

# 부동산 웹거래 플랫폼

팀 명 : 조니  
지도 교수 : 이병천 교수님  
팀 장 : 조니백  
팀 원 : 조니백

2024. 11.

중부대학교 정보보호학과

# 목 차

<b>1. 서 론</b>	
1.1 연구배경	4
1.2 연구 목적 및 필요성	4
<b>2. 관련연구</b>	
2.1 Node JS	5
2.2 NestJ	5
2.3 GraphQL	5
2.4 JavsScript	6
2.5 TypeScript	6
2.6 Websocket	6
2.6.1 MongoDB	7
2.6.2 NextJS	7
2.6.3 MUI	7
2.6.4 Hostinger	8
2.6.5 Docker	8
<b>3. 본 론</b>	
3.1 서비스 구상	9
3.2 프론트엔드 구성	10
3.3 백엔드 구성	21
<b>4. 서비스 안내</b>	
4.1 웹서비스 구성	24
<b>5. 결 론</b>	
5.1 결론 및 기대효과	25
<b>6. 별 첨</b>	
6.1 팀원 소개	26

# 1. 서론

## 1.1 연구 배경

현재 부동산 시장은 빠르게 디지털화되고 있으며, 많은 국가에서 온라인 플랫폼을 통해 부동산 거래가 이루어지고 있습니다. 특히 한국에서는 **직방**이나 **다방**과 같은 부동산 플랫폼이 활성화되어, 사용자들이 집을 찾는 방식이 기존의 오프라인 방식에서 온라인으로 전환되고 있습니다. 그러나 이러한 기술적 발전이 모든 국가에서 동일하게 이루어지지 않는 것입니다.

특히, 중양아시아에서는 부동산 거래의 디지털화가 아직 초기 단계에 있으며, 대부분의 거래가 전통적인 방식으로 이루어집니다. 이런 상황에서 사용자들이 **시간**과 **비용**을 절감하면서도 효율적으로 부동산 정보를 얻고 거래를 진행할 수 있는 디지털 플랫폼에 대한 수요가 높아지고 있습니다.

## 1.2 연구 목적 및 필요성

본 연구의 목적은 중양아시아 시장에 적합한 디지털 거래 플랫폼을 개발하는 것입니다. 이를 통해 중개사무소와 일반 사용자 간의 상호작용을 원활하게 하고, 집을 구하거나 광고할 때 발생하는 **시간적, 비용적 비효율성**을 해결하고자 합니다.

구체적으로 본 연구는 다음의 목적을 달성하고자 합니다:

- **사용자 친화적인 부동산 검색 기능 개발:** 필터링을 통해 자신에게 맞는 집을 빠르고 쉽게 찾을 수 있는 시스템 구현
- **중개사무소와의 상호작용 강화:** 중개사무소가 매물을 쉽게 등록, 수정, 관리할 수 있는 기능 제공
- **사용자 경험 향상:** 커뮤니티 기능을 통해 사용자 간 소통을 활성화하고, 리뷰 시스템을 통해 매물 및 중개사에 대한 신뢰도를 높임
- **플랫폼의 효율성 검증:** 이 플랫폼을 통해 실제 사용자들이 겪는 시간적, 경제적 효율성을 분석하여 부동산 거래에서의 긍정적인 변화를 입증

중양아시아에서는 현재 부동산 거래 플랫폼의 부재로 인해, 사람들이 직접 현장을 방문하거나 지인들의 추천에 의존하는 비효율적인 방식으로 집을 구하고 있습니다. 이러한 방식은 시간과 비용 측면에서 많은 부담을 주며, 또한 거래 과정에서 투명성이 떨어질 수 있습니다.

따라서 **부동산 거래의 디지털 전환**이 필요한 상황에서, 사용자들이 손쉽게 매물을 검색하고, 중개사와 소통하며, 안전하게 거래할 수 있는 플랫폼을 제공하는 것이 필수

적입니다. 이러한 플랫폼은 **시간 절약**, **정보의 투명성 강화**, 그리고 **거래의 효율성**을 제공함으로써 사용자 경험을 크게 개선할 수 있습니다.

