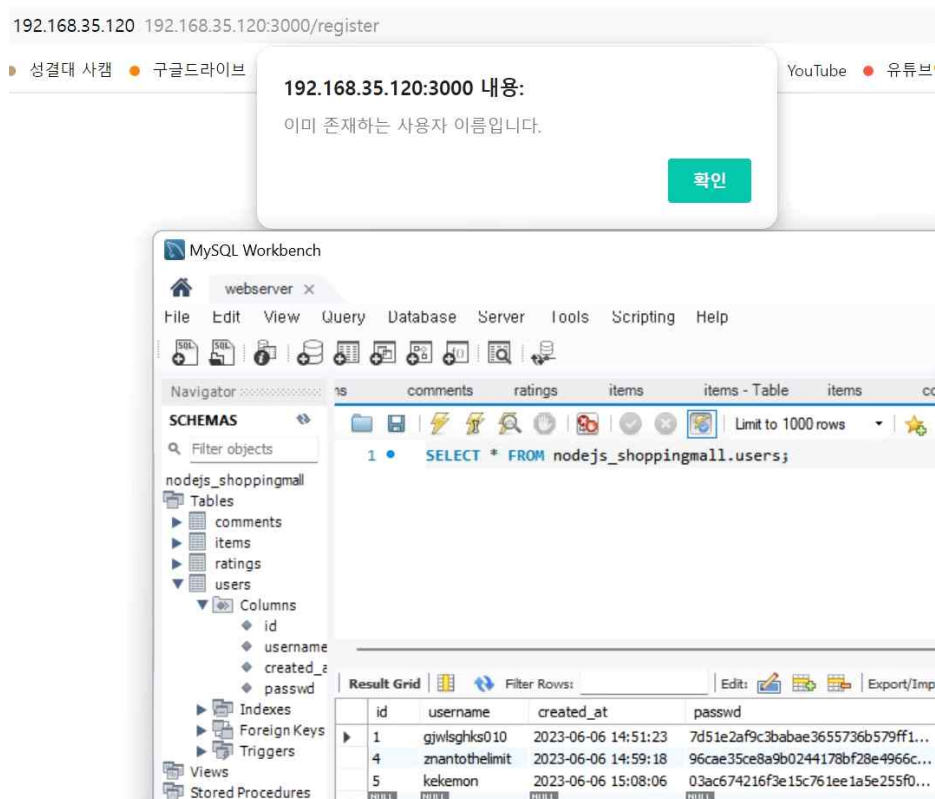
회원가입



회원가입 버튼 클릭시 register.pug가 랜더링되어 출력된다.



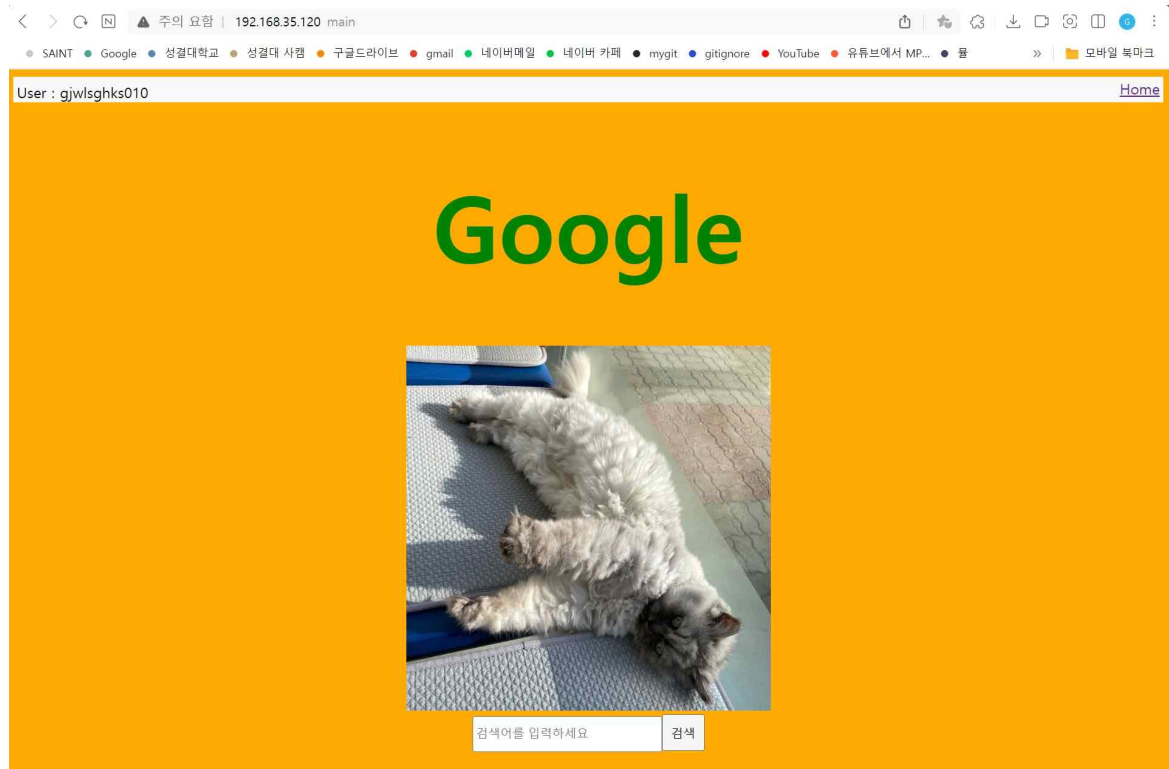
회원가입에 성공하면, 패스워드가 암호화되어 사용자명과 함께 데이터베이스 users 테이블에 저장된다.



