



UGV

(유지비, Ugly Vegetable)

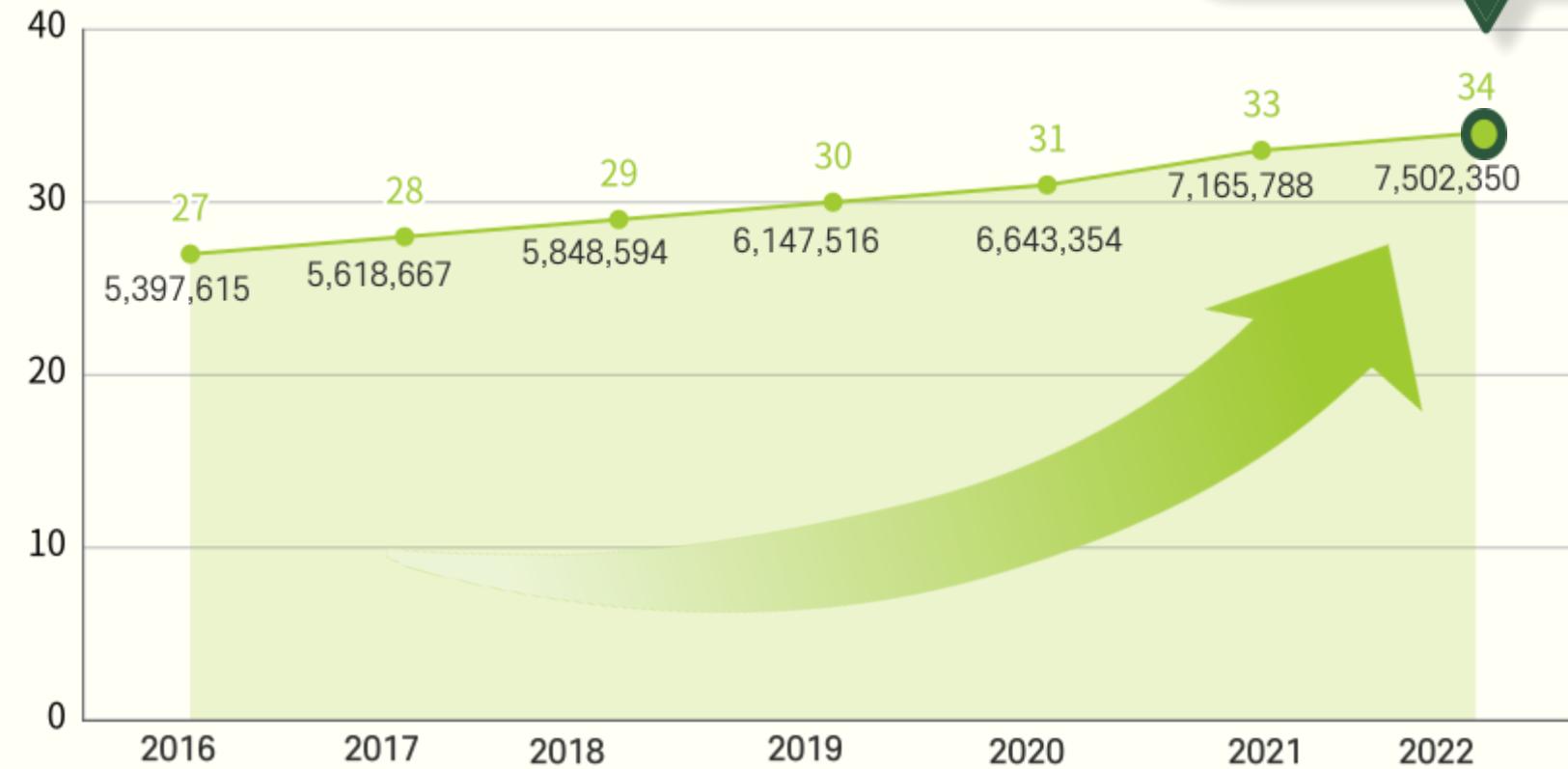
1인 가구를 위한 못난이 과일, 채소 판매 반응형 웹사이트

기획의도



- 1인 가구수 비율 (통계청 KOSIS 지표)

2022년
1인 가구 비율 34.5%
1인 가구 수 750만 2,350
가구



최근 1인 가구의 비율이 꾸준히 증가하고 있습니다. 이에 대형마트에서 소량의 채소나 반조리 식품 등을 판매하고 있습니다. 하지만, 농산물의 기본적인 가격이 비싼데다가 1인분으로 소량 판매하기 위해 세척했다는 이유로 일반 채소들보다 비싸기도 합니다.

이런 이유로 혼자 사는 사람들도 편안하게 야채와 과일을 구매할 수 있는 사이트를 기획하게 되었습니다.

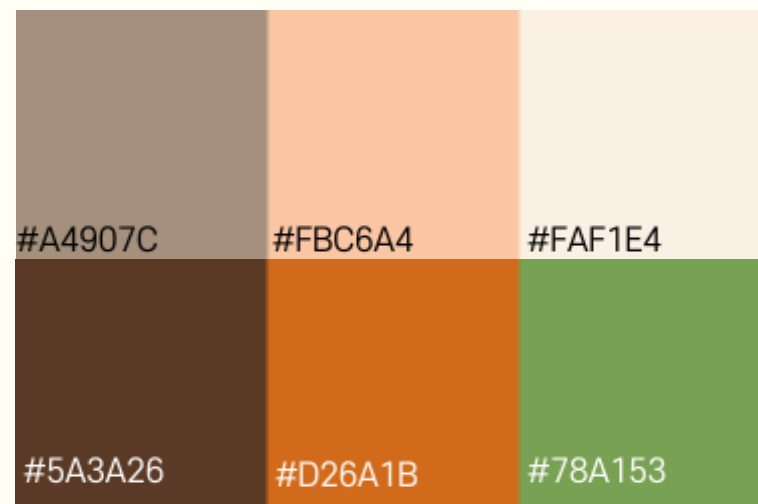
- 규격 외 농산물의 문제

의미	맛과 영양 등 품질에는 아무런 이상이 없지만, 모양과 크기가 시장 기준과 다른 못난이 농산물.
현실	이상기후로 인해 규격 외 농산물의 생산량은 점점 증가하고 있지만, 상품성이 떨어진다는 이유로 판매되지 못하고 폐기되는 것이 대부분. 이렇게 폐기되는 농산물은 국내 전체 생산량의 15%이며, 전 세계적으로는 13t에 달함.
문제	아무 이상이 없는 농산물이 식탁에 오르지 못하고, 생김새 때문에 폐기된다는 점에서 큰 경제적 손실을 야기하며, 노동력이 낭비됨. 또한 버려진 농산물들이 부패하는 과정에서 지구 온난화를 촉진하는 온실 가스 발생. 결국, 버려지는 농산물이 이상기후를 가속화하고 이로 인해 못난이 농산물 생산량이 증가하는 악순환이 반복. 이렇게 폐기되는 농산물이 많아지면, 결국 일반 농산물들의 가격이 올라 소비자들의 부담이 더욱 커지게 됨.

주제 선정 (Concept)

주제

1인 가구가 지속적으로 증가하는 요즘,
비싼 야채와 과일을 소량으로 합리적인
가격에 구매할 수 있는 반응형 웹 사이트



색상

메인 컬러와 포인트 컬러로 흙과 신선한
농산물이 떠오르는 색상을 선정.
보조 컬러는 메인 컬러의 높은 채도를
완화시켜주는 연한 채도의 색상을 선정.

지속가능한
생산 및 소비
추구

규격 외 농산물로
분류된 신선한
농산물을 저렴한
가격에 판매

소량으로
손질된 채소
판매

'1인 가구'
타겟

새벽 배송
정기 배송

가치소비

로고

로고는 당근을 연상하게 하는 로고와
메인 컬러와 포인트 컬러를 사용한
텍스트 로고 두 가지 사용.



의미

UGV에는 "Ugly Vegetable"이라는
의미와 1인 가구의 (생활) 유지비를
지켜준다는 중의적인 의미가 담겨
있음.