## 2. 관련 연구

### 2.1 Node JS

Node JS는 확장성 있는 네트워크 애플리케이션(특히 서버 사이드) 개발에 사용되는 소프트웨어 플랫폼으로, Chrome V8 JavaScript 엔진으로 빌드된 Javascript 런타임 플랫폼입니다. 작성 언어로 자바스크립트를 활용하며 논블로킹(Non-blocking) I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능을 가지고 있습니다. 내장 HTTP 서버 라이브러리를 포함하고 있어 웹 서버에서 아파치 등의 별도의 소프트웨어 없이 동작하는 것이 가능하며 이를 통해 웹 서버의 동작에 있어 더 많은 통제를 가능케 합니다. 블록 체인에 저장하는 거래 내역과 같은 데이터 이외의 채팅, 부가적인 사용자 정보 등을 Node JS의 프레임워크인 Express를 사용하여 만든 서버를 통해 저장하고 사용합니다.

### 2.2 Nest JS

NestJS는 Node.js 환경에서 서버 사이드 애플리케이션을 구축하기 위한 프레임워크로, TypeScript로 작성되었습니다. 이 프레임워크는 Angular와 유사한 아키텍처를 따르며, 모듈화된 구조를 제공하여 애플리케이션을 효율적으로 조직하고 관리할 수 있도록 돕습니다. NestJS는 의존성 주입 패턴을 지원하여 클래스 간의 의존성을 쉽게 관리할 수 있으며, HTTP 요청을 처리하는 미들웨어와 요청 데이터의 유효성을 검사하는 파이프를 쉽게 추가할 수 있습니다. TypeScript의 데코레이터를 사용하여 라우트와 의존성을 정의할 수 있어 코드가 간결해지고 가독성이 높아집니다. 또한, NestJS는 GraphQL과 REST API를 모두 지원하여 다양한 유형의 애플리케이션을 구축할 수 있는 유연성을 제공합니다. 이러한 특징들 덕분에 NestJS는 높은 생산성과 코드 품질을 추구하는 대규모 애플리케이션 개발에 적합한 프레임워크로 자리잡고 있습니다

### 2.3 GraphQL

GraphQL은 API를 위한 쿼리 언어이자 서버에서 클라이언트로 데이터를 효율적으로 전송하기 위한 런타임입니다. 페이스북에서 개발되었으며, REST API의 대안으로 자리 잡고 있습니다. GraphQL의 주요 특징은 클라이언트가 필요한 데이터의 구조를 정의할 수 있다는 점입니다. 즉, 클라이언트는 필요한 데이터의 형식을 정확하게 요청할 수 있으며, 서버는 그에 맞춰 데이터를 반환합니다. 이러한 방식 덕분에 GraphQL은 데이터 전송량을 줄이고, 클라이언트와 서버 간의 통신을 최적화할 수 있습니다. 예를 들어, RESTFULL API에서는 여러 엔드포인트를 호출해야 할 경우, GraphQL은 하나의 요청으로 필요한 모든 데이터를 가져올 수 있습니다. GraphQL의 주요 구성 요소는 다음과 같습니다. 첫째, **스키마**는 API의 데이터 구조를 정의하며, 각 필드와 그 타입을 명세합니다. 둘째, **쿼리**는 클라이언트가 서버에 요청하는 데이터의 구조를 정의하는 방식입니다. 셋째, **뮤테이션**은 데이터를 수정하는 요청을 나타냅니다.

마지막으로, **서브스크립션**은 실시간 데이터 업데이트를 처리하는 기능입니다.

GraphQL은 다양한 플랫폼에서 사용할 수 있으며, 프론트엔드와 백엔드 간의 데이터 통신을 보다 유연하고 효율적으로 만들기 위해 설계되었습니다.