이미 존재하는 사용자명일 경우, 로그인 실패 및 데이터베이스에 저장되지 않는다.



로그인 시 입력한 패스워드를 hash 암호화하여 데이터베이스에 암호화되어 저장된 암호와 비교한다. 같은 값을 가지지 않는 경우 위와 같이 로그인 실패한다.



로그인에 성공할 경우, main.pug로 랜더링되어 사용자에게 출력된다.

| | id | price | category | name | stored_at | link | image | mallName |
|--------------------|----|---------|----------|---|---------------------|--|--|---------------|
| ▶ Items | 3 | 364800 | 생활/건강 | 아미와 파시파카 112M 112V 울격티카... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_84637... | 콘보트리 |
| ▼ Columns | 4 | 359900 | 생활/건강 | 스모어 스포츠복스울격티카S...... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_83114... | 악기장용품 종합악기 소품 |
| id | 5 | 285000 | 생활/건강 | 스모어 E100 배낭 울격울격티카... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_83866... | 악기장용품 |
| price | 6 | 285000 | 생활/건강 | 텍스 입문용 울격티카E100 ... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_20792... | 악기장용품 |
| category | 7 | 249000 | 생활/건강 | 루트스 울격티카GX-100/전기 기타/... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_83953... | 휴직 생활 |
| ▼ Triggers | 8 | 310000 | 생활/건강 | 텍스 입문용 울격티카E100 PLUS | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_38638... | 대미버 |
| ▼ users | 9 | 358000 | 생활/건강 | 코로나 Corona STANDARD ST OW Maple ... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_37627... | 대미버 |
| ▼ ratings | 10 | 268700 | 생활/건강 | 아미와 파시파카 울격티카Yamaha... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_39699... | 대미버 |
| ▼ Stored Procedure | 11 | 210000 | 생활/건강 | 플루트 G110 울격티카... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_56042... | 대미버 |
| ▼ Functions | 12 | 285000 | 생활/건강 | 텍스 울격티카스울격티카T180 T200 | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_35763... | 대미버 |
| Administration | 13 | 285000 | 생활/건강 | 바커스 울격티카BST-2 RSM ... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_34660... | 대미버 |
| Information | 14 | 290000 | 생활/건강 | 텍스 G250 | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_81087... | 대미버 |
| Table: | 15 | 279800 | 생활/건강 | 바커스 울격티카BST 2 RSM 프로크 ... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_82958... | 콘보트리 |
| Items: | 16 | 495000 | 생활/건강 | 길모어 로먼 빈티지 EX Gilmour MODERN VINTA... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_36989... | 대미버 |
| id | 17 | 198000 | 생활/건강 | 저는 울격 3 입문용 울격티카ON 백... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_85113... | 엔타보 |
| price | 18 | 489000 | 생활/건강 | 스모어 아미와 파시파카 울격티카... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_83098... | 악기장용품 종합악기 소품 |
| category | 19 | 493000 | 생활/건강 | 아미와 Gibson Custom 1985 Les Paul Custom... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_26721... | 대미버 |
| ▼ View | 20 | 3659000 | 생활/건강 | 삼파코 플루트 아미와 플루트 스... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_33749... | 기타의류 |
| ▼ Stored | 21 | 918000 | 생활/건강 | 텍스 울격티카G300 Raw Natural Satin | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_38986... | 대미버 |
| ▼ Procedure | 22 | 379990 | 생활/건강 | 텍스 입문용 울격티카W300 | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_37737... | 대미버 |
| ▼ Functions | 23 | 403200 | 생활/건강 | 코로나 로먼 스탠다드 울격티카... | 2023-06-06 14:54:42 | https://search.shopping.naver.com/gate/nhnh... | https://shopping-phinf.pstatic.net/main_35690... | 대미버 |