목차 (Index)

01

역할 및 소감

02

개발환경 & 적용 기술

03

요구사항분석

04

DB 모델링

05

형상 관리

06

협업 관리

07

일정 관리

08

주요기능소개

09

코드 리뷰

01. 역할 및 소감



채승병 부팀장

(FrontEnd)

- 메인/챗봇 API

(BackEnd)

- 로그인/로그아웃

+ 협업 관리(피그마)

리액트 기반 웹사이트 프로젝트를 진행하면서 많은 것을 배우고 성장할 수 있었습니다. 이 프로젝트를 통해 리액트의 강점과 유연성에 대한 깊은 이해를 얻었으며, 컴포넌트 기반 아키텍처를 활용하여 코드를 구조화하는 방법을 익혔습니다.

프로젝트에서 발생한 다양한 문제들을 해결하며 문제 해결 능력과 팀원들간의 협력이 중요하다는 것을 느꼈습니다. 또한 사용자 경험을 고려한 UX/UI 디자인에 대한 고민과 구현을 통해 웹사이트의 완성도를 높이는 중요성을 깨닫게 되었습니다.

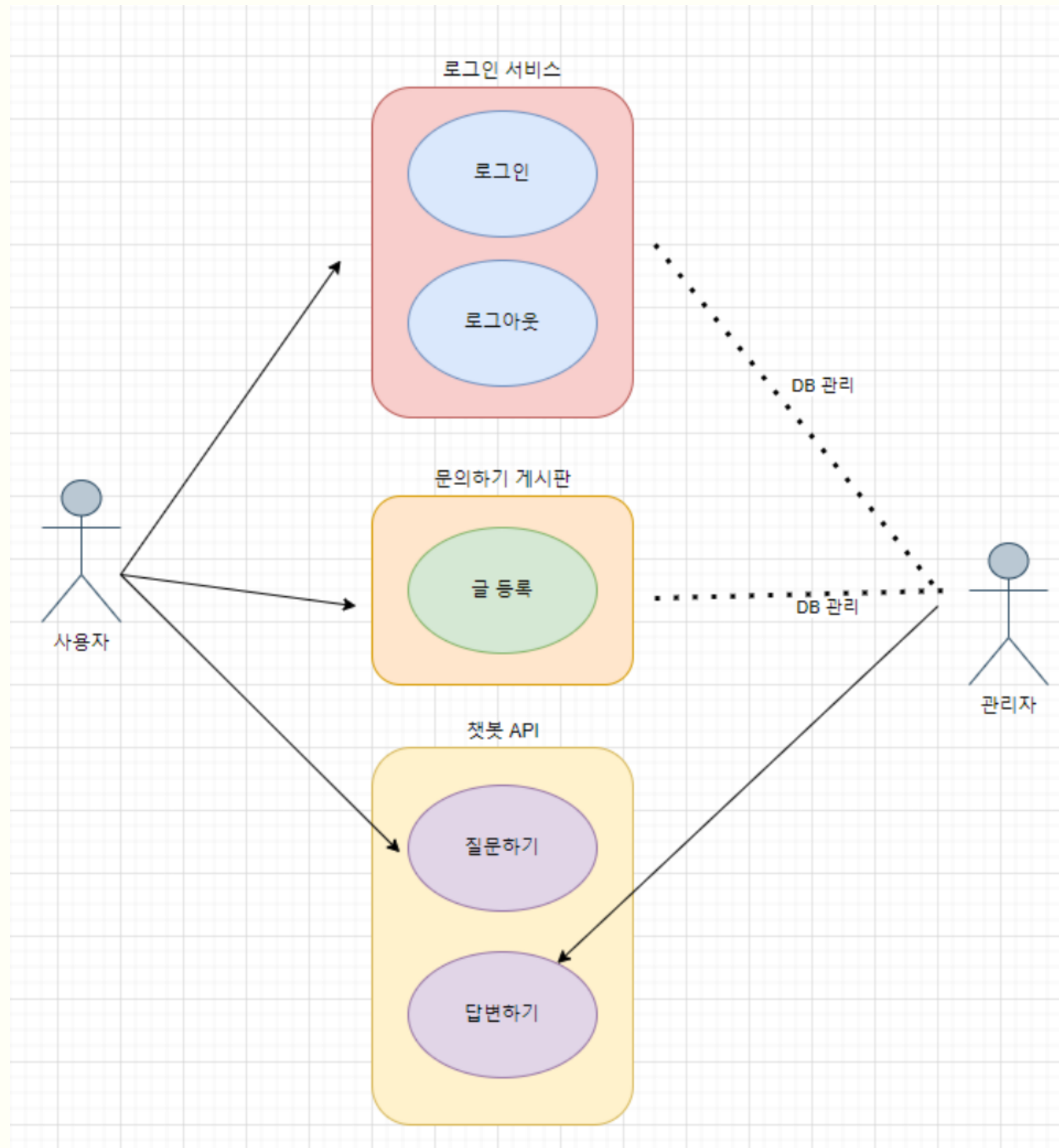
앞으로의 프로젝트에서도 이러한 경험을 바탕으로 더 나은 웹 개발을 위해 노력하고 성장해 나가고 싶습니다.

02. 개발 환경 & 적용 기술





03. 요구사항분석 (Use Case Diagram)


Draw.io를 활용하여 작성한
Use Case Diagram




04. DB 모델링

Member	
	MEMBER_IDX number(10,0) (NN)
	MEMBER_ID varchar2(255) (NN)
	MEMBER_PW varchar2(255) (NN)
	MEMBER_NAME varchar2(255) (NN)
	MEMBER_EMAIL varchar2(255) (NN)

consultation	
	bno number(10,0) (NN)
	title varchar2(255) (NN)
	author varchar2(255) (NN)
	email varchar2(255)
	content varchar2(255) (NN)

회원	
	회원번호 number(10,0) (NN)
	아이디 varchar2(255) (NN)
	비밀번호 varchar2(255) (NN)
	이름 varchar2(255) (NN)
	이메일 varchar2(255) (NN)

문의하기 게시판	
	문의 번호 number(10,0) (NN)
	문의 제목 varchar2(255) (NN)
	작성자 varchar2(255) (NN)
	이메일 varchar2(255)
	문의 내용 varchar2(255) (NN)

05. 형상관리

(Git/Github/Sourcetree)

- ① 커밋 메시지 구조
<type> : 커밋 제목
<커밋 설명>
<이슈 트래커(옵션)>
- ② type
feat : 새로운 기능 추가
refactor : 리팩토링(결과는 같지만 코드를 개선함)
fix : 버그 수정
style : 스타일 수정
comment : 주석 수정 및 삭제
Docs : 문서와 관련된 모든 것
Implement : 코드가 추가된 정도보다 더
주목할만한 구현체를 완성시켰을 때
Remove : 파일을 삭제하는 작업만 수행한 경우

- ③ 이슈트래커
: 해당 커밋이 어떤 이슈를 해결하였는지 표기.
(혹은 해결하지 못한 이슈도 작성 가능)

Active branches

ugv/react/jeeeun	Updated 1 hour ago by jeeeunkim1012	0 7	New pull request			
ugv/springBoot/sujin	Updated 7 hours ago by tnwlsdlWkdWkdaos1321323	0 4	New pull request			
ugv/springBoot/jeeeun	Updated yesterday by jeeeunkim1012	0 2	New pull request			
ugv/react/sujin	Updated yesterday by tnwlsdlWkdWkdaos1321323	0 4	New pull request			
ugv/springBoot/chaeb	Updated 3 days ago by chaesb98	0 10	New pull request			
ugv/react/chaeb	Updated 3 days ago by chaesb98	0 9	New pull request			
ugv/react/jui	Updated 4 days ago by jui990706	0 5	New pull request			
ugv/react/younghak	Updated 4 days ago by kimyounghak	0 1	New pull request			
ugv/springBoot/team	Updated 5 days ago by tn					
ugv/react/team	Updated 5 days ago by tnwlsdlWkdWkdaos1321323					

ugv/springBoot... 11 branches 0 tags

Go to file Add file > Code >

This branch is 10 commits ahead of master. Contribute >

chaesb98 <type>: react와 springBoot (DB)연동 후 로그인 기능 구현 ... 941ab84 3 days ago 11 commits

