

# 웹 응용 프로그래밍 텀 프로젝트

소속 : 정보컴퓨터공학부

학번 : 202355514

이름 : 강지원

## 1. 프로젝트 주제 및 선정 계기

### Green Shop

최근 인건비가 오르면서 농산물 판매에 있어서 납품업체와의 거래 가격이 오르고, 날씨의 변화로 인해 상품성이 떨어지는 농작물들이 많아 농민들이 어려움을 겪는다는 기사를 보았다.

이를 보고 농민들이 직접 상품을 도매하는 사이트를 만들어 보고자 하였다. 농민들이 직접 가격을 책정해 사고 팔 수 있고 농작물 뿐만 아니라 종자 등 농업과 관련 물품을 팔 수 있는 사이트를 제작하였다.

## 2. 개발 기능

### 1) 핵심 시나리오

Form을 통해서 상품 정보 입력 받은 후, Products 테이블에 상품을 등록

등록된 상품을 홈 화면에 이름, 사진, 수량, 가격과 함께 표시

등록된 상품을 카테고리별로 검색할 수 있는 기능

가장 최근에 구매 이력이 있는 상품은 "인기상품" 표시

재고가 1개 남은 상품은 "매진 임박" 표시

로그인 후 구매 시, purchases 테이블에 구매 내역 등록

로그인 하지 않아도 상품 구매 가능

### 2) 로그인 기능 구현

Passport 미들웨어의 local Strategy 전략을 사용해 구현하였으며 users 테이블에 사용자 정보를 저장한다. 홈 화면에서 로그인 버튼 클릭 시, 로그인 페이지로 이동하도록 구현되었으며 로그인 페이지에서 회원가입 또한 할 수 있다.

로그인한 유저에 한해서 현재까지의 구매 내역 (상품의 이름, 수량, 가격)을 공개하며, 구매 횟수에 따라 등급을 부여한다.

\*제출 파일의 passport 폴더 및 controllers/auth.js, routes/auth.js, views/login.html, views/join.html에 구현하였다.

### 3) Express 사용

Static과 cookie-parser, json 사용하였다.

// app.js

```
app.use(express.static(path.join(__dirname, 'public')));
app.use('/img', express.static(path.join(__dirname, 'uploads')));
app.use(express.json());
app.use(cookieParser(process.env.COOKIE_SECRET));
```

### 4) Rest API 사용

server와 client 간의 통신에서 get, post 메서드에 요청, 응답하도록 구현하였다.. 많은 구현 부분에 사용하였는데 controllers/page.js 부분을 예로 설명하자면