SAINT Google 성결대학교 성결대 사경 구글드라이브 gmail 네이버메일 네이버 카페 mygit gltignore YouTube 유튜브에서 MP... 물 모바일 북마크

검색어: 일렉기타 [다시 검색하러가기](#) [Home](#)

정확도 순 | **가격 높은 순** | 가격 낮은 순

실재고보유 / 펜더 아메리칸 울트라 스트라토캐스터 일렉 기타

카테고리: 생활/건강
 가격: 3659000
 판매처: 기타센터
[웹사이트 이동](#)

댓글 입력 댓글 달기

등록된 댓글이 없습니다.

등록된 평점이 없습니다. 1점 평점 남기기

PRS SE 2023 일렉기타 Custom 24 Floyd Charcoal

카테고리: 생활/건강
 가격: 1848000
 판매처: 네이버
[웹사이트 이동](#)

댓글 입력 댓글 달기

등록된 댓글이 없습니다.

등록된 평점이 없습니다. 1점 평점 남기기

펜더 재팬 하이브리드 2 스트라토캐스터 일렉 기타

카테고리: 생활/건강
 가격: 1599000
 판매처: 기타센터
[웹사이트 이동](#)

댓글 입력 댓글 달기

등록된 댓글이 없습니다.

등록된 평점이 없습니다. 1점 평점 남기기

야마하 일렉기타 퍼시픽 PAC-612VIIFM PAC612VIIFM

SAINT Google 성결대학교 성결대 사경 구글드라이브 gmail 네이버메일 네이버 카페 mygit gltignore YouTube 유튜브에서 MP... 물 모바일 북마크

검색어: 일렉기타 [다시 검색하러가기](#) [Home](#)

정확도 순 | **가격 높은 순** | 가격 낮은 순

튼튼한 일렉기타입문 일렉트릭 전자기타 배우기 세트

카테고리: 생활/건강
 가격: 109840
 판매처: 네이버
[웹사이트 이동](#)

댓글 입력 댓글 달기

등록된 댓글이 없습니다.

등록된 평점이 없습니다. 1점 평점 남기기

제논 일렉 GP-3 입문용 일렉기타 완벽셋팅후 출고

카테고리: 생활/건강
 가격: 198000
 판매처: 엔타브
[웹사이트 이동](#)

댓글 입력 댓글 달기

등록된 댓글이 없습니다.

등록된 평점이 없습니다. 1점 평점 남기기

클트 G110 일렉기타

카테고리: 생활/건강
 가격: 210000
 판매처: 네이버
[웹사이트 이동](#)

댓글 입력 댓글 달기

등록된 댓글이 없습니다.

등록된 평점이 없습니다. 1점 평점 남기기

갑슨 레스폴 DIY 일렉기타 커스텀 빈티지 스탠다드


순서대로 가격 높은순, 낮은 순을 클릭했을 때 나오는 결과이다.

검색어: 일렉기타

정확도 순 | 가격 높은 순 | 가격 낮은 순

야마하 퍼시픽카 112VM 112V 일렉기타

카테고리: 생활/건강
가격: 364800
판매처: 준보트리
[웹사이트 이동](#)



댓글 입력 | 댓글 달기

평가: 우와 이 기타 이쁘다
ID: gjwlsghks010, 작성일: 2023. 6. 7. 오전 3:58:04


평가: 인정 이쁘다
ID: gjwlsghks010, 작성일: 2023. 6. 7. 오전 3:58:11

평가: 저는 좀 별로인듯요
ID: znantothelimit, 작성일: 2023. 6. 7. 오전 3:59:38

★ : 3.67점 1점 | 평점 남기기

스콰이어 스트라토캐스터 일렉기타 SQUIER BULLET STRAT HT AWT

카테고리: 생활/건강
가격: 359000
판매처: 악기정류장 종합악기 소품몰
[웹사이트 이동](#)