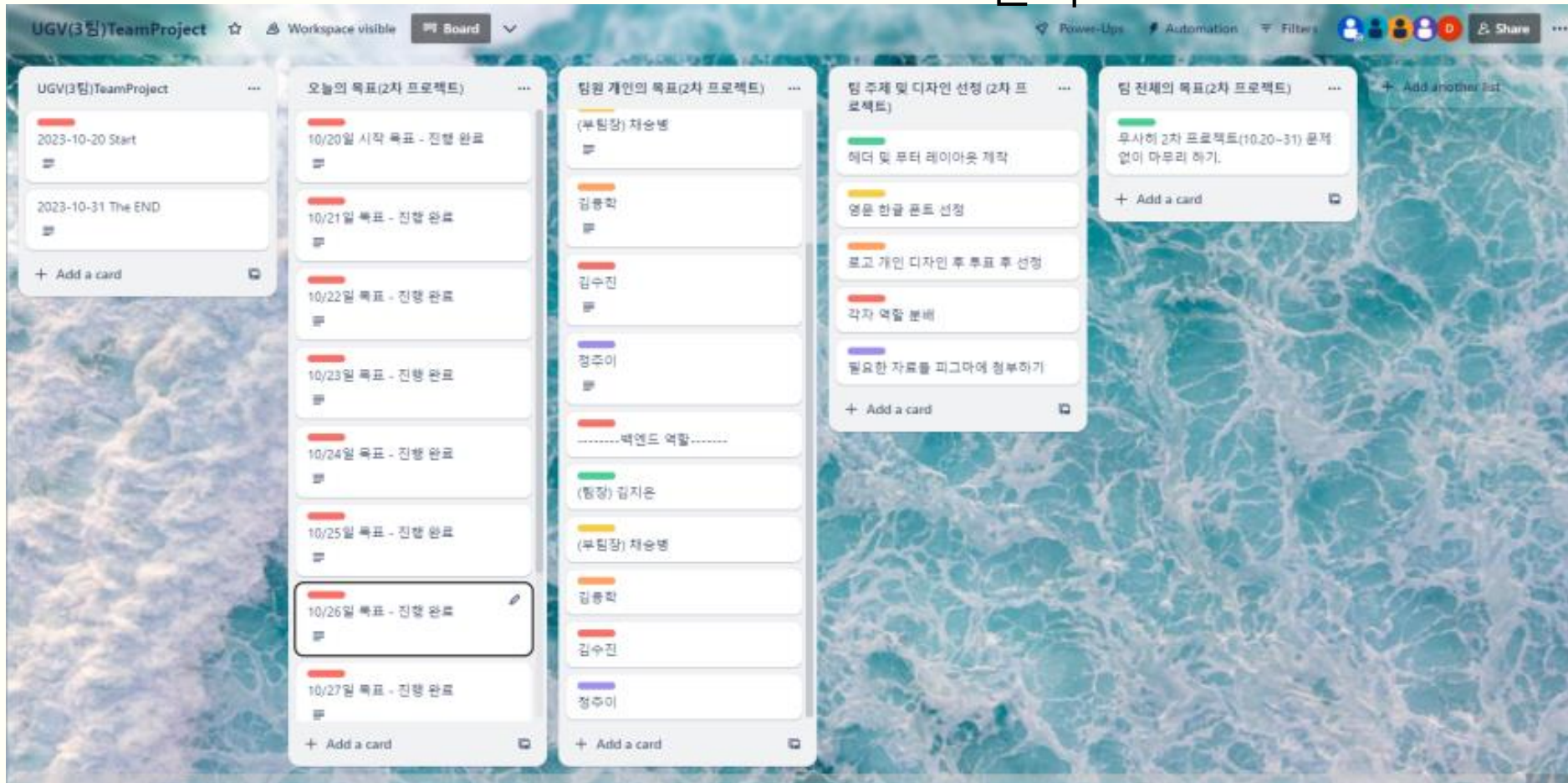
231025_react_chaeb.zip	<type>: feat 메인페이지	last week
231026_UGV.zip	<type>: feat 메인페이지 완성 후 페이지 합치기	5 days ago
231028_UGV.zip	<type>: 페이지 병합 완료 후 로그인 버튼 수정	3 days ago
Test.txt	테스트입니다.	last week
Test_branch.txt	브랜치 테스트입니다.	last week
UGV_SpringBoot.zip	<type>: react와 springBoot (DB)연동 후 로그인 기능 구현	3 days ago

Help people interested in this repository understand your project by adding a README. Add a README

06. 협업 관리

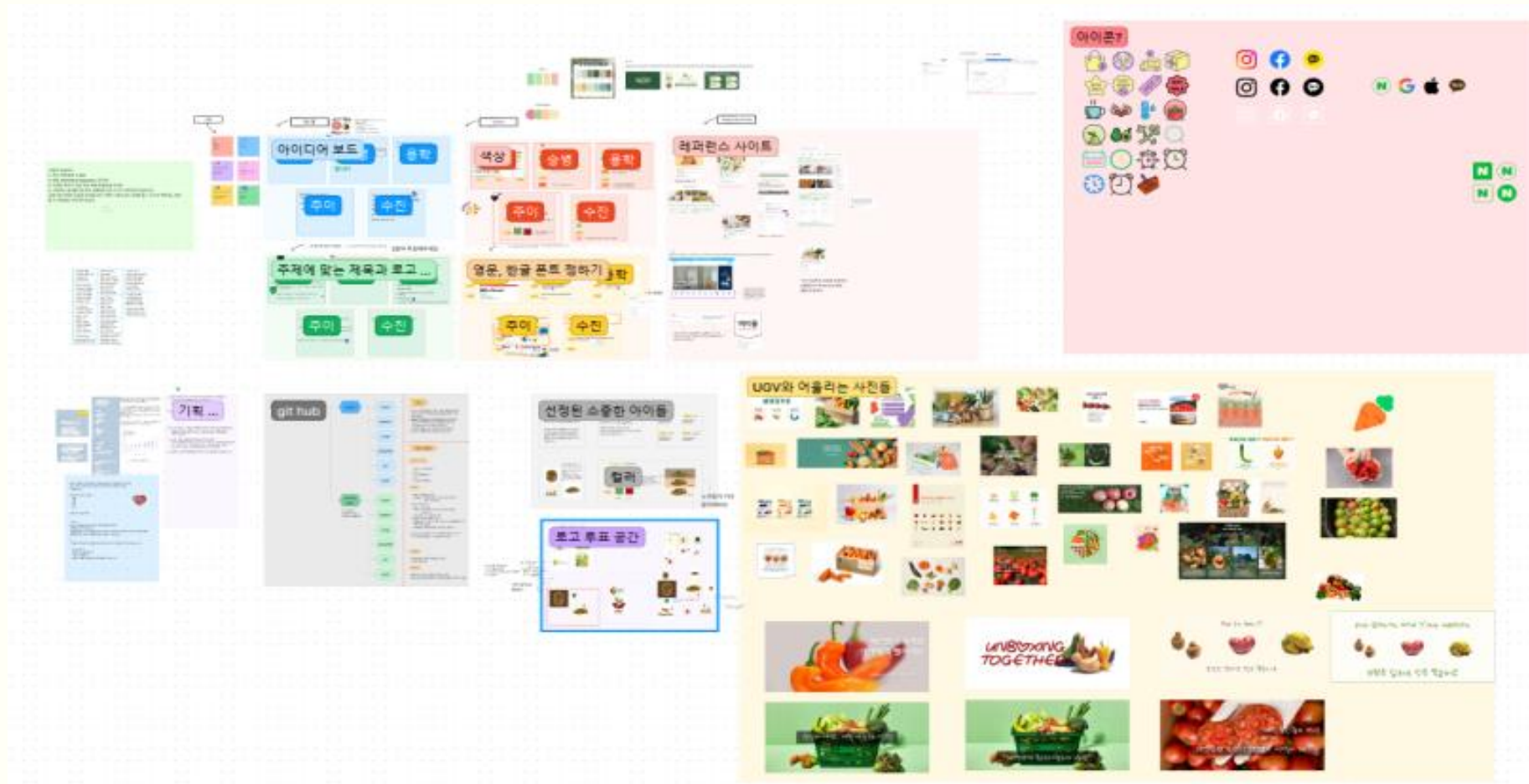
1. Agile Trello Kanbanboard

전체적인 일정 공유 및 관리



2. Figma

UI 디자인 선정 및 자료 공유



회의

회의 캘린더 유형별 리스트 내 회의 전체

참석자 유형 생성자 이벤트 시간 필터 추가

10/30 결과물 점검 및 코드 리뷰

10/27 중간 점검

10/26 DB 내용 회의

10/24 React 코딩 시작 전, 기획 회의 및 역할 분담

10/23 UI 정하기

10/20 주제 및 컨셉 회의

팀 주간 회의@2023년 10월 19일

짧은 회의

팀 주간 회의

짧은 회의

짧은 회의

짧은 회의

브레인스토밍

팀 주간 회의

3. Notion

회의 일정 관리

07. 일정 관리

(Gantt Chart)