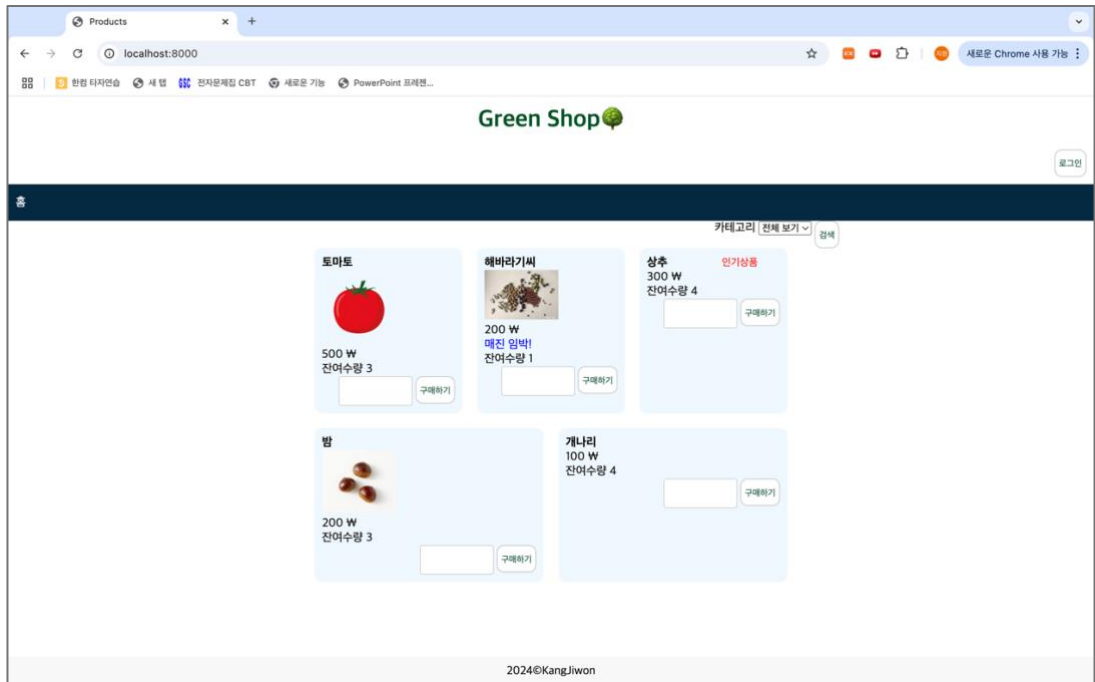
```
1 exports.renderMain = async (req, res, next) => {
2   try {
3     const queryTag = req.query.category;
4     const userId = req.user ? req.user.id : null;
5     let productsQuery = `
6       SELECT p.id, p.name, p.quantity, p.price, p.img
7       FROM products p`;
8
9     if (queryTag) {
10      productsQuery += ` WHERE p.code LIKE ? ORDER BY p.createdAt DESC`;
11    } else {
12      productsQuery += ` ORDER BY p.createdAt DESC`;
13    }
14
15    const [products] = await db.execute(productsQuery, [queryTag ? `%${queryTag}%` : '']);
16
17    let purchaseCount = 0;
18    if (userId) {
19      const [purchases] = await db.execute(`
20        SELECT COUNT(*) AS purchaseCount
21        FROM purchases
22        WHERE userId = ?
23        `, [userId]);
24      purchaseCount = purchases[0].purchaseCount;
25    }
26
27    const [latest] = await db.execute(`
28      SELECT * FROM purchases
29      ORDER BY purchaseDate DESC
30      LIMIT 1
31    `);
32    const [categories] = await db.execute(`SELECT DISTINCT code FROM products;`);
33    const latestP = latest.length > 0 ? latest[0] : null;
34    res.render('main', {
35      title: 'Products',
36      user: req.user,
37      products,
38      purchaseCount,
39      latestP,
40      categories
41    });
42
43    console.log('Products:', products);
44
45  } catch (err) {
46    console.error(err);
47    next(err);
48  }
49 };
```

Server에서 DataBase를 통해서 데이터를 가져오고 Client에게 해당 데이터를 json형식(빨간 박스 부분)으로 제공해 화면에 보여준다. 해당 모듈은 routes/page.js에서 **router.get('/', renderMain);** 으로 라우터를 구현하였다.

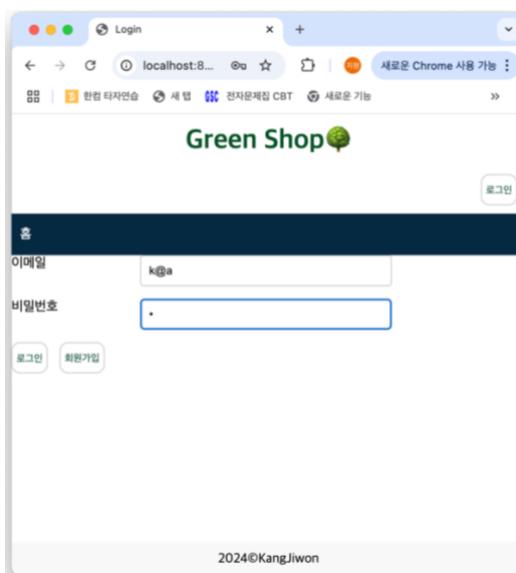
Json형식 데이터 전달은 controllers/post.js에서 구현하였다.

### 3. 사용 방법

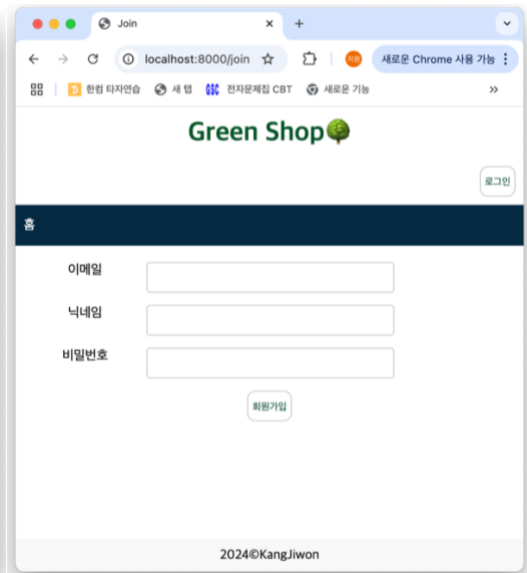
홈 화면 모습 ( 로그인 전 )