댓글 입력 | 댓글 달기

평가: 스콰이어가 좋저가 근본이지
ID: gjwlsghks010, 작성일: 2023. 6. 7. 오전 3:58:40

평가: 저도 인정 근데 색깔이 때가 탈 것 같아서 1점 드려요
ID: znantothelimit, 작성일: 2023. 6. 7. 오전 3:59:53

★ : 3.00점 1점 | 평점 남기기

위와 같이 서로 다른 사용자로 댓글과 평점을 남기면, 아래와 같이 댓글은 데이터베이스 comments 테이블에, 평점은 ratings 테이블에 각각 저장되고, 출력될 때 여기서 꺼내와진다.

comments

- Columns
 - id
 - commenter
 - comment
 - created_at
 - item_name
- Indexes
- Foreign Keys
- Triggers
- items
- ratings
- users
- Views
- Stored Procedures
- Functions

| | id | commenter | comment | created_at | item_name |
|---|----|----------------|--------------------------------|---------------------|---------------------------------|
| ▶ | 1 | gjwlsghks010 | 우와 이 기타 이쁘다 | 2023-06-06 14:58:04 | 야마하 퍼시픽카 112VM 112V 일렉기타 |
| | 2 | gjwlsghks010 | 인정 이쁘다 | 2023-06-06 14:58:11 | 야마하 퍼시픽카 112VM 112V 일렉기타 |
| | 3 | gjwlsghks010 | 스콰이어가 좋저가 근본이지 | 2023-06-06 14:58:40 | 스콰이어 스트라토캐스터 일렉기타 S... |
| | 4 | znantothelimit | 저는 좀 별로인듯요 | 2023-06-06 14:59:38 | 야마하 퍼시픽카 112VM 112V 일렉기타 |
| | 5 | znantothelimit | 저도 인정 근데 색깔이 때가 탈 것 같아서 1점 ... | 2023-06-06 14:59:53 | 스콰이어 스트라토캐스터 일렉기타 S... |
| | 6 | znantothelimit | 베이스는 기타로 안칩니다 | 2023-06-06 15:02:20 | 렉스 B100 일문용 베이스 기타 |

comments

- Columns
 - id
 - commenter
 - comment
 - created_at
 - item_name
- Indexes
- Foreign Keys
- Triggers
- items
- ratings
- users
- Views
- Stored Procedures
- Functions

| | id | commenter | comment | created_at | item_name |
|---|----|----------------|--------------------------------|---------------------|---------------------------------|
| ▶ | 1 | gjwlsghks010 | 우와 이 기타 이쁘다 | 2023-06-06 14:58:04 | 야마하 퍼시픽카 112VM 112V 일렉기타 |
| | 2 | gjwlsghks010 | 인정 이쁘다 | 2023-06-06 14:58:11 | 야마하 퍼시픽카 112VM 112V 일렉기타 |
| | 3 | gjwlsghks010 | 스콰이어가 좋저가 근본이지 | 2023-06-06 14:58:40 | 스콰이어 스트라토캐스터 일렉기타 S... |
| | 4 | znantothelimit | 저는 좀 별로인듯요 | 2023-06-06 14:59:38 | 야마하 퍼시픽카 112VM 112V 일렉기타 |
| | 5 | znantothelimit | 저도 인정 근데 색깔이 때가 탈 것 같아서 1점 ... | 2023-06-06 14:59:53 | 스콰이어 스트라토캐스터 일렉기타 S... |
| | 6 | znantothelimit | 베이스는 기타로 안칩니다 | 2023-06-06 15:02:20 | 렉스 B100 일문용 베이스 기타 |

8. 평가

8.1 정량적/정성적 목표달성도 평가

| 항목 | 목표값 | 달성률 (%) | 비고 |
|-------------------|--|---------|----|
| 웹서버 생성 | Linux Cent OS에 Node.js의 익스플로러 프레임워크 활용 | 100 % | . |
| 로그인 기능 | 회원가입 시 데이터베이스 내 users 테이블에 user 정보 저장 및 password 암호화 후 저장 기능 | 100 % | . |
| 상품 가격순 정렬 기능 | 낮은 가격순 / 높은 가격순 별 정렬 기능 | 100 % | . |
| 네이버 쇼핑 API 데이터 처리 | MySQL에서 특성별로 테이블링 ; 카테고리별 저장 | 100 % | . |
| 상품 정보 출력 기능 | 상품명, 가격, 이미지, 판매처명 출력 기능 | 100 % | . |
| 상품 리다이렉션 기능 | 상품 클릭 시 네이버 쇼핑으로 리다이렉션 기능 | 100 % | . |
| user 데이터 저장 | MySQL에서 관리, user password hash암호화 거친 후 후 저장 | 100 % | . |
| 상품별 댓글 입력 | 댓글 입력 후 데이터베이스 내 comments 테이블에 저장 후 출력 | 100 % | . |
| 상품별 평점 입력 | 평점 입력 후 데이터 베이스 내 ratings 테이블에 저장 후 출력 | 100 % | . |