프로젝트 일정관리를 위한
바(bar) 형태의 도구인 간트
차트로 일정을 관리

프로젝트 일정 (10월)

20일 21일 22일 23일 24일 25일 26일 27일 28일 29일 30일 31일

주제 선정 및 컨셉트 기획

FrontEnd 개발

BackEnd 개발

최종 점검

김지은	팀장
채승병	부팀장
김용학	팀원
김수진	팀원
정주이	팀원

2차 프로젝트 기간	2023-10-20 ~ 2023-10-31(12일간)	
Front		
주제 선정	2023-10-20	과제

메인/첫화면	회사소개,서류관리	헤더,푸터,상당문의	자주묻는질문,찾아오시는길,상품소개	로그인/로그아웃/배송안내
승병	용학	수진	주이	지은

3Team Project (UGV) 간트차트				완료여부 표시			20일	21일	22일	23일	24일	25일	26일	27일	28일	29일	30일	31일
				진행완료	진행일수	진행여부	2023/10/20 ~ 2023/10/31											
공통	UI/UX 컨셉트 기획	프트폴리오 : 주제/제목/부제목/기획의도 선정	10/20	10/23	4일	진행완료												
	테스트	Unit Test(유닛 테스트)	10/20															
	품질 관리	코드 리뷰	10/20	10/30	12일	진행완료												
김지은		Front-End react/bootstrap 로그인 만들기	10/25	10/26	2일	진행완료												
		Front-End react/bootstrap 로그아웃 만들기	10/25	10/26	2일	진행완료												
		Front-End react/bootstrap 회원가입 만들기	10/29	10/30	2일	진행완료												
		Front-End react/bootstrap 배송안내 만들기	10/30	10/30	1일	진행완료												
채승병		Front-End react/bootstrap 메인페이지 만들기	10/25	10/26	2일	진행완료												
		Front-End Javascript 반응형 웹 : 챗봇 API 연동 고객 실시간 문의 웹 구현	10/26	10/26	1일	진행완료												
		React 연동 Back-End OracleDB 설계 및 구축 : 로그인/로그아웃	10/27	10/27	1일	진행완료												
김용학		Front-End react/bootstrap 회사소개 만들기	10/25	10/26	2일	진행완료												
		각종 서류 관리(트렐로,칸토차트 등 필요한 서류 관리)	10/20	10/31	12일	진행완료												
정주이	사용자 웹 서비스 (Front-End HTML5, CSS3, Javascript, jQuery, Bootstrap, 반응형 웹, Back-End Java Servlet/JSP, SpringMVC, Maven, Mybatis, OracleDB	Front-End react/bootstrap 찾아오시는 길 kakao api 적용	10/24	10/24	1일	진행완료												
		Front-End react/bootstrap 찾아오시는 길 만들기	10/24	10/26	3일	진행완료												
		Front-End react/bootstrap 자주묻는질문 만들기	10/26	10/26	1일	진행완료												
		Front-End react/bootstrap 상품소개 만들기	10/26	10/27	2일	진행완료												
김수진	사용자 웹 서비스 (Front-End HTML5, CSS3, Javascript, jQuery, Bootstrap, 반응형 웹, Back-End Java Servlet/JSP, SpringMVC, Maven, Mybatis, OracleDB	Front-End react/bootstrap 헤더, 푸터 만들기	10/24	10/25	2일	진행완료												
		Front-End react/bootstrap 상당문의 만들기	10/25	10/25	1일	진행완료												
		Back-End SpringMVC Mybatis 자유 게시판 글목록조회	10/26	10/30	5일	진행완료												
		Back-End OracleDB 설계 및 구축 : 자유 게시판 글목록조회	10/26	10/30	5일	진행완료												

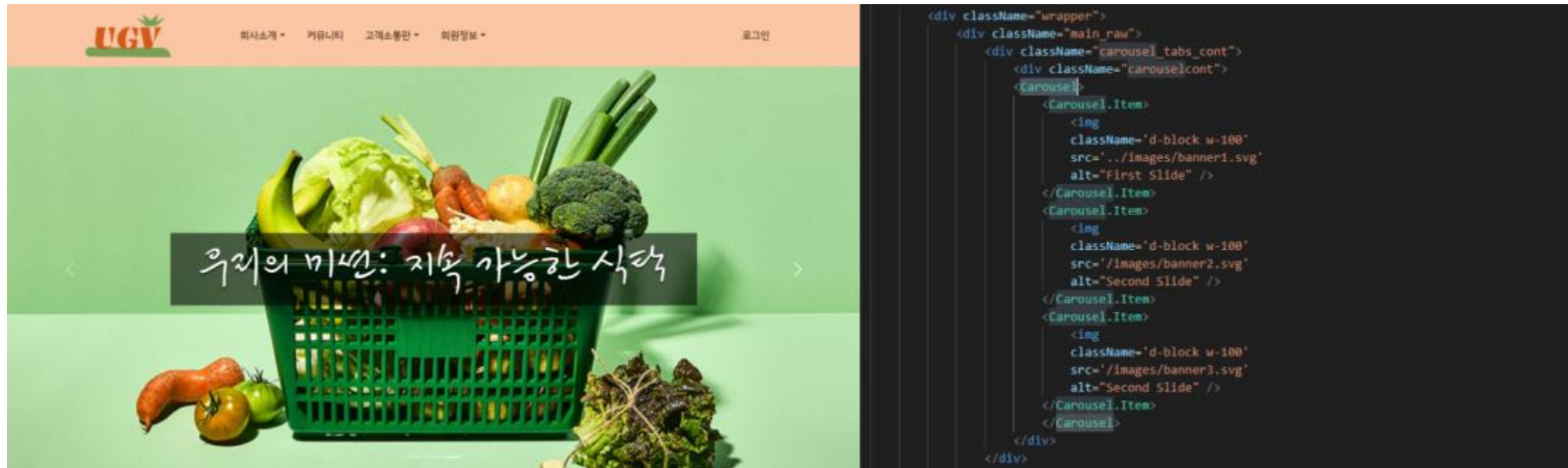
08. 주요 기능 소개 (기능명세서)

3팀 UGV(유지비) 기능명세서													
프로젝트 개요		목적 및 목표	React를 적극 활용하여 Front 웹 페이지 코딩을 제작하고, Backend 작업을 통해, 로그인과 자유게시판 글 목록 조회 및 등록을 구동하게끔 함.										
기능 목록	로그인	입력 사항	아이디(ID),비밀번호(PW)										
		로그인	로그인 성공 ("UserName + 회원님. 환영합니다!") 아이디/비밀번호 오류 ("회원 정보가 없습니다!")										
	문의게시판	입력 사항	문의사항(Enquiry), 이메일(Email), 이름(Name), 제목(Title), 개인정보수집동의(체크,Check)										
		글 목록 등록	등록 완료 ("문의사항이 등록되었습니다")										
	챗봇API	입력 사항	질문 사항 입력 (이후 전화번호와 이메일 입력)										
		글 목록 등록	연락처를 남겨주세요. 오프라인 상태가 되시면 문자(알림톡) 및 이메일로 답변 알림을 보내드려요. (수집된 개인정보는 상담 답변 알림 목적으로만 이용되고, 직접 삭제하기 전까지 보유합니다. 제출하지 않으시면 상담 답변 알림을 받으실 수 없어요.)										

