

# 캡스톤 디자인 I 종합설계 프로젝트

프로젝트 명	EZOrder
팀 명	EZOrder - 32
문서 제목	중간보고서

Version	1.3
Date	2024-03-31

팀원	박민서 (팀장)
	문원기
지도교수	이창우 교수

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder – 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 소프트웨어융합대학 소프트웨어학부 및 소프트웨어학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 "EZOrder"를 수행하는 팀 "32조"의 팀원들의 자산입니다. 국민대학교 소프트웨어학부 및 팀 "32조"의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역


<b>Filename</b>	중간보고서-EZOrder.doc
<b>원안작성자</b>	박민서, 문원기
<b>수정작업자</b>	박민서, 문원기

수정날짜	대표수정자	Revision	추가/수정 항목	내 용
2024-03-30	박민서	1.0	최초 작성	프로젝트 기획 배경/서비스 대상 분석 일부/주요기능 명세/프론트 기술 스택 선정 작성
2024-03-30	문원기	1.1	내용 추가	프로젝트 개요/서비스 대상 분석 일부/시스템 아키텍처 설계/MVC 패턴 적용 작성
2024-03-31	문원기	1.2	내용 추가	구현 진척도/향후 추진 계획/고충 및 건의사항 작성
2024-03-31	박민서	1.3	내용 추가	구현 진척도 내용 추가/고충 및 건의사항 추가

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder – 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

## 목 차

1	프로젝트 목표 .....	4
1.1	프로젝트 개요 .....	4
1.2	프로젝트 기획 배경 .....	5
1.3	유사 서비스 분석 .....	6
1.4	서비스 대상 유저 분석 .....	7
2	수행 내용 및 중간결과 .....	9
2.1	주요 기능 명세 .....	9
2.2	수행내용 .....	11
2.2.1	시스템 아키텍처 설계 .....	11
2.2.2	프론트 기술 스택 선정 .....	12
2.2.3	MVC 패턴의 적용 .....	13
2.2.4	구현 진척도 .....	14
3	향후 추진계획 .....	16
3.1	향후 계획의 세부 내용 .....	16
4	고충 및 건의사항 .....	17

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	EZOrder	
	팀 명	EZOrder - 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

# 1 프로젝트 목표

## 1.1 프로젝트 개요

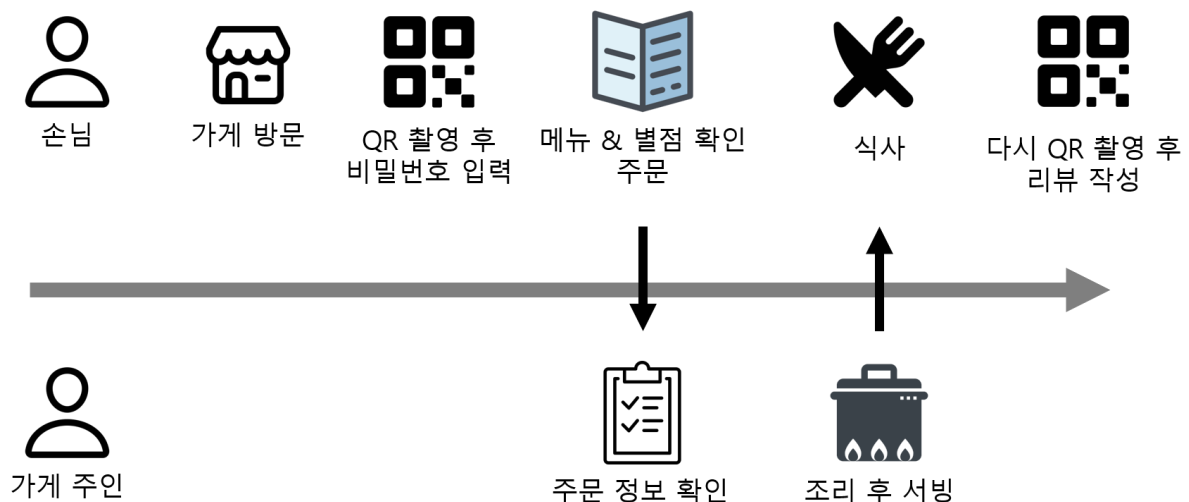
EZOrder는 음식점에서 테이블에 부착된 QR 코드를 통해 간편하게 주문을 할 수 있는 웹 서비스이다.