8.2 현실적 제한요소 달성도 평가

| 현실적 제한 요소들 | 목표 | 달성결과 |
|------------|---|---|
| 경제 | <ul style="list-style-type: none"> - 오픈소스 서버를 이용하여 비용 절감 - 온라인 상에 무료로 배포하는 오픈소스 API를 활용한 경제성 고려 - 클라이언트-Server 간의 통신 간 CPU, 메모리 사용량을 최소화 하는 합리적인 코드 작성 | <ul style="list-style-type: none"> - MySQL을 활용해서 데이터를 합리적으로 적재할 수 있다. - 무료로 배포하는 네이버 쇼핑 오픈API를 활용해서 경제성을 고려했다. - 합리적인 쇼핑을 장려함으로써 메모리 사용뿐만 아니라 실제 경제사회에도 도움이 되도록 하였다. |
| 편리 | <ul style="list-style-type: none"> - 여러 사이트에서 일일이 검색할 필요 없이 한 군데에서 다양한 가격대를 가지는 제품을 한눈에 볼 수 있는 가시성 | <ul style="list-style-type: none"> - 상품 구매사이트와 바로 연결할 수 있는 하이퍼링크와 리뷰, 별점 등을 통해 간단하게 물건을 비교할 수도 있다. |
| 윤리 | <ul style="list-style-type: none"> - 개발제품이 불법 또는 도덕적 윤리적 타락화에 사용되지 않을 것. | <ul style="list-style-type: none"> - 유저 데이터를 암호화해서 보관하기 때문에 윤리적 문제를 줄였다. |
| 사회 | <ul style="list-style-type: none"> - 과소비, 충동구매 등 건전하지 않은 소비습관에 대한 고민 | <ul style="list-style-type: none"> - 정확도, 가격별로 데이터를 보여줌으로 인해 합리적인 소비가 가능하다. |

8.3 기능적 요구사항 달성도 평가

| ID | 요구사항 | 내용 | 우선 순위 | 달성률 (%) | 비고 |
|------|------------|--|-------|---------|----------------------------------|
| P_01 | 웹서버 생성 | Node.js 웹서버를 생성했고 클라이언트와 정상적으로 연결이 되었다. | 1 | 100% | |
| P_02 | DBMS 설치 | MySQL DBMS를 사용해서 데이터를 테이블에 맞게 정리하고 서버와 요청 및 응답이 정상적으로 이루어졌다. | 2 | 100% | |
| P_03 | 외부 쇼핑물 API | 네이버 쇼핑 API를 활용해서 프로젝트 과정에 필요로 하는 속성을 가져와서 재구성하였다. | 3 | 80% | 타 API는 사업자등록을 해야만 코드를 부여하는 등의 제한 |
| P_04 | 사용자 평점 시스템 | 사용자가 각 상품에 대한 별점을 남길 수 있도록 한다. | 4 | 100% | |
| P_05 | 자체 상품 랭킹 | 정확도, 가격 높은 순, 가격 낮은 순으로 검색 결과를 보여주었다. | 5 | 90% | |
| P_06 | 편리한 UI 설계 | 홈 버튼, 재검색 버튼 등을 이용해서 편리함을 높였고 배너, 애니메이션 등을 사용해 깔끔하게 설계했다. | 6 | 90% | |

8.2 문제점 및 해결방안

1주차) 설계 목표 설정 및 팀원 간 역할 분담, 전체적인 구성 및 방법 결정

설계 목표는 조원들과 토론하며 결정했고, 팀장인 허진환 학우는 전체 총괄 및 웹서버 구축/ DB관리를 하기로 했고 김태현 학우는 메인 코딩, UI 설계를 맡았고, 정재호 학우는 자료조사 및 API 활용에 대한 부분을 담당하기로 했다.

2주차) 소프트웨어 요구사항 파악 및 웹서버 생성

웹서버를 여는 과정에서 포트포워딩에 문제가 생겨서 메인 서버를 담당한 PC에 접근을 하지 못해서 각자 PC의 localhost를 사용해서 공유받은 코드로 실행을 하는 방식으로 진행하기로 했다.