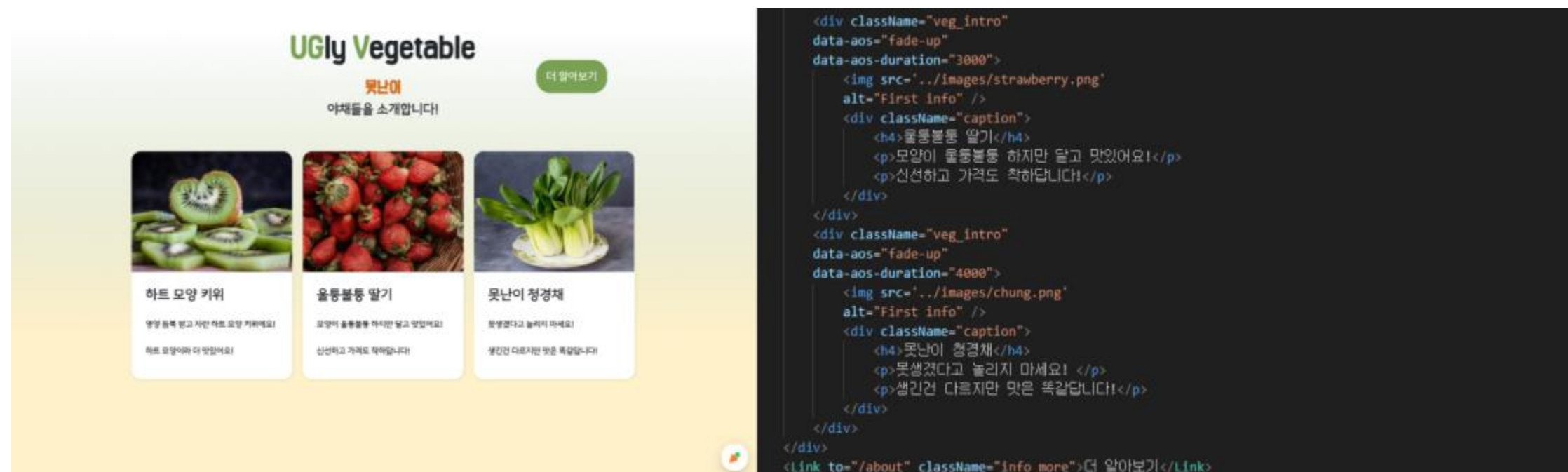
09. 코드 리뷰 (Code Review)

메인 (Main)

1) Carosel을 이용한 풀스크린 배너. 이미지들이 자동으로 슬라이드 됨.



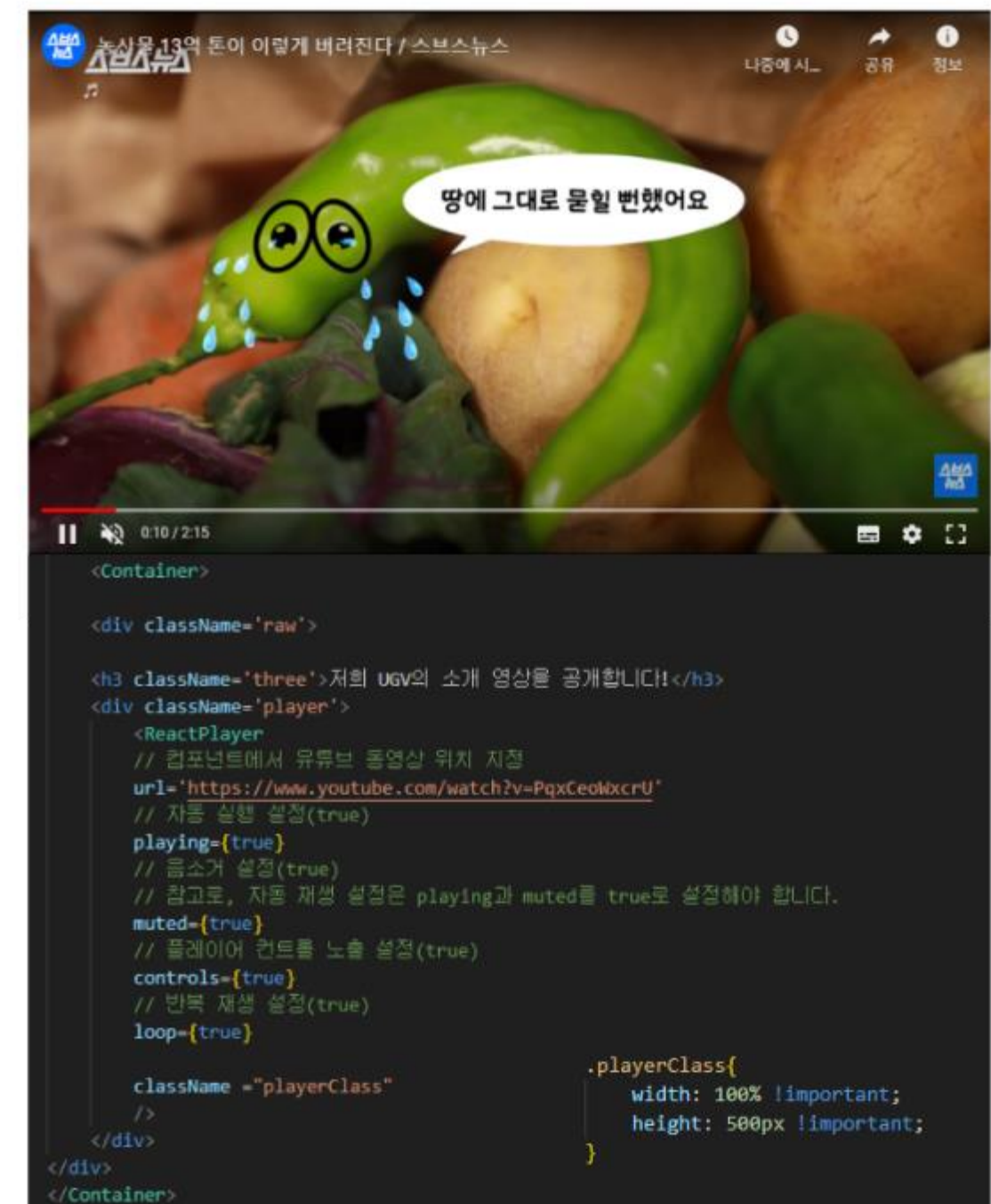
2) AOS 라이브러리를 이용한 효과.
스크롤 이동이 아래에서 위로 올라오며, 각각 효과 시간을 다르게 적용.



회사 소개

1) React Plyer를 이용하여 유튜브 영상 삽입

저희 UGV의 소개 영상을 공개합니다!



09. 코드 리뷰 (Code Review)

Header

1) React Bootstrap의 nav 태그 중 NavDropdown 태그를 이용하여 토글 형식으로 서브메뉴가 Dropdown 됨.

```
<Nav className="me-auto">
  { /* <Nav.Link href="/About">회사소개(About)</Nav.Link> */ }
  <NavDropdown title="회사소개" id="basic-nav-dropdown">
    <NavDropdown.Item href="/About">회사소개</NavDropdown.Item>
    <NavDropdown.Divider />
    <NavDropdown.Item href="/Location">찾아오시는 길</NavDropdown.Item>
  </NavDropdown>
  <Nav.Link href="/Community">커뮤니티</Nav.Link>
  { /* <Nav.Link href="/QnA">자주묻는질문</Nav.Link> */ }
  { /* <Nav.Link href="/Location">회사위치(Location)</Nav.Link> */ }
  <NavDropdown title="고객소통란" id="basic-nav-dropdown">
    <NavDropdown.Item href="/notice">공지사항</NavDropdown.Item>
    <NavDropdown.Divider />
    <NavDropdown.Item href="/freeboard">자유게시판</NavDropdown.Item>
    <NavDropdown.Divider />
    <NavDropdown.Item href="/ReviewList">후기 게시판</NavDropdown.Item>
    <NavDropdown.Divider />
    <NavDropdown.Item href="/QnA">자주묻는 질문</NavDropdown.Item>
  </NavDropdown>
  <NavDropdown title="회원정보" id="basic-nav-dropdown">
    <NavDropdown.Item href="/notice">로그인</NavDropdown.Item>
    <NavDropdown.Divider />
    <NavDropdown.Item href="/freeboard">회원가입</NavDropdown.Item>
    <NavDropdown.Divider />
    <NavDropdown.Item href="/communityboard">마이페이지 </NavDropdown.Item>
    <NavDropdown.Divider />
    <NavDropdown.Item href="/Consultation">상담문의 </NavDropdown.Item>
  </NavDropdown>
</Nav>
```



```
<Row>
  <Col>
    <Navbar.Brand href="/" className='logo'><img src='/images/ugv_logo.png' alt='로고 이미지' /></Navbar.Brand>
  </Col>
  <Col md="auto">
    <div className="conBtn button"> <Link to='/Consultation'>상담문의</Link></div>
  </Col>
  <Col xs lg="2">
    <div className="conBtn button"> <Link to='/QnA'>자주묻는 질문</Link></div>
  </Col>
</Row>
```

Footer

1) React Bootstrap의 링크 태그를 이용하여 각 버튼에 맞는 링크로 이동하도록 함.



09. 코드 리뷰 (Code Review)

찾아오는 길

1) 카카오 API를 활용한 반응형 맵 삽입.