우측 상단 로그인 버튼 클릭 시 화면

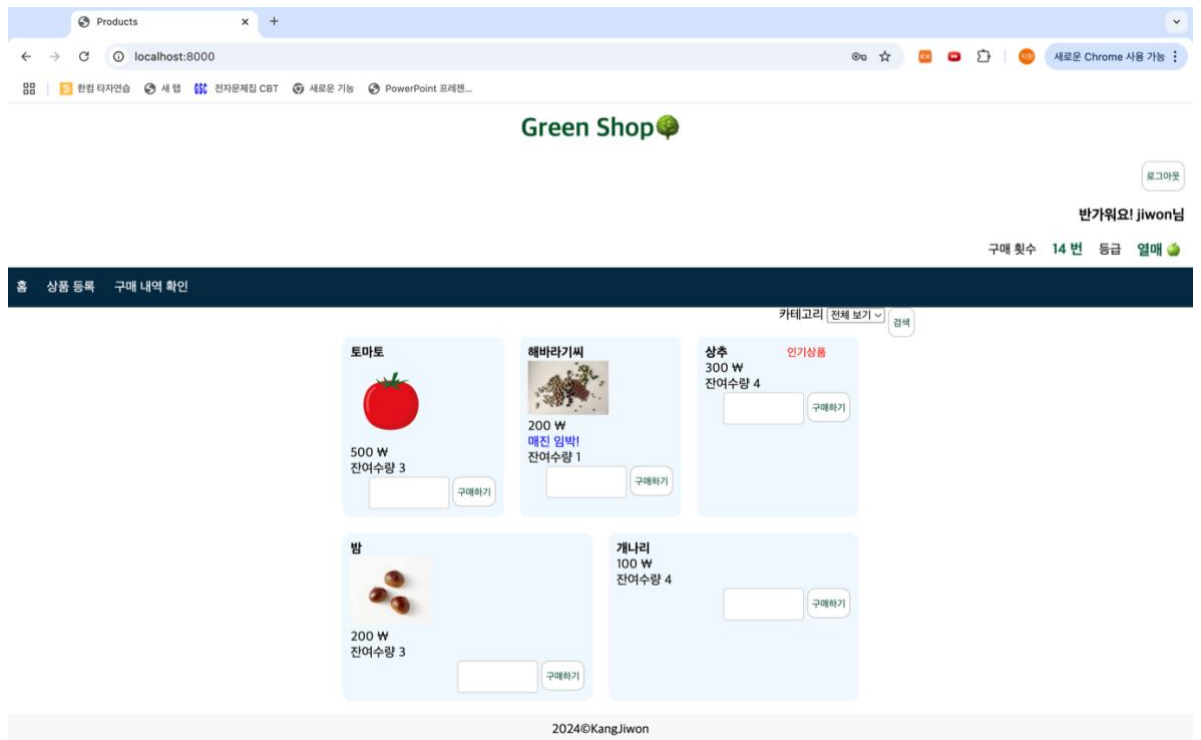


회원 가입 버튼 클릭 시 화면



## 로그인 후 홈 화면 모습

\*좌측 상단 로그인 유저 이름과 구매 횟수, 등급 표시 및 상품등록, 구매내역확인 탭이 상단바에 생성됨