3주차) 웹서버 메인 코딩 및 외부API 사용

여러 외부 API를 받아오려고 했지만 네이버 쇼핑 API를 제외한 대부분의 API가 해당 API를 사용하기 위한 코드를 받아오기 위해 사업자등록 번호가 필요하다고 하여 다양한 API 활용에 제한이 생겼다. 이는 API를 받아오기만 하면 바로 처리할 수 있도록 준비를 마쳐서 코드만 있으면 언제라도 적용할 수 있게 했다.

4주차) 데이터베이스 상품 적재 및 테이블링

DB접근도 포트포워딩의 문제로 각자의 PC에서 Table의 속성을 모두 통일한 뒤 진행했다. MySQL 테이블링에 따른 데이터 실제 적재 과정 설계 방법에 대한 고민이 있었지만 Javascript 내에서 MySQL서버 접속 및 쿼리문 사용 가능하다는 지도 교수님의 상담내용을 바탕으로 해결했다.

5주차) 웹서버 보안 설계, 디버그, 시험 및 검증

보안 문제를 해결하기 위해 ID와 PW에 관한 부분을 POST메서드로 처리함으로서 보이지 않게 했고 PW를 데이터베이스 서버로 넘기는 과정에 Hash함수를 통한 암호화를 진행하면서 서버의 보안에 대한 안정성을 높였다.

9. 추진체계

팀장(허진환) : 보고서 작성, 가상머신 환경 설정 및 웹서버 구축/DB 관리

팀원(김태현) : 회의록 작성, Main/Additional Function/UI Programming

팀원(정재호) : 자료조사, API/UI Programming

10. 설계 추진 일정: 2023년 5월 ~ 2023년 6월

| 수행 내용 | | 일정 (1주 단위) | | | | | |
|-----------------|---|------------|----|----|----|----|----|
| | | 10 | 11 | 12 | 13 | 14 | 15 |
| 목표와 기준 설정 | - 설계목표 설정 - 팀원 간 역할 분담 | | | | | | |
| | - 기능 블록 구성 - 기능별 구현방법 결정 - 소프트웨어 요구사항 파악 | | | | | | |
| 합성 | - 순서도 작성 - 목표 달성 가능성 확인 - 웹Server 생성 - UI 설계 | | | | | | |
| | | | | | | | |
| 분석 | - 웹서버 메인 코딩 - UI 구현 | | | | | | |
| | | | | | | | |
| 제작 | - 데이터 베이스 내 상품 데이터 적재 및 테이블링 - 웹서버 추가 기능 코딩 | | | | | | |
| | | | | | | | |
| 시험/ 평가 | - 시험 및 검증 - Trouble shooting - 재설계 | | | | | | |
| | | | | | | | |
| 결과 | - 결과보고 및 시연 | | | | | | |
| | | | | | | | |

11. 결론

웹서버를 생성하는 데에 문제없이 성공했고 클라이언트와 연결도 정상적으로 이루어졌다. DBMS와 API를 활용해서 검색한 상품을 데이터베이스에 담고 이를 일정한 조건(정확도, 가격)에 맞추어 출력할 수 있었다. 사용자 평점 및 리뷰 시스템도 정상적으로 구현하였다. 유저의 정보도 암호화하여 보안의 중요성을 유지했다. 이번 프로젝트를 진행하면서 데이터베이스를 활용한 효율적인 데이터 저장과 검색과 더불어서 테이블별로 데이터를 구성하는 것이 활용하는 과정에서 훨씬 효율적이었다. 웹 서버 개발에 서버구성 말고도 UI, 데이터베이스, 외부 API, 보안등 신경 써야 할 부분이 생각보다 많았고 이러한 과정이 웹 개발에 대한 이해도와 기술력을 증진시키는 발판이 되었다. 모든 조원이 HTML 및 PUG를 처음 다뤄보아서 프론트 쪽 개발에 많은 시간과 과정이 필요했는데 완벽한 결과물을 만들기보다 지속적인 고민과 피드백을 통해 부족한 부분을 차근차근 채워나가는 것이 작지만 뿌듯했다. 그리고 사용자의 개인 정보와 같은 민감한 데이터를 암호화함으로써 데이터의 신뢰성을 보장했다. 마지막으로 팀원들과 역할과 책임을 분담해 프로젝트를 진행한 것이 효과적인 팀 프로젝트의 중추적인 역할을 했다고 생각한다.

부록

1. 프로젝트 수행결과물 첨부

https://github.com/znantothelimit/webserver_shoppingmall.git