상품 소개

1) Reac Bootstrap의 카드, 버튼, 이미지 Form 적용

정기 배송 서비스

신선한 제철 농산물을 깨끗하게 손질하여 안전하게 배송드립니다!

집에서 간편하게 신선한 제철 농산물을 받아보세요.

미연에는 어떤 맛있는 농산물이 도착할지 두근대는 마음에 UGV에서 배송하는 날만 기다리게 될거예요.
배송지점 농산물은 수확기와 상황에 따라 변경되며 사전 고지되지 않습니다. 또한 농산물의 종류로 인한 교환 및 환불은 불가합니다.
알레르기 등으로 섭취하지 못하는 농산물이 있다면 구매신청 하실 때 미리 요청사항에 작성해주세요.
신선하고 건강한 농산물이 선물처럼 찾아가겠습니다~!



[구독하러 가기!](#)

```
<Card className="text-center mb150">
  <Card.Header>신선한 제철 농산물을 깨끗하게 손질하여 안전하게 배송해드립니다!</Card.Header>
  <Card.Body>
    <Card.Text>집에서 간편하게 신선한 제철 농산물을 받아보세요. <br /> 까다롭게 엄선된 농산물을 깨끗하게 손질하여 바로 드실 수 있도록 집 앞까지 배송해드립니다! <br /> 이번에는 어떤 맛있는 농산물이 도착할지 두근대는 마음에 ugv에서 배송하는 날만 기다리게 될거예요. <br /> 배송되는 농산물은 시기와 상황에 따라 변경되며, 사전 고지되지 않습니다. 또한 농산물의 종류로 인한 교환 및 환불은 불가합니다. <br /> 알레르기 등으로 섭취하지 못하는 농산물이 있다면 구독신청 하실 때 미리 요청사항에 작성해주세요.<br /> 신선하고 건강한 농산물이 선물처럼 찾아가겠습니다!~</Card.Text>
    <Image src="images/delivery.png" fluid />
    <Card.Text></Card.Text>
    <Button variant="danger">구독하러 가기!</Button>
  </Card.Body>
</Card>
```

자주 묻는 질문

자주 묻는 질문

01. UGV는 무슨 의미인가요?

저희 UGV(유지비, Ugly Vegetable)는 장동성온은 조금 떨어지지만, 신선하고 맛있는 못나에 과일과 채소를 판매합니다. UGV에는 못나에 채소라는 뜻과 1인 가구의 (생계) 유지비를 지키기 위한 서비스라는 뜻이 담겨있습니다. 혼자 생활하는 1인 가구가 지속적으로 증가하는 요즘, 비싼 농산품을 부담없는 가격에 합리적인 양만 구매할 수 있는 곳이 있었으면 좋겠다는 마음에 세 시작하게 되었습니다.

UGV는 주목받지 못하는 논과 밭, 농부와 농민들이 지속적으로 관심을 갖고 농작물의 진정한 가치를 알리기 위해 노력합니다. 상품성이 떨어지다는 이유로 폐기되는 농산물은 지구온난화를 가속시키고, 사회적, 경제적으로 큰 손실을 야기합니다. 불필요한 운송비와 소비자 사치와 연병에 따른 소비를 할 수 있으며, 농민들이 합리적인 노력의 결실을 볼 수 있는 **지속가능 소비 및 생산의 과정**에 참여합니다. 조금은 늦지만, 성장하고 있으며 건강하고 신진 사치와 가치를 여러모로 인식하게 되었습니다.

02. UGV에서 판매하는 농산물들은 왜 가격이 저렴한가요?

Q3. 교환 및 환불 기준에 어떻게 되나요?

Q4. 표정제는 외수에 기재나요?

Q5. 대량구매도 가능한가요?

Q6. 엘리베이터가 없는 고층 건물도 배수에 되나요?

```

<Container>
  <div className="margin">
    <div>
      <div className="accordion_wrap">
        {/} 다른 defaultActiveKey="a"를 설정하면 4번의 버튼으로 열려 있게 됩니다.
        열려지 상태, 닫기, 추가같은 기능 설정합니다. {/}
        <Accordion defaultActiveKey="a">
          <Accordion.Item eventKey="a">
            <Accordion.Header id, QW는 무슨 의미인가요? <Accordion.Header>
              <Accordion.Body>
                <Accordion.Item>
                  <Accordion.Item eventKey="1">
                    <Accordion.Header id, QW에서 변경하는 논문내용을 열 거예요 저글한가요? <Accordion.Header>
                      <Accordion.Body>
                    </Accordion.Body>
                  </Accordion.Item>
                  <Accordion.Item eventKey="2">
                    <Accordion.Header id, 교란 및 환경 기준이 어떻게 되나요? <Accordion.Header>
                      <Accordion.Body>
                    </Accordion.Body>
                  </Accordion.Item>
                  <Accordion.Item eventKey="3">
                    <Accordion.Header id, 포집되는 회수해 가시나요? <Accordion.Header>
                      <Accordion.Body>
                    </Accordion.Body>
                  </Accordion.Item>
                  <Accordion.Item eventKey="4">
                    <Accordion.Header id, 대정구마도 가능하나요? <Accordion.Header>
                      <Accordion.Body>
                    </Accordion.Body>
                  </Accordion.Item>
                </Accordion.Item>
              </Accordion.Item>
            </Accordion.Item>
          </Accordion>
        </div>
      </div>
    </div>
  </Container>
</div>

```

1) React Bootstrap의 아코디언 적용

09. 코드 리뷰 (Code Review)

문의하기 (Front)

1) 문의 내용을 입력하고 성공하면 문의사항이 등록됐다는 alert 창이 뜨게 되고, 실패했을 경우 문의사항 등록에 실패했다는 alert 창이 뜬다.

2) React Bootstrap의 Form 태그를 이용해서 Form 안의 내용을 입력하면 한 번에 같이 전송되도록 onSubmit() 사용.

```
JS Consultation.js U X
ugv_project_copy > src > components > JS Consultation.js > Consultation
17
20   });
21 }
22
23 const submitCon = (e) => {
24   e.preventDefault(); // submit이 action을 안타고 자기 할일을 그만함.
25   fetch("/Consultation",{
26     method:"Post",
27     headers:{
28       "Content-Type":"application/json; charset=utf-8"
29     },
30     body:JSON.stringify(con)
31   }).then(res=> {
32     console.log("정상", res);
33     if(res.status === 201){
34       return res.json();
35     } else{
36       return null;
37     }
38   }).then(res=> {
39     if(res !== null){
40       props.history.push("/");
41       alert("문의사항이 등록되었습니다.");
42     } else{
43       alert("문의사항 등록에 실패했습니다.");
44     }
45   });
46 });
47
48
```

localhost:3000 내용:
문의사항이 등록되었습니다.

확인

2. 수입 목적 : 문의자 확인, 문의에 대한 회신 등의 처리

3. 보유 기간 : 목적 달성 후 파기, 단, 관계법령에 따라 또는 회사 정책에 따른 정보보유사유가 발생하여 보존할 필요가 있는 경우에는 필요한 기간 동안 해당 정보를 보관합니다. 전자상거래 등에서의 소비자 보호에 관한 법률, 전자금융거래법, 통신비밀보호법 등 법령에서 일정한 기간 정보의 보관을 규정하는 경우, 이 기간 동안 법령의 규정에 따라 개인정보를 보관하며, 다른 목적으로는 절대 이용하지 않습니다. (개인정보처리방침 참고)

4. 귀하는 회사의 정보수집에 대해 동의하지 않거나 거부할 수 있습니다. 다만, 이때 원활한 문의 및 서비스 이용 등이 제한될 수 있습니다.