이 서비스의 목표는 사용자가 모바일을 통해 빠르고 편리하게 주문할 수 있는 시스템을 제공하는 것이다.

손님들은 테이블에 부착된 QR 코드와 가게에 부착된 인증 번호를 이용하여 자신의 스마트폰으로 메뉴에 바로 접근할 수 있다. 이를 위해 로그인이나 앱 다운로드와 같은 번거로운 절차는 필요하지 않다.

메뉴를 확인하고 리뷰 및 별점을 살펴본 후, 손님들은 원하는 음식을 주문할 수 있다. 가게 주인은 이러한 주문을 '주문 대기열' 페이지를 통해 즉각적으로 확인할 수 있다.

음식을 맛본 후에는 손님들은 자신의 경험을 리뷰로 남길 수 있다. 테이블에 부착된 QR 코드를 통해 리뷰 페이지에 접근할 수 있으며, 작성된 리뷰는 메뉴에 반영되어 다른 손님들이 확인할 수 있다.



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder - 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

## 1.2 프로젝트 기획 배경

2019년부터 시작된 코로나19 사태는 우리의 삶을 크게 변화시켰다. 비대면이 필수가 되는 시대로 바뀌면서 키오스크는 우리의 일상에 자리를 잡게 되었다.

자영업자들은 임대료와 최저임금 증가 등으로 어려움을 겪게 되면서, 편리하고 효율적인 수단을 찾게 되었다. 이로 인해 키오스크의 사용량은 계속해서 증가하고 있다. 이전에는 주로 패스트푸드점 등에 설치되어 있던 키오스크가 이제는 일반 음식점에서도 자주 볼 수 있게 되었다.

테이블 오더는 키오스크의 다른 형태로, 테이블에 앉아서 메뉴판 없이도 태블릿 PC를 통해 직접 주문할 수 있는 장점이 있다. 특히 매장이 넓고 테이블이 많은 업체에서는 테이블 오더를 점점 더 많이 사용하고 있다. 이는 고객 주문을 누락하지 않고 인건비를 절감하는 데에 도움이 된다.

하지만 키오스크와 테이블 오더의 기기 비용이 상당히 높은 편이다. 키오스크는 평균적으로 한 대당 200~500만원으로 형성되어 있으며, 대형 기기의 경우 1000만원을 넘기도 한다.

기기 당 300만원으로 가정할 때, 2대를 설치하면 종업원 1명의 월급 3개월치가 손익분기점이다. 테이블 오더의 경우에도 평균적으로 태블릿 한 대당 40~50만원 정도로 비용이 발생한다. 매장 내 테이블이 20개 있다면 설치 비용만 800~1000만원에 달할 것이다.

비용은 초기에만 부담할 수 있다면 괜찮을 것으로 생각할 수 있지만, 기기에 결함이 생기거나 유지 보수가 필요할 경우 비용이 더 많이 발생할 수 있다.

이러한 문제점과 경제적 부담을 고려하여 회원가입만으로도 사용 가능한 QR 스캔을 통한 주문 "웹 서비스"를 제공하고자 한다. 이를 통해 기기 설치 비용과 유지 보수 비용 등에 대한 문제를 해결하고자 한다.

### 1.3 유사 서비스 분석

비슷한 방식의 QR 주문을 제공하는 서비스들은 시중에 많이 나와있으나, QR에서 비롯된 문제점을 해결하는 방법에 있어서는 차이를 보였다.


QR 주문에서 마주할 수 있는 가장 큰 보안 문제로는 'QR 재접근' 문제가 있다. 이는 악성 사용자가 특정한 가게의 QR 코드를 저장한 후에 매장 밖에서 악성 주문을 날리는 경우이다. 이렇게 되면 가게에서는 실제로는 존재하지 않는 손님을 위해 음식을 조리하며, 실제 손님의 주문이 밀리게 되는 등 운영에 큰 지장을 입게 된다. 위 예시처럼 QR 코드 기술의 취약점을 이용한 다양한 문제들이 존재하며, 시장에서는 이러한 문제를 해결하기 위해 다양한 솔루션을 도입하고 있었다.

그 중 가장 대표적인 '네이버 QR 주문'과 '핸드 오더'의 경우를 비교해보았다.