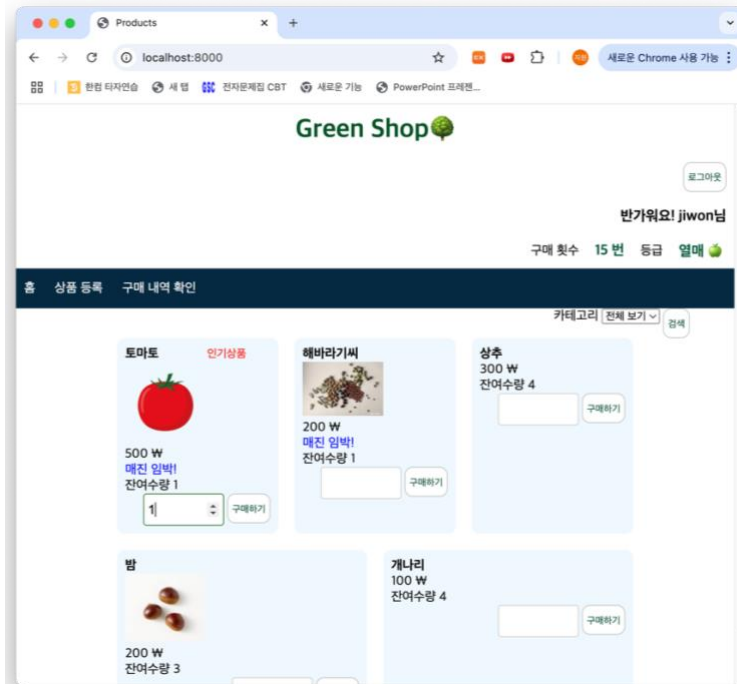
## 구매 수량 입력 후 구매하기 버튼 클릭



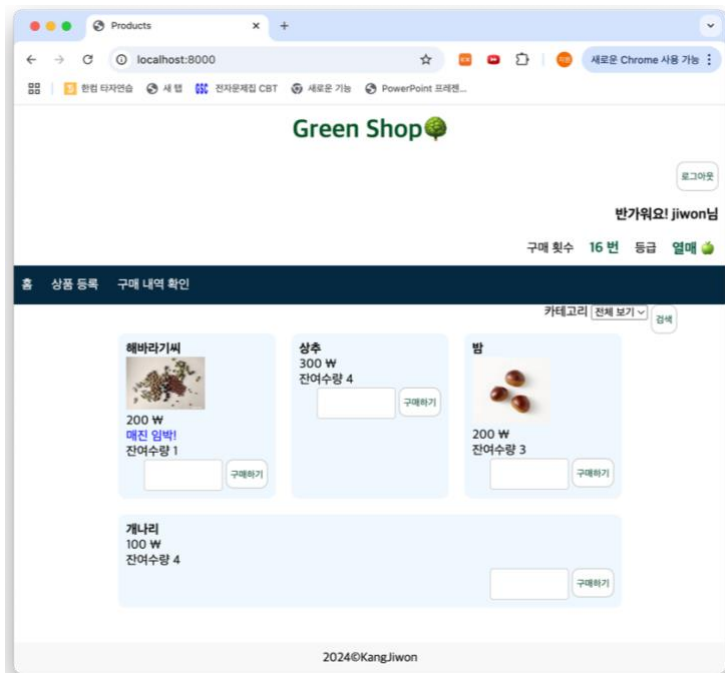
## 구매하기 클릭 후의 모습



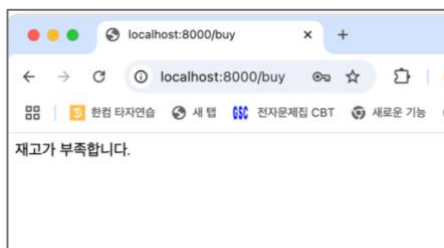
마지막 남은 수량 구매 시, 상품이 화면에서 제외된다.