등록하기

```
<Form onSubmit={submitCon}>
  <Form.Group className="mb-3 formSmall" controlId="exampleForm.ControlInput1">
    <Form.Control type="text" name="author" onChange={changeValue} placeholder="이름을 입력하세요" />
  </Form.Group>
  <Form.Group className="mb-3 formSmall" controlId="exampleForm.ControlInput1">
    <Form.Control type="text" name="email" onChange={changeValue} placeholder="이메일을 입력하세요" />
  </Form.Group>
  <Form.Group className="mb-3" controlId="exampleForm.ControlInput1">
    <Form.Control type="text" name="title" onChange={changeValue} placeholder="제목을 입력하세요" />
  </Form.Group>
  <Form.Group className="mb-3" controlId="exampleForm.ControlTextarea1">
    <Form.Control as="textarea" name="content" onChange={changeValue} rows={3} placeholder="문의사항을 입력하세요" />
  </Form.Group>
  <div className="textBox">
    <Form.Check aria-label="option 1" />
    <p className="textHD">개인정보 수집 동의 </p>
    <span className="textContent">
      1. 수집하는 개인정보 항목 : 이름, 이메일 <br/><br/>
      2. 수집 목적 : 문의자 확인, 문의에 대한 회신 등의 처리 <br/><br/>
      3. 보유 기간 : 목적 달성 후 파기, 단, 관계법령에 따라 또는 회사 정책에 따른 정보보유사유가 발생하여 보존할 필요가 있는 경우에는 필요한 기간 동안 해당 정보를 보관합니다. 전자상거래 등에서의 소비자 보호에 관한 법률, 전자금융거래법, 통신비밀보호법 등 법령에서 일정한 기간 정보의 보관을 규정하는 경우, 이 기간 동안 법령의 규정에 따라 개인정보를 보관하며, 다른 목적으로는 절대 이용하지 않습니다. (개인정보처리방침 참고) <br/><br/>
      4. 귀하는 회사의 정보수집에 대해 동의하지 않거나 거부할 수 있습니다. 다만, 이때 원활한 문의 및 서비스 이용 등이 제한될 수 있습니다.
    </span>
  </div>

  <div className="d-grid gap-2">
    <Col md={{ span: 6, offset: 3 }}>
      <Button variant="primary" type="submit" size="lg" className="conButton">
        등록하기
      </Button>
    </Col>
  </div>
</Form>
```

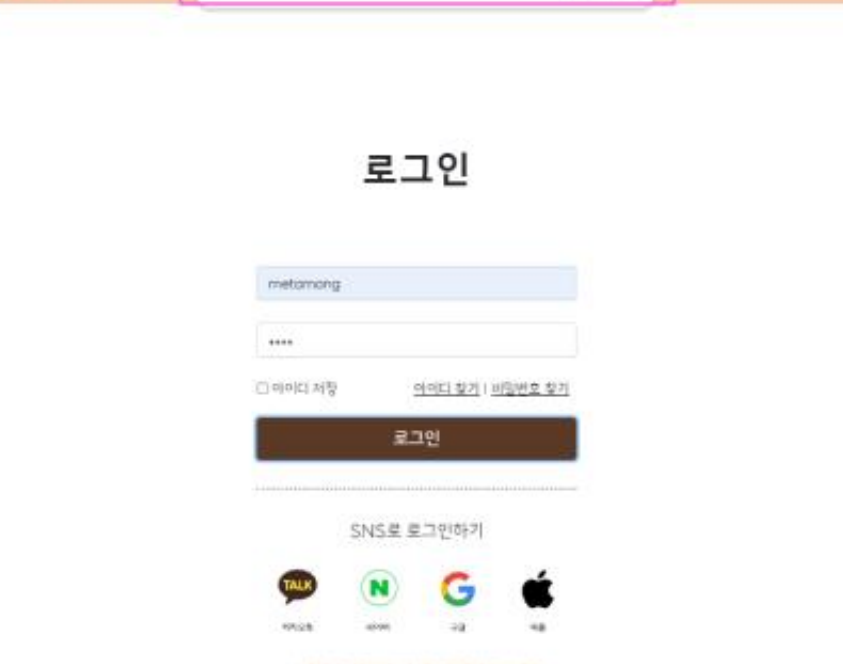

09. 코드 리뷰 (Code Review)

로그인 (FrontEnd)

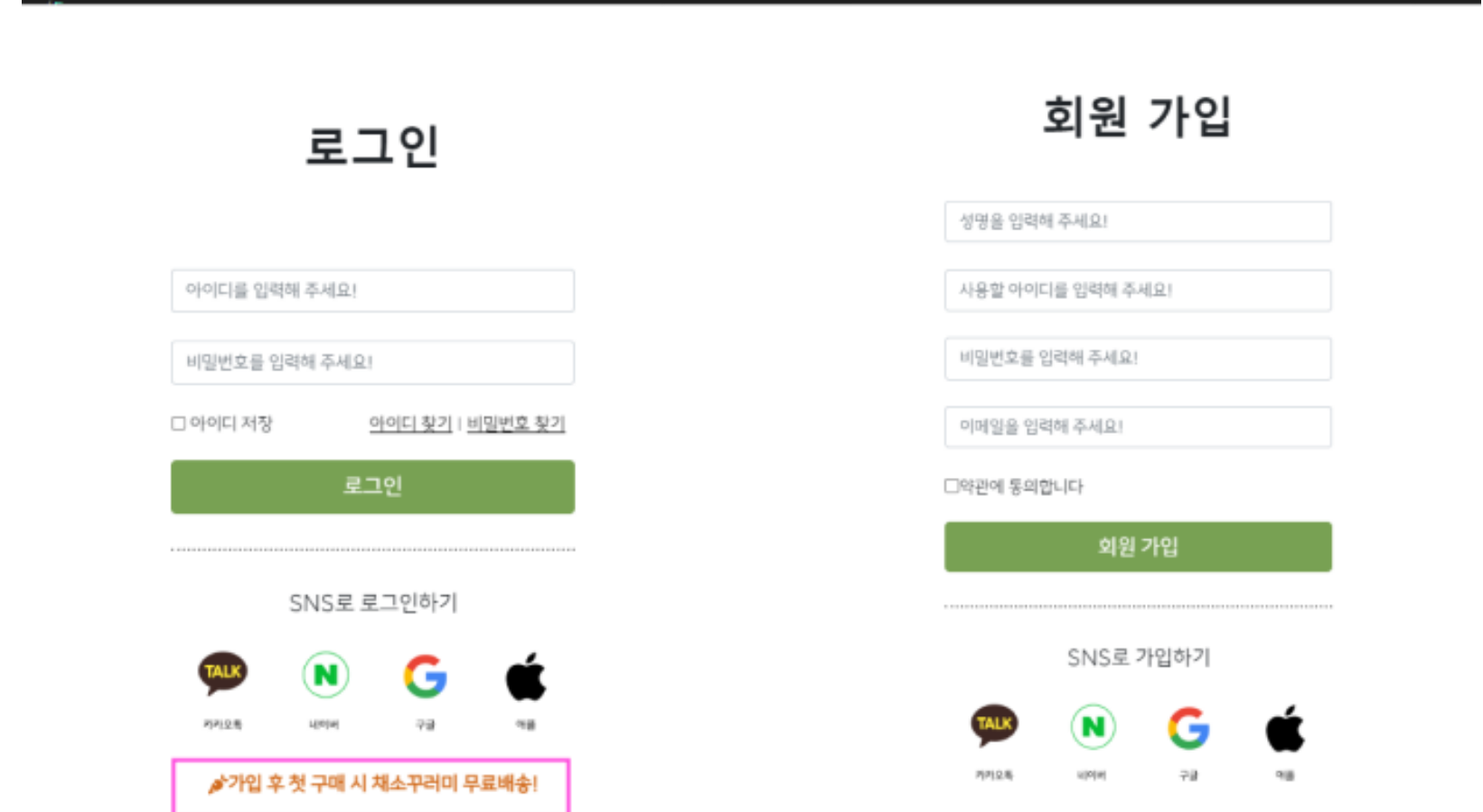
1) 로그인 시 회원정보 유무에 따라 모달창의 내용이 바뀜.