네이버 QR 주문의 경우, 네이버 계정 연동이 필수적이다. 네이버 계정이라는 단단한 보안 시스템을 통해 문제점을 해결하고, 또한 네이버 페이 시스템과 연계되어 결제까지 가능하다는 장점이 있다. 하지만 당장 네이버 페이를 사용하지 않는 사용자의 경우, 결국에는 다시 대면으로 주문을 진행해야 한다는 단점이 있다.

'핸드 오더'의 경우, 위변조와 복사가 불가능한 디지털 QR 장비를 도입하여 이를 해결하였다. 다만 이는 QR 코드를 표시하기 위한 특수 디지털 장비가 필요하다는 단점이 존재한다.



 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder - 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

## 1.4 서비스 대상 유저 분석

'EZOrder의 서비스 대상은 가게를 운영하는 점주와 해당 가게를 이용하는 손님으로 나뉜다.

먼저 손님 측의 입장에서 이러한 디지털 주문 서비스에 어느 정도의 익숙함을 가지고 있는지가 매우 중요하기에, 이에 관하여 분석을 해보았다.'

EZOrder 서비스는 모바일을 통한 QR 주문을 핵심 기능으로 삼고 있다. 따라서 음식점 손님, 소비자들의 QR에 대한 인식과 현 시점의 스마트폰 보급률이 상당히 중요한 요소로 작용할 것으로 예측된다.

코로나 시대 때, 타인과의 접촉을 최소화하는 언택트 활동의 비율이 높아지면서 많은 사람들이 QR과 같은 디지털 정보 전달 매체에 대해 긍정적으로 인식하고 있는 것으로 보인다. 가까운 예시로 질병관리청에서 진행한 코로나19 예방접종 증명 방법 안내에서도 QR 코드를 통한 인증을 내세우고 있다.

코로나 시대 때 이루어진 경제 활동은 디지털 기기를 활용한 비대면 소비가 주를 이루었는데, 이 당시 서울연구원에서 2020년에 발표한 "2/4분기 서울시 소비자 체감경기와 비대면 경제" 리포트에 따르면, 당시 이루어진 비대면 소비 경험 분야 중 54%가 음식 관련 분야인 것으로 드러났다.

EZOrder 서비스 역시 디지털 기기를 활용한 음식 주문 서비스로서, 코로나 시절에 디지털 기기를 활용한 음식 소비 활동에 주체적으로 참여한 소비자들이 거리감 없이 접근할 수 있는 환경을 제공할 것으로 기대된다.

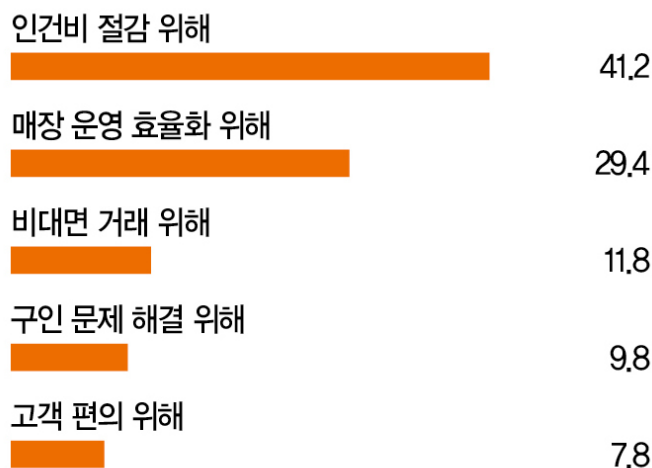
또한 한국 갤럽이 발표한 '2012-2023 스마트폰 사용률 & 브랜드, 스마트워치, 무선이어폰에 대한 조사'에 따르면 나이에 관계없이 많은 사람들이 스마트폰을 가지고 있는 것으로 나타났다. 해당 조사에 따르면 60대 노인층의 경우 98%, 70대 이상도 85%의 비율로 스마트폰을 사용하고 있다고 한다. 따라서 EZOrder 서비스는 넓은 연령층에게 접근 가능한 플랫폼을 제공할 수 있을 것으로 기대된다.

다음으로 'EZOrder' 서비스의 도입을 결정하는 주체인 음식점을 운영하는 점주들을 기준으로 분석해보았다.