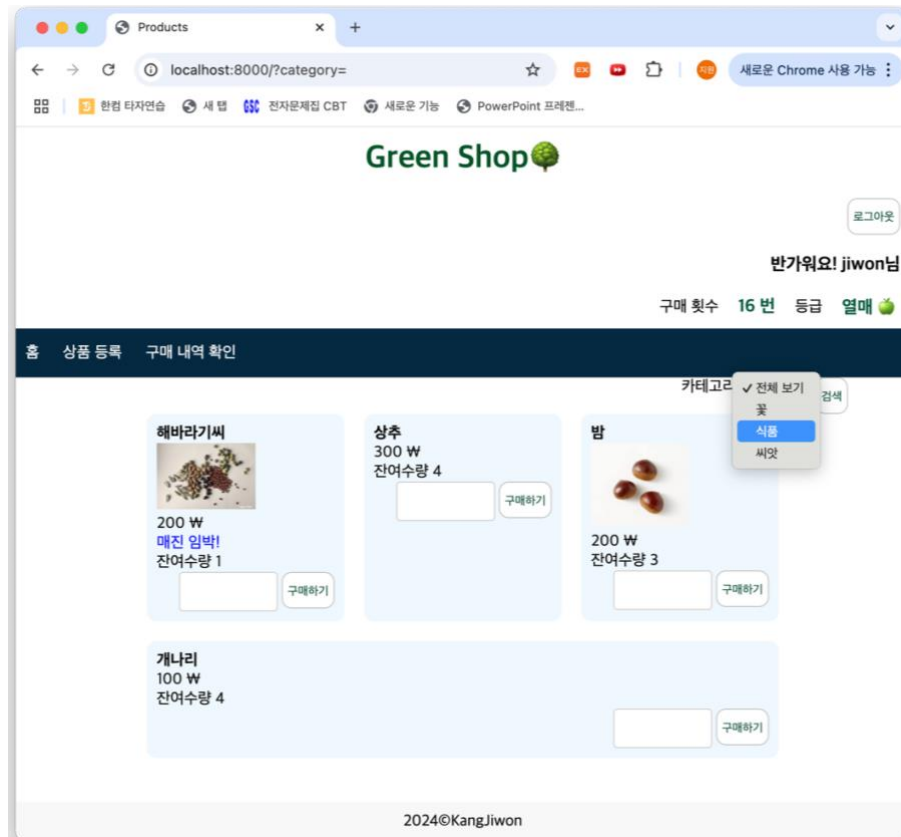
구매 후의 모습



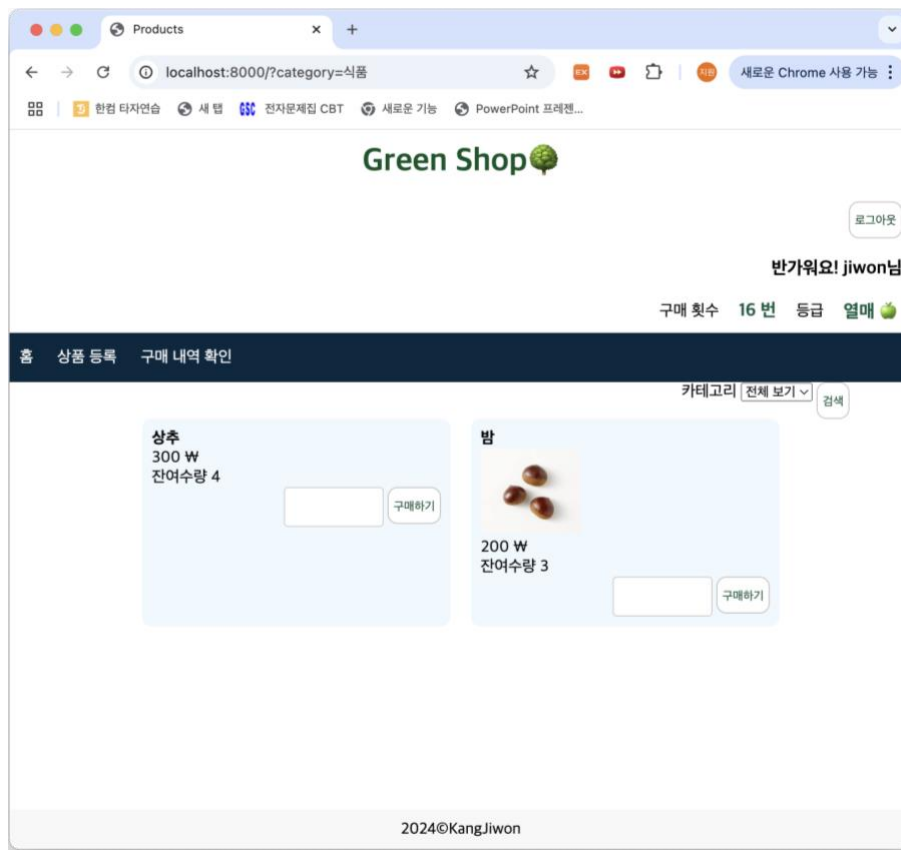
재고보다 많은 수량을 구매하고자 하는 경우



카테고리 선택 시, 카테고리에 맞는 상품만 표시



카테고리 선택 후, 검색 버튼 누른 모습



## 상품 등록 화면

Upload

게나리 사진 - Google 검색

localhost:8000/post

반가워요! jiwon님

구매 횟수 16 번 등급 열매

홈 상품 등록 구매 내역 확인

### 상품 등록

상품명 토마토

상품코드 식품

수량 5

가격 500

사진 업로드

상품 등록

## 등록 후 홈화면으로 이동된 모습

Products

게나리 사진 - Google 검색

localhost:8000

반가워요! jiwon님