2) 로그인 창에서 하단 글씨를 클릭하면 회원가입 창으로 연결됨.

```
14
15 const Login = () => {
16
17   const fn_signIn = () => {
18     var submitYN = false;
19
20     const memberId = document.getElementById('memberId').value;
21     const memberPw = document.getElementById('memberPw').value;
22
23     if(document.getElementById("memberId").value.length < 1){
24       alert("아이디를 입력해 주세요!");
25       document.getElementById("memberId").focus();
26       return;
27     }
28
29     if(document.getElementById("memberPw").value.length < 1){
30       alert("비밀번호를 입력해 주세요!");
31       document.getElementById("memberPw").focus();
32       return;
33     }
34
35     if(!submitYN){
36       var submitYN = true;
37
38       const form = new FormData();
39       form.append('memberId', memberId);
40       form.append('memberPw', memberPw);
41
42       console.log("click login");
43       console.log(memberId);
44       console.log(memberPw);
45
46
47
48
49       axios.post('http://localhost:9008/loginAction.do',
50         form,{
51           headers:{
52             'content-type': 'application/json',
53           },
54         },
55       )
56       .then((res) => {
57         console.log(res.data);
58         if(res.data != "N"){
59           alert(res.data + " 회원님, 환영합니다!");
60           // sessionStorage에 id 저장 처리함
61           sessionStorage.setItem("member_id", memberId);
62           // sessionStorage에 name 저장 처리함
63           sessionStorage.setItem("name", res.data);
64           document.location.href="/";
65         } else{
66           alert("회원 정보가 없습니다!");
67           // 회원 가입 컴포넌트를 만들고, url을 /signup 설정했을 경우
68           // alert("회원 정보가 없습니다. 회원 가입을 해주시기 바랍니다.");
69           // document.location.href="/signup";
70         }
71       })
72       .catch()
73     }
74   }
75 }
```



```
</div>
<div className="btnCarrot">
<a className="btnBottom" href="/signup"><FontAwesomeIcon icon={faCarrot} />가입 후 첫 구매 시 채소꾸러미 무료배송!</a>
</div>
```



09. 코드 리뷰 (Code Review)

로그인
(BackEnd)

```
@RequestMapping(value = "/loginAction.do")
public @ResponseBody String loginAction(@ModelAttribute("memberVO") MemberVO memberVO, HttpServletRequest request, HttpServletResponse response, HttpSession ses
String result = "N";

try {
    MemberVO resultVO = memberService.selectMemberByIdByPw(memberVO);
    System.out.println("로그인 -----");
    // 멤버의 정보가 있다면, 멤버의 이름을 리턴 처리합니다.
    if(resultVO != null) {
        System.out.println("로그인 합니다-----");
        result = resultVO.getMemberName();
    }
} catch (Exception e) {
    e.printStackTrace();
}
return result;
}
```

1) 로그인 버튼을 클릭하면, 아이디 값과 비밀번호값을 Springboot로 넘겨서 정보가 일치하면 "ooo 회원님 환영합니다" 라는 alert 창이 뜬.

localhost:3000 내용:
채승영 회원님, 환영합니다!

확인

문의하기
(Back)

```
1 package com.springboot.project.domain;
2
3 import javax.persistence.Column;
12
13 @SequenceGenerator(
14     name = "CONSULTATION_IDX_SEQ_GEN",
15     sequenceName = "CONSULTATION_IDX_SEQ",
16     initialValue = 1,
17     allocationSize = 1
18 )
19
20 @Data
21 @Entity
22 @Table(name = "consultation")
23 public class Consultation {
24
25     @Id // pk를 해당 변수로 하겠다.
26     @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "CONSULTATION_IDX_SEQ")
27     @SequenceGenerator(name = "CONSULTATION_IDX_SEQ", sequenceName = "CONSULTATION_IDX_SEQ_GEN", allocationSize = 1)
28     @Column(name = "bno")
29     private Long bno;
30
31     @Column(name="title", nullable=false)
32     private String title;
33
34     @Column(name="author", nullable=false)
35     private String author;
36
37     @Column(name="email")
38     private String email;
39
40     @Column(name="content", nullable=false)
41     private String content;
42
43 }
44
```

1) Springboot에서 테이블 생성

로그인

chae

☐ 아이디 저장

[아이디 찾기](#) | [비밀번호 찾기](#)

로그인

SNS로 로그인하기



카카오톡



네이버



구글



애플

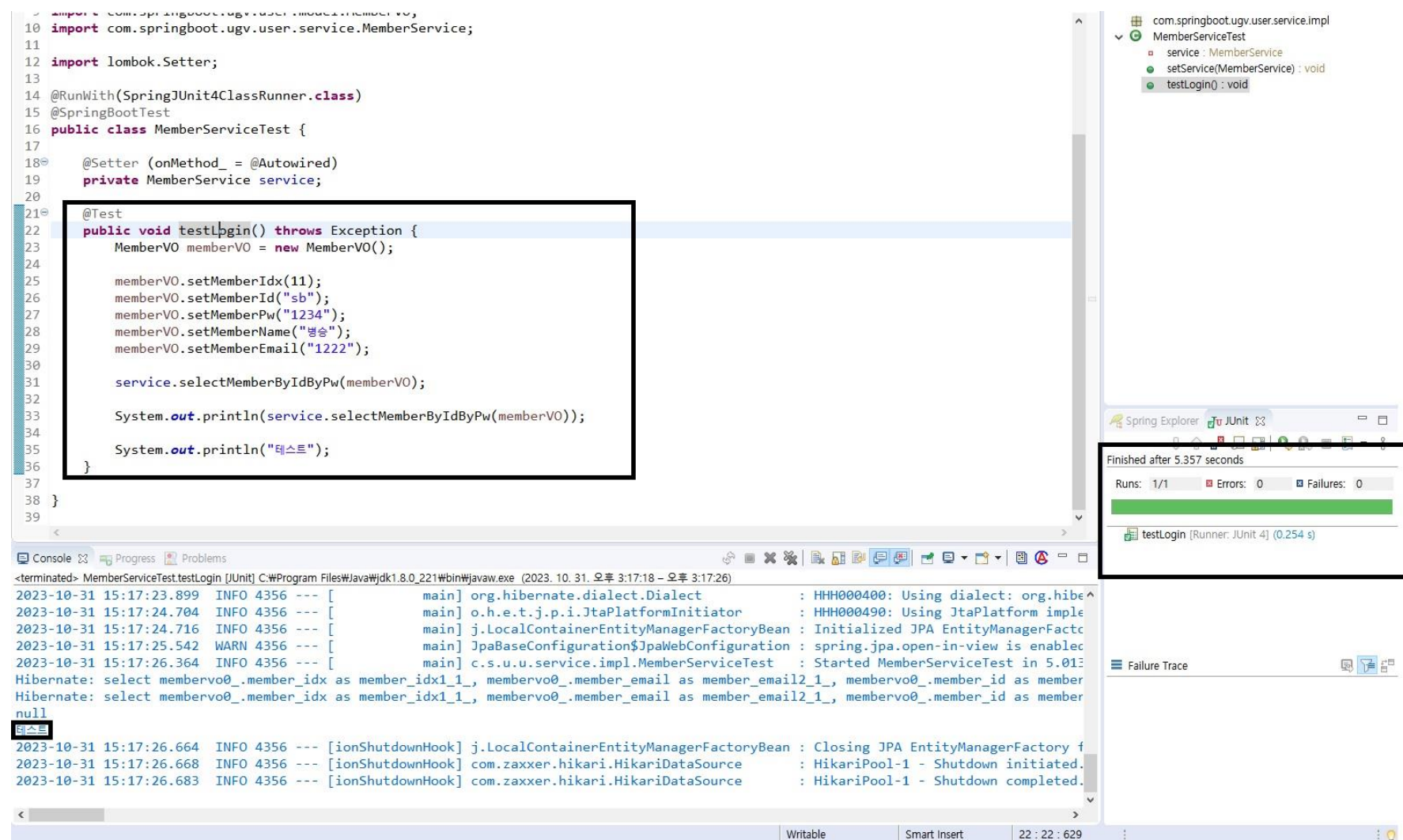
09. 코드 리뷰 (Code Review)

로그인
(BackEnd)

```
@RequestMapping(value = "/loginAction.do")
public @ResponseBody String loginAction(@ModelAttribute("memberVO") MemberVO memberVO, HttpServletRequest request, HttpServletResponse response, HttpSession ses
String result = "N";

try {
    MemberVO resultVO = memberService.selectMemberByIdByPw(memberVO);
    System.out.println("로그인 -----");
    // 멤버의 정보가 있다면, 멤버의 이름을 리턴 처리합니다.
    if(resultVO != null) {
        System.out.println("로그인 합니다-----");
        result = resultVO.getMemberName();
    }
} catch (Exception e) {
    e.printStackTrace();
}
return result;
}
```

로그인 Service 단 Juit 테스트 결과



localhost:3000 내용:
채승병 회원님, 환영합니다!

확인

로그인

☐ 아이디 저장 [아이디 찾기](#) | [비밀번호 찾기](#)

로그인

SNS로 로그인하기



1) 로그인 버튼을
클릭하면, 아이디 값과
비밀번호 값을
Springboot로 넘겨서
정보가 일치하면
“ooo 회원님
환영합니다” 라는 alert
창이 뜬.

감사합니다 😊