한국농촌경제연구원이 외식업체를 대상으로 실시한 '푸드테크 사용의향 조사' 결과에 따르면 '인건비 절감(41.2%)', '매장 운영 효율화(29.4%)' 등이 무인주문기 사용의향 이유로 선정됐다. 최저임금의 상승으로 인해 인건비 부담이 커지고 있고, 고된 일을 꺼리는 추세이다 보니 노동 시장에서 일할 인력을 구하기 어렵다고 한다. 점주들 입장에서는 인력 관리가 항상 고민거리이기 때문에 인력을 줄이는 방법을 찾을 수밖에 없다고 한다.

#### ■푸드테크 기술별 사용의향 이유 (단위 : %)

##### [무인주문기]



[자료 | 한국농촌경제연구원 외식업체 조사, 참고 | 2022년 기준]

하지만 앞서 프로젝트 기획 배경에서 언급하였듯이 무인주문기를 도입하는 것은 기기 설치비, 유지 보수비, 여기에 추가적으로 서비스 이용료 및 수수료 등의 경제적 부담이 따르게 된다. 인건비를 줄이기 위해 이러한 무인주문기 서비스를 사용하려던 점주들의 우려가 생겨나고 있다.

'EZOrder'는 QR 스캔을 통한 주문 방식과 회원가입 후 로그인만 하면 웹에서 바로 사용가능하다는 점을 통해 서비스 이용에 대한 점주들의 이러한 경제적인 부담을 줄이고자 한다.



## 2 수행 내용 및 중간결과


### 2.1 주요 기능 명세

- 회원 관리

페이지	세부 기능	기능 설명
로그인 페이지	로그인	<ul style="list-style-type: none"> <li>아이디, 비밀번호를 입력 받아 로그인한다.</li> </ul>
	회원가입 리다이렉트	<ul style="list-style-type: none"> <li>회원가입 페이지로 이동한다.</li> </ul>
회원가입 페이지	회원가입	<ul style="list-style-type: none"> <li>아이디, 비밀번호, 이름을 입력받는다.</li> <li>중복 아이디 검증을 한다.</li> </ul>
회원정보관리 페이지	회원정보 변경	<ul style="list-style-type: none"> <li>가게 이름, 주소, 사진을 변경한다.</li> <li>계정 비밀번호를 변경한다.</li> </ul>

- 테이블 관리

페이지	세부 기능	기능 설명
테이블 관리 페이지	테이블 추가, 삭제	<ul style="list-style-type: none"> <li>매장 내에서 운영하고 있는 테이블을 추가, 삭제한다.</li> </ul>
	테이블 이름 변경	<ul style="list-style-type: none"> <li>테이블 이름을 변경한다.</li> </ul>
	테이블 별 QR 코드 생성	<ul style="list-style-type: none"> <li>각 테이블과 매칭되는 QR코드를 생성한다.</li> </ul>
테이블 현황 페이지	테이블 상태 확인	<ul style="list-style-type: none"> <li>어느 테이블이 이용 중인지, 이용 가능한 지 확인할 수 있다.</li> <li>각 테이블의 주문내역을 확인할 수 있다.</li> </ul>
	테이블 결제	<ul style="list-style-type: none"> <li>테이블을 선택하여 해당 테이블의 주문서를 출력할 수 있다.</li> <li>주문서 출력을 통해 결제를 진행한다. (후불 결제)</li> <li>주문서 출력을 하면 해당 테이블은 이용 가능 상태가 된다.</li> </ul>

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder – 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

● 메뉴 관리


페이지	세부 기능	기능 설명
메뉴 관리 페이지	메뉴 추가, 삭제	<ul style="list-style-type: none"> <li>메뉴를 추가, 삭제한다.</li> <li>메뉴 설명을 적을 수 있으며, 사진을 업로드할 수 있다.</li> </ul>
	메뉴 리뷰 확인	<ul style="list-style-type: none"> <li>각 메뉴에 대한 손님의 별점과 리뷰를 확인할 수 있다.</li> </ul>

● 주문 시스템 (점주 기준)