## 2.4 JavaScript

JavaScript는 웹 개발에서 가장 널리 사용되는 프로그래밍 언어 중 하나로, 클라이언트 측 및 서버 측 애플리케이션을 작성하는 데 사용됩니다. 원래 브라우저에서 동적인 웹 페이지를 만들기 위해 개발되었지만, 현재는 Node.js와 같은 환경 덕분에 서버 측 개발에서도 많이 활용됩니다. JavaScript는 객체 기반의 언어로, 함수형 프로그래밍을 지원하며, 비동기 프로그래밍을 쉽게 할 수 있는 기능(예: 콜백, 프로미스, async/await 등)을 제공합니다. 이 덕분에 사용자와의 상호작용에 반응하는 웹 애플리케이션을 만들 수 있으며, 실시간 데이터 업데이트와 같은 기능을 구현할 수 있습니다. 또한, JavaScript는 다양한 라이브러리와 프레임워크(예: React, Angular, Vue.js 등)와 함께 사용되어 웹 애플리케이션의 개발을 더욱 간편하고 효율적으로 만들어 줍니다. 이러한 생태계 덕분에 JavaScript는 프론트엔드와 백엔드 모두에서 광범위하게 사용되며, 전 세계 개발자들 사이에서 인기가 높습니다. JavaScript의 특징 중 하나는 동적 타이핑(dynamic typing)으로, 변수의 타입을 명시적으로 선언할 필요가 없고, 실행 시점에 변수의 타입이 결정됩니다. 그러나 이로 인해 타입 관련 오류가 발생할 수 있기 때문에, TypeScript와 같은 정적 타입 시스템을 사용하는 방법도 인기를 끌고 있습니다.

## 2.5 TypeScript

TypeScript는 JavaScript의 상위 집합으로, 마이크로소프트에서 개발한 프로그래밍 언어입니다. TypeScript는 JavaScript의 모든 기능을 포함하면서, 정적 타입 시스템과 최신 ECMAScript 기능을 추가하여 개발자가 더 안전하고 효율적인 코드를 작성할 수 있도록 도와 줍니다. TypeScript의 주요 특징 중 하나는 정적 타입을 지원한다는 점입니다. 즉, 변수, 함수, 객체 등의 데이터 타입을 명시적으로 선언할 수 있어, 컴파일 시점에서 타입 관련 오류를 사전에 발견할 수 있습니다. 이러한 기능은 코드의 가독성과 유지보수성을 높이며, 대규모 애플리케이션 개발 시에 유용합니다. 또한, TypeScript는 최신 JavaScript 기능(예: 비동기 함수, 클래스, 모듈 등)을 지원하며, 이들 기능을 사용하여 더 구조적이고 모듈화된 코드를 작성할 수 있게 합니다. TypeScript로 작성된 코드는 컴파일 과정을 통해 순수 JavaScript 코드로 변환되므로, 기존의 JavaScript 실행 환경에서 문제없이 실행할 수 있습니다. TypeScript는 다양한 프레임워크와 라이브러리와 호환되며, 특히 Angular와 같은 프론트엔드 프레임워크에서 널리 사용됩니다. 많은 개발자들이 TypeScript를 사용하여 코드의 안정성과 생산성을 높이고 있으며, 강력한 도구와 지원을 통해 JavaScript 개발의 품질을 향상시키고 있습니다.

## 2.6 WebSocket

WebSocket은 클라이언트와 서버 간의 양방향 통신을 가능하게 하는 프로토콜로, 실시간 데이터 전송이 필요한 애플리케이션에서 매우 유용하게 사용됩니다. 전통적인 HTTP 요청-응답 모델에서는 클라이언트가 요청을 보내면 서버가 응답을 반환하는 단방향 통신 방식이지만,

WebSocket은 연결이 수립된 후 클라이언트와 서버가 자유롭게 메시지를 주고받을 수 있도록 설계되었습니다. 이로 인해 WebSocket은 지속적인 연결을 유지하면서 반복적으로 연결을 열고 닫는 오버헤드를 줄여줍니다. 또한, 클라이언트와 서버는 서로 독립적으로 메시지를 전송할 수 있어, 서버가 클라이언트의 요청 없이도 데이터를 푸시할 수 있습니다. 이는 채팅 앱, 실시간 게임, 주식 거래 플랫폼 등에서 높은 성능과 빠른 응답 속도를 제공하는 데 큰 장점을 가지고 있습니다. WebSocket은 초기 핸드셰이크 후에 경량 프로토콜로 전환되며, 데이터 전송 시 헤더가 작고 효율적이기 때문에 네트워크 대역폭을 절약할 수 있습니다. 이러한 특성 덕분에 WebSocket은 실시간 데이터 전송이 필요한 다양한 웹 애플리케이션에서 중요한 기술로 자리 잡고 있습니다.