구매 횟수 16 번 등급 열매

로그아웃

홈 상품 등록 구매 내역 확인

카테고리 전체 보기 검색

토마토 인기상품

500 W  
잔여수량 5

구매하기

해바라기씨

200 W  
매진 임박!  
잔여수량 1

구매하기

상추

300 W  
잔여수량 4

구매하기

밤

200 W  
잔여수량 3

구매하기

개나리

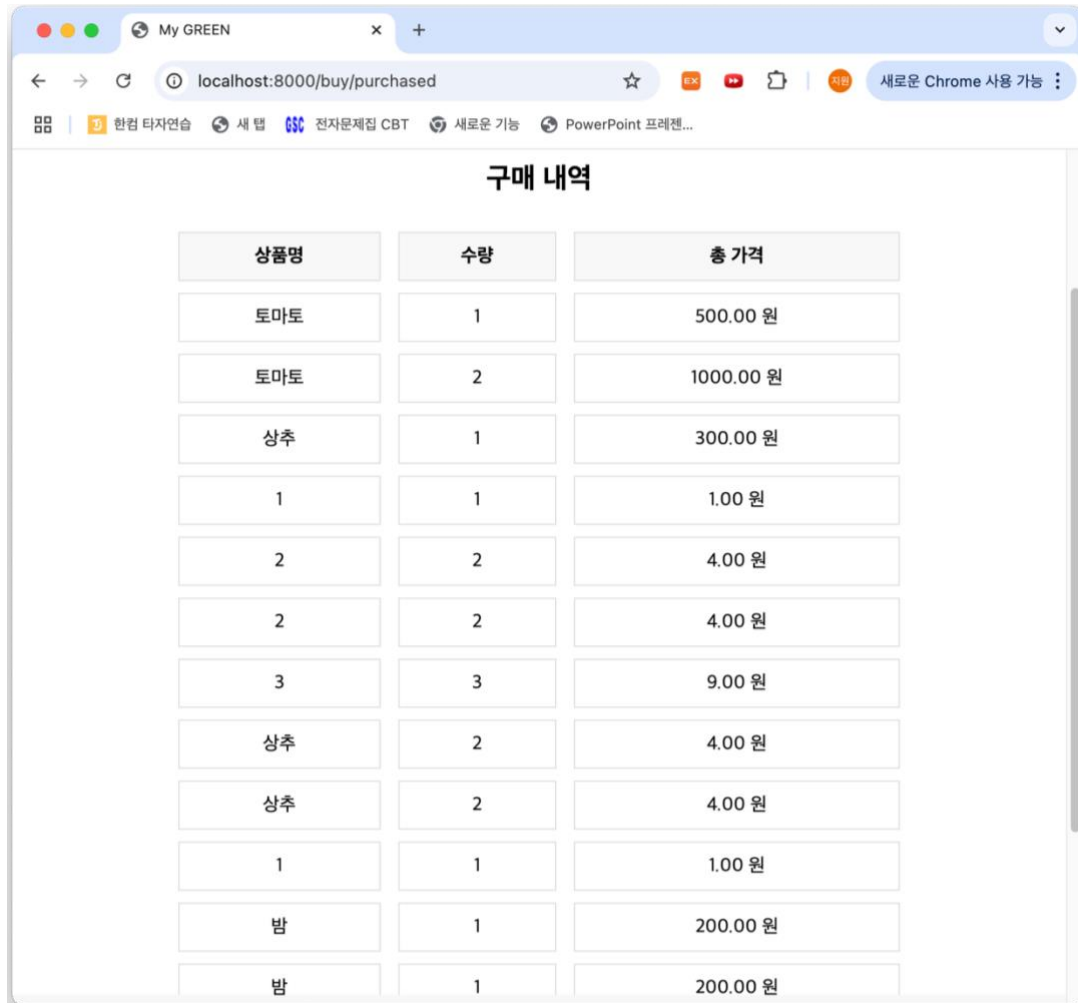
100 W  
잔여수량 4

구매하기

2024©KangJiwon



## 구매 내역 확인 탭 화면



상품명	수량	총 가격
토마토	1	500.00 원
토마토	2	1000.00 원
상추	1	300.00 원
1	1	1.00 원
2	2	4.00 원
2	2	4.00 원
3	3	9.00 원
상추	2	4.00 원
상추	2	4.00 원
1	1	1.00 원
밤	1	200.00 원
밤	1	200.00 원