페이지	세부 기능	기능 설명
주문 대기열 확인 페이지	주문 확인	<ul style="list-style-type: none"> <li>주문을 주문서 형태로 볼 수 있다.</li> <li>대기열의 형태로 먼저 주문한 순서로 볼 수 있다.</li> <li>각 주문이 어떤 테이블에서 주문됐는지 알 수 있다.</li> </ul>
	주문 상태 반영	<ul style="list-style-type: none"> <li>주문의 상태는 접수-진행-완료로 나뉜다.</li> <li>주문 접수 상태에서는 주문을 수정할 수 있다.</li> <li>주문 진행 상태로 변경할 수 있다.</li> <li>진행 상태로 변경되면 더 이상 해당 주문은 수정할 수 없다.</li> <li>음식이 준비되면 주문 완료 상태로 변경한다.</li> </ul>

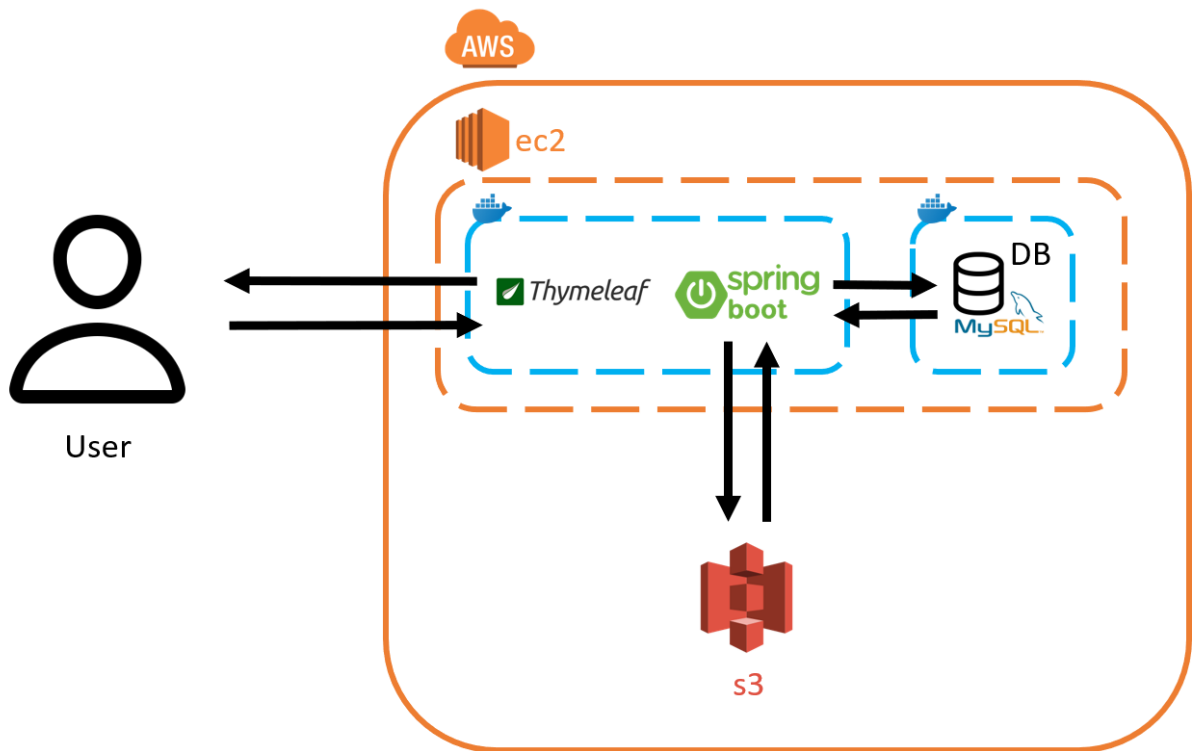
● 주문 시스템 (손님 기준)

페이지	세부 기능	기능 설명
주문 페이지	주문하기	<ul style="list-style-type: none"> <li>원하는 메뉴를 선택하여 주문할 수 있다.</li> </ul>
	리뷰보기/작성하기	<ul style="list-style-type: none"> <li>각 메뉴의 별점과 리뷰를 볼 수 있다.</li> <li>실제로 메뉴를 주문한 손님만 리뷰를 작성할 수 있다.</li> <li>메뉴를 주문한 지 2시간 이내에만 작성할 수 있다.</li> </ul>

 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	EZOrder	
	팀 명	EZOrder - 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

## 2.2 수행내용


### 2.2.1 시스템 아키텍처 설계



EZOrder의 시스템은 위와 같이 구성했다.