## 2.7 MongoDB

MongoDB는 문서 지향 데이터베이스 관리 시스템으로, NoSQL 데이터베이스의 일종입니다. 주로 JSON과 유사한 BSON(Binary JSON) 형식으로 데이터를 저장하며, 비정형 데이터와 복잡한 데이터 구조를 유연하게 처리할 수 있습니다. MongoDB는 대규모 데이터를 처리하고, 빠른 읽기 및 쓰기 성능을 제공하는 데 최적화되어 있으며, 수평적 확장이 용이하다는 특징이 있습니다.

## 2.8 Next JS

Next.js는 React 기반의 프레임워크로, 서버 사이드 렌더링(SSR)과 정적 사이트 생성(SSG)을 지원하여 웹 애플리케이션을 개발하는 데 매우 유용한 도구입니다. Vercel에서 개발 및 유지 관리하며, React의 장점을 활용하면서도 성능과 SEO(검색 엔진 최적화)에 중점을 둡니다. Next.js의 주요 특징 중 하나는 페이지 기반 라우팅입니다. 파일 시스템 기반 라우팅을 통해 각 페이지가 파일로 구성되어 있어, 새로운 페이지를 추가할 때 파일을 생성하는 것만으로도 자동으로 라우트가 생성됩니다. 이는 개발자가 라우팅을 설정하는 데 드는 시간을 절약할 수 있게 해줍니다. 또한, Next.js는 서버 사이드 렌더링을 통해 페이지를 서버에서 미리 렌더링하여 클라이언트에 전송할 수 있습니다. 이를 통해 초기 로딩 시간을 단축하고, 검색 엔진 크롤러가 페이지 내용을 쉽게 인식할 수 있도록 도와줍니다. 정적 사이트 생성 기능을 통해 빌드 시점에 페이지를 미리 생성하여, 더 빠른 성능을 제공할 수 있습니다. Next.js는 API 라우트를 지원하여, 서버리스 기능을 활용한 API를 쉽게 구축할 수 있습니다. 이 덕분에 프론트엔드와 백엔드의 경계를 허물고, 동일한 코드베이스에서 API를 관리할 수 있는 장점을 제공합니다. 또한, CSS 모듈, Sass, Styled Components 등 다양한 스타일링 옵션을 지원하여 개발자가 선호하는 방식으로 스타일링할 수 있도록 유연성을 제공합니다.

## 2.9 MUI(Material-UI)

MUI(Material-UI)는 React 애플리케이션을 위한 인기 있는 UI 라이브러리로, 구글의 Material Design 원칙을 기반으로 설계되었습니다. 이 라이브러리는 개발자가 빠르고 쉽게 아름답고 반응형 웹 애플리케이션을 구축할 수 있도록 다양한 컴포넌트와 스타일링 도구를 제공

합니다. MUI의 주요 특징 중 하나는 재사용 가능한 컴포넌트를 제공한다는 점입니다. 버튼, 텍스트 필드, 카드, 다이얼로그 등 다양한 UI 요소를 미리 정의된 형태로 사용할 수 있어, 개발자는 UI를 빠르게 구성하고 일관성 있는 디자인을 유지할 수 있습니다. MUI는 TypeScript와의 호환성도 뛰어나, 타입 안전한 개발을 지원합니다. 또한, 다양한 문서와 예제가 잘 갖춰져 있어, 개발자들이 쉽게 라이브러리를 배우고 활용할 수 있습니다

## 2.10 Hostinger

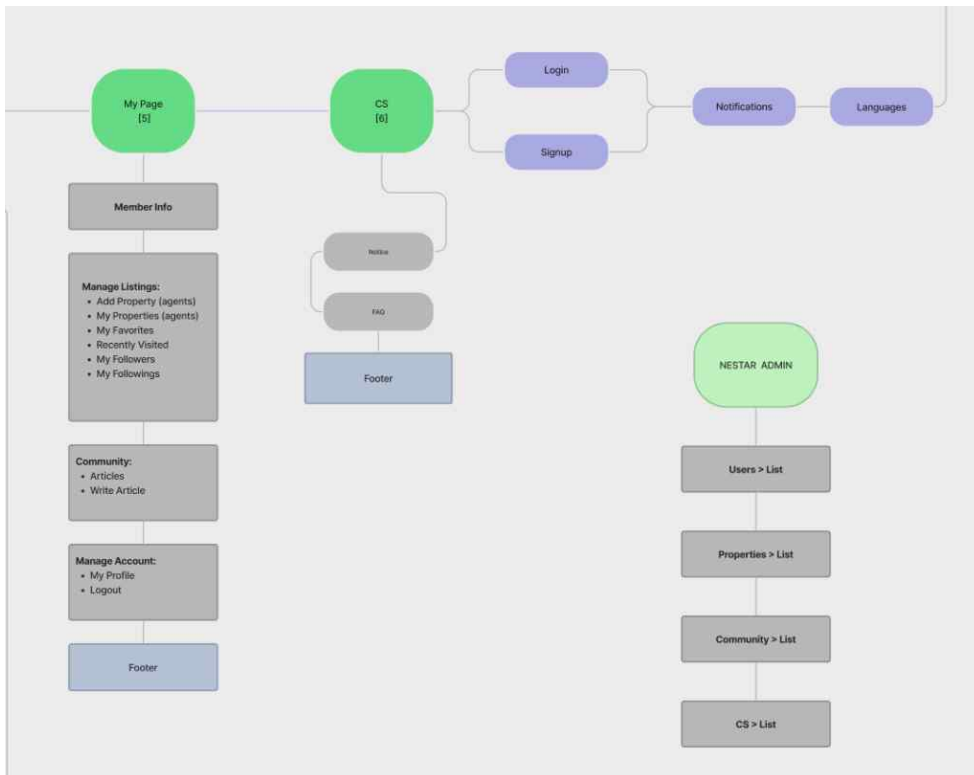
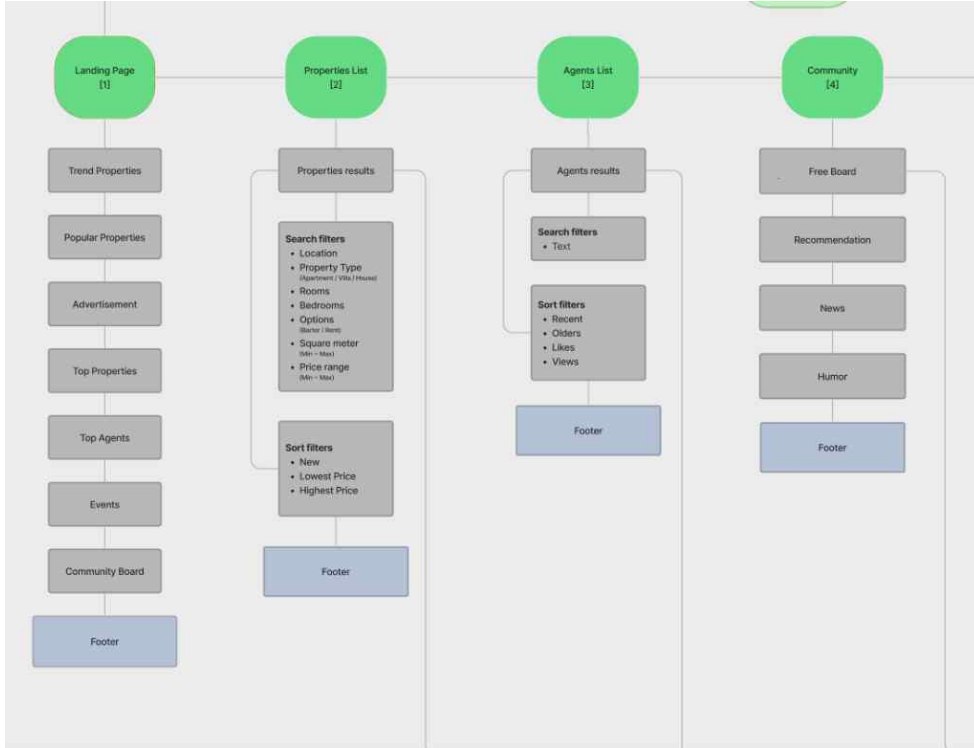
Hostinger는 웹 호스팅 서비스 제공업체로, 다양한 호스팅 솔루션을 제공합니다. 개인 웹사이트부터 대규모 기업용 웹 애플리케이션까지, 다양한 고객의 요구에 맞춰 공유 호스팅, VPS(가상 사설 서버), 클라우드 호스팅, 전용 서버 등 여러 호스팅 옵션을 제공합니다. Hostinger의 주요 특징 중 하나는 저렴한 가격과 함께 높은 성능을 제공한다는 점입니다. 신규 고객을 위해 다양한 할인 혜택을 제공하며, 기본 호스팅 패키지에서도 많은 기능을 포함하고 있습니다. 이러한 경제적인 접근 덕분에 개인 블로거와 중소기업에 인기가 높습니다. 또한, Hostinger는 사용자 친화적인 인터페이스를 제공하여 웹사이트를 쉽게 관리할 수 있도록 돕습니다. 자체 개발한 hPanel을 통해 도메인 관리, 데이터베이스 설정, 이메일 계정 생성 등을 직관적으로 수행할 수 있습니다.