AWS의 EC2 서버에 Docker를 통해 2개의 컨테이너를 올릴 계획이다. 백엔드 프레임워크로는 Spring Boot를 사용할 생각이며, 다른 컨테이너에 올려진 MySQL을 DB로 사용할 것이다. 또한 음식 메뉴의 사진을 올리는 기능을 위해 AWS S3를 사용할 것이다. 팀의 기술적 한계로 인해 프론트엔드 프레임워크는 사용하지 못하지만, 그것을 대체할 수단으로 Thymeleaf 템플릿 엔진을 사용할 생각이다. 해당 엔진을 통해 백엔드에서 직접 HTML과 JS를 사용자에게 전달할 것이다. Thymeleaf의 자세한 설명은 2.2.2에 자세히 설명되어 있다.

추가적으로 주문 대기열 페이지와 같이 즉각적인 반응이 중요한 페이지에 한정하여 WebSocket 기술을 활용할 계획이다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder – 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

## 2.2.2 프론트 기술 스택 선정


본 팀은 주로 백엔드 기술 스택을 중심으로 학습해왔다. 따라서 새로운 프론트엔드 프레임워크를 익히고 능숙하게 사용하는 데에는 상당한 시간이 소요될 것으로 예상했다. 따라서 프론트엔드 프레임워크를 도입하는 대신 스프링과 연계해서 사용 가능한 Thymeleaf 템플릿 엔진을 선택했다.

Thymeleaf를 선택한 이유 중 하나는 스프링과의 호환성이다. Thymeleaf는 스프링 프레임워크와의 통합이 용이하며, HTML을 템플릿으로 사용하고 서버 측에서 데이터를 동적으로 렌더링할 수 있는 기능을 제공한다. 이로써 스프링 애플리케이션의 백엔드와 프론트엔드를 간편하게 연결할 수 있다.

또한 Thymeleaf는 HTML과 자바 코드를 혼합하여 사용할 수 있는 템플릿 엔진으로, HTML 파일을 그대로 유지하면서도 서버 측에서 데이터를 쉽게 삽입하고 제어할 수 있다.

이러한 이유들로 Thymeleaf를 선택했으며, 스프링 프레임워크에 익숙한 우리 팀에게는 좋은 선택이라고 판단된다.

Thymeleaf를 사용함으로써 팀원들이 프론트엔드 개발에 필요한 추가적인 학습과 익숙해질 필요가 없으며, 프로젝트 진행에 더욱 집중할 수 있다.

 <b>국민대학교</b> <b>소프트웨어학부</b> <b>캡스톤 디자인 I</b>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder – 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

### 2.2.3 MVC 패턴의 적용

Spring Boot를 백엔드 프레임워크로 선정하면서 해당 프레임워크가 제시하는 MVC 패턴을 적극적으로 활용할 계획이다.

MVC(모델-뷰-컨트롤러) 패턴은 소프트웨어 디자인 패턴 중 하나로, 소프트웨어의 구성 요소를 모델, 뷰, 컨트롤러 세 가지 역할로 나누어 구현하는 것이다. 각 구성 요소는 역할에 따라 서로 다른 책임을 가지며, 애플리케이션의 구조를 분리하여 유지보수성과 확장성을 향상시킨다.

모델(Model)은 애플리케이션의 데이터와 비즈니스 로직을 담당한다. 데이터의 상태를 표현하고, 데이터를 조작하거나 처리하는 동작을 수행한다. EZOrder 서비스에서는 ORM 방식 중 JPA를 활용하여 자바 코드로 작성된 Entity 클래스를 데이터베이스와 연동시킨다.

뷰(view)는 사용자가 보는 부분으로, 데이터를 시각적으로 표현하고 사용자와의 상호작용을 담당한다. HTML, XML, JSON의 형식으로 데이터를 표시하거나 사용자 입력을 받으며, 서비스 안에서 일어나는 변화를 업데이트하여 사용자에게 보여준다. EZOrder 서비스에서는 Thymeleaf 템플릿 엔진을 활용하여 사용자에게 완성된 HTML과 JS, CSS를 제공한다.