## 2.11 Docker

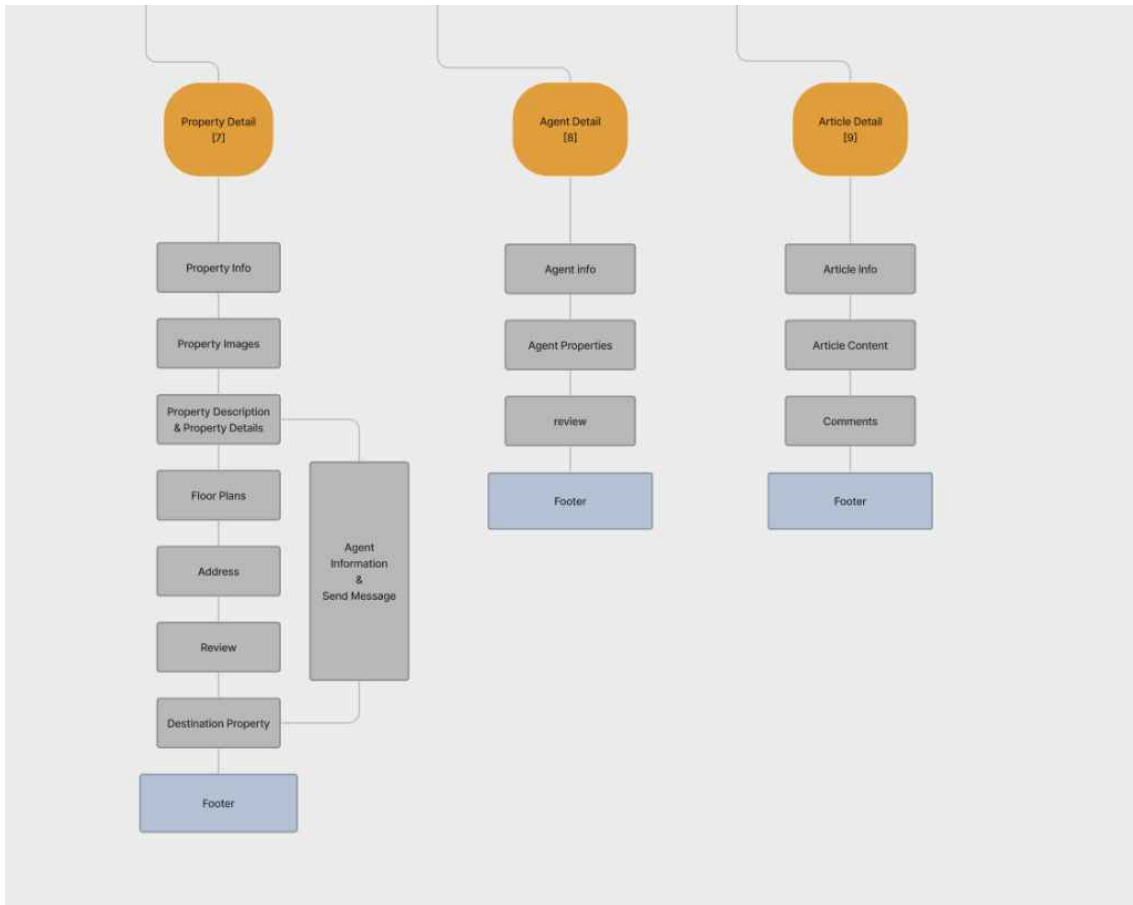
Docker는 애플리케이션을 컨테이너라는 가상화된 환경에서 실행할 수 있도록 해주는 플랫폼입니다. 컨테이너는 애플리케이션과 그에 필요한 모든 종속성을 함께 패키징하여, 어디서나 일관되게 실행될 수 있는 환경을 제공합니다. 이를 통해 개발자는 다양한 환경(예: 개발, 테스트, 운영)에서 발생할 수 있는 "내 컴퓨터에서는 잘 되는데" 문제를 최소화할 수 있습니다.