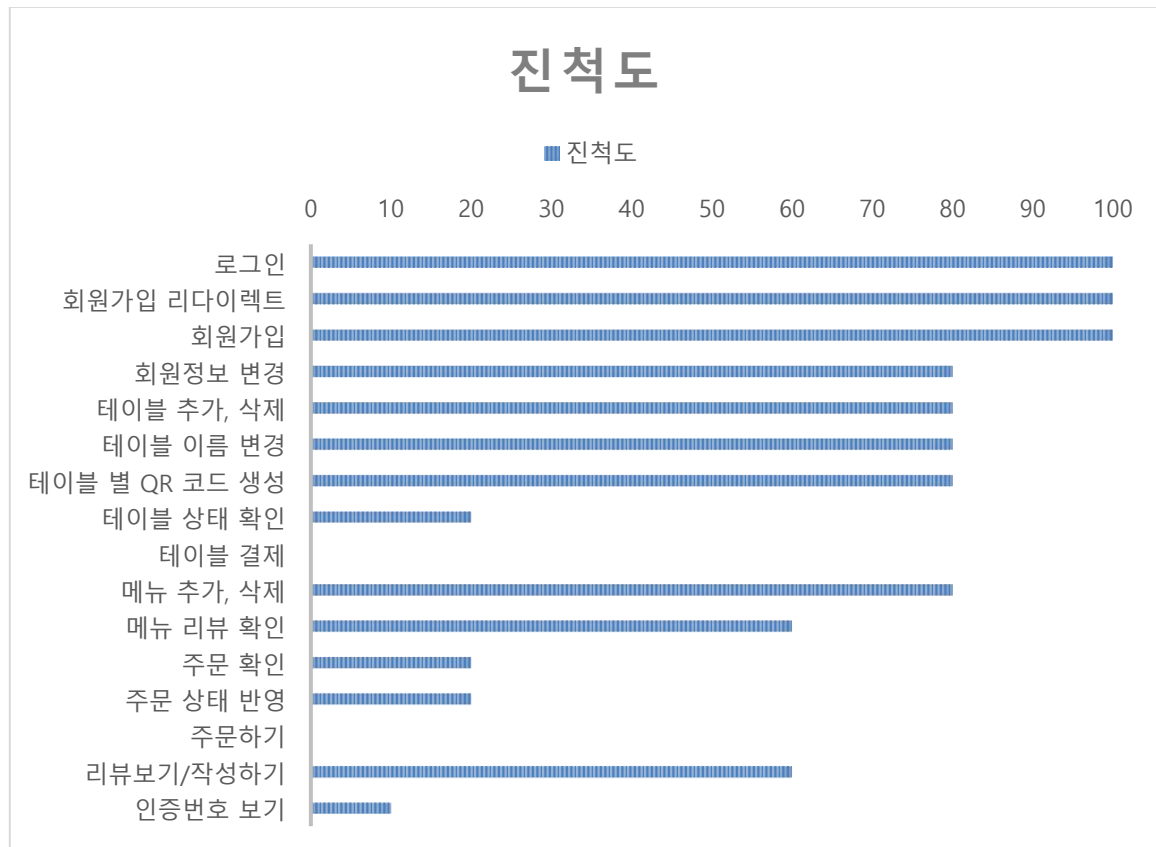
컨트롤러(Controller)는 모델과 뷰 사이의 상호 작용을 조정하고 사용자의 입력을 처리한다. 사용자의 요청을 받아 데이터를 처리하고, 그 결과를 적절한 뷰에 전달하여 화면을 갱신한다. 컨트롤러는 사용자의 요청을 해석하고 그에 따른 작업을 수행하는데, EZOrder 서비스에서는 Service라는 추가적인 단계를 적용하여 이를 더 세밀하게 분리하여, 컨트롤러 단계에서는 입력과 출력에 관련된 부분만 남기고자 한다.

모델에 직접적으로 영향이 가해지는 로직은 전부 Service 단계로 분리하였다. 이는 모델이 DB에 직접적으로 JPA를 통해 연결되어 있는 상태에서, 모델을 변경하는 로직이 상대적으로 수정이 잦은 Controller에 있으면 위험하다는 판단 하에 이루어졌다. 또한 모델에 이루어지는 작업은 CRUD와 같이 공통적인 부분이 많기 때문에, 코드 재사용을 높여서 생산성을 향상시키기 위해 Service를 분리하였다.

## 2.2.4 구현 진척도

구현 기준	진척도
구현 방향에 대한 계획 설계가 되어있는가?	10%
엔티티를 구현하고 JPA 연결이 되어 있는가?	20%
해당 엔티티를 가공할 수 있는 기본적인 서비스가 구현되어 있는가?	40%
서비스에서 컨트롤러에 전달할 DTO의 양식이 결정되고 구현되어 있는가?	60%
컨트롤러의 URL 매핑까지 완료되어 있는가?	80%
사용자에게 전달할 Html과 디자인이 완료되어 있는가?	100%

EZOrder 서비스 구현을 하면서 진척도에 대한 정확한 판단을 위해 위와 같은 측정 기준을 정하였다. 측정 기준은 위에서 서술된 MVC 패턴의 구조도를 기반으로 결정하였다.



 <div> 국민대학교  소프트웨어학부  캡스톤 디자인 I </div>	중간보고서		
	프로젝트 명	EZOrder	
	팀 명	EZOrder – 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

현재까지 완성되어 있는 화면

- 회원가입, 로그인

←

회원가입

로그인

아이디 \*

비밀번호 \*

비밀번호 확인 \*

이름 \*

다음


아이디 \*  
hightuv

비밀번호 \*  
.....

로그인

계정이 없으신가요? [회원가입](#)

- 메인페이지


EZOrder

Q Search...

🏠 마이페이지

✏️ 메뉴 관리

📦 주문 현황

🕒 매출 확인


♥ 리뷰 관리

🚗 테이블 관리

🖨 프린터 설정

👤 로그아웃

🌙 다크모드

 <div> <b>국민대학교</b>  <b>소프트웨어학부</b>  <b>캡스톤 디자인 I</b> </div>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder – 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

## 3 향후 추진계획


### 3.1 향후 계획의 세부 내용

현재 주요 기능 중에서 구현되지 않은 부분은 '인증번호', '주문 대기열 확인', '테이블 현황 확인' 등이 있다.

위에서 언급한 3가지 기능은 모두 EZOrder 서비스의 핵심 기능들이며, 해당 내용의 구현이 잘 못되었을 경우 전체적인 서비스 이용이 불가능하다. 따라서 서비스의 견고함을 목표로 '검증'과 '에러 처리'을 위주로 긴 시간을 투자할 생각이다.

특히 인증번호와 주문 대기열 페이지 같은 경우, 변동에 따른 화면 변화를 사용자에게 즉각적으로 전달하는 것이 중요하다. 따라서 WebSocket 기술을 활용하여 해당 기능을 구현할 계획이며, 그 부분에 대한 추가적인 팀원들의 연구가 필요할 것으로 보인다.



 <div> <b>국민대학교</b>  <b>소프트웨어학부</b>  <b>캡스톤 디자인 I</b> </div>	<b>중간보고서</b>		
	<b>프로젝트 명</b>	EZOrder	
	<b>팀 명</b>	EZOrder – 32	
	Confidential Restricted	Version 1.3	2024-MAR-31

## 4 고충 및 건의사항

### 1) Websocket 구현 관련 이슈

WebSocket 기술을 제대로 다뤄본 경험이 없다는 점이 문제로 여겨진다. 단순한 구조의 웹 서비스에서는 경험이 있지만, 보안 인증 절차가 존재하는 서버에서의 구현 경험이 부족하여 상당히 우려스러운 상황이다. 그러나 다행히도 WebSocket 기술은 상당히 널리 사용되고 있고 관련 레퍼런스도 많기 때문에, 문제를 극복할 수 있을 것으로 기대된다.

### 2) 프론트엔드에 대한 이슈

팀이 전반적으로 프론트엔드에 익숙하지 않다보니, 간단한 HTML, CSS, JS를 다룰 줄 아는 정도이다. Thymeleaf는 기본적인 HTML 지식만 있으면 사용하기 어렵지 않기 때문에 팀에 큰 도움이 될 것으로 예상된다. 우선은 Thymeleaf를 활용하여 간단한 HTML 화면을 만들어서 백엔드에서 구현한 기능들이 잘 수행되는지를 확인하고자 한다. 이후에 추가적인 공부를 거쳐 CSS나 JS를 활용하여 조금 더 아름다운 UI로 디자인하고자 한다. 이 부분은 앞으로 공부를 더 한다면 충분히 커버되는 부분이라고 생각한다. 실제로 현재 만들어진 메인페이지도 추가적인 공부를 통해 만든 화면이다.