### 3. 본론

#### 3.1 서비스 구성

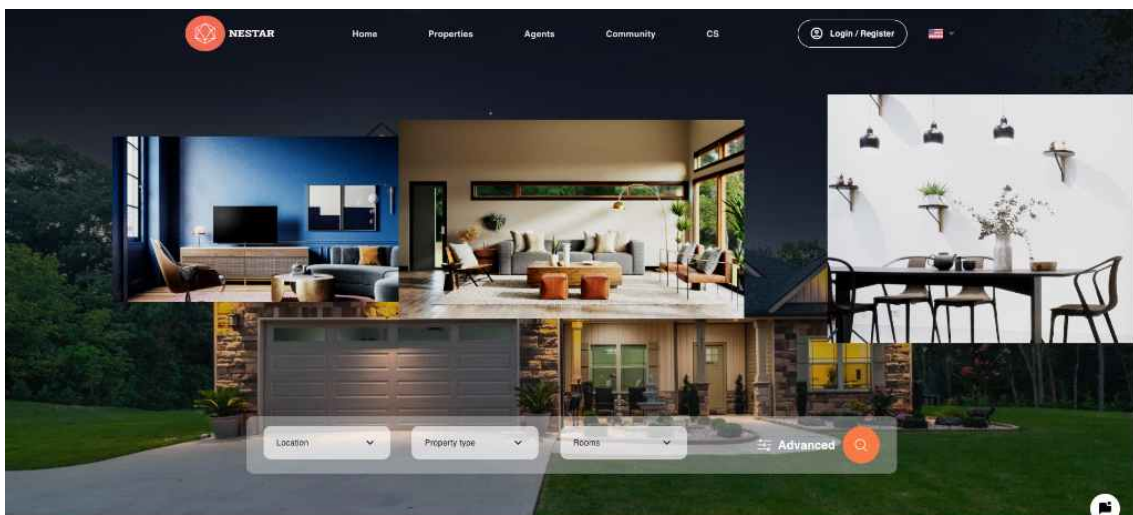






### 3.2 프론트엔드 구성

#### 1) 내비게이션바



Home: 홈페이지.

Properties: 매물 정보를 검색하고 볼 수 있는 페이지.

Agents: 중개사무소를 검색하고 볼 수 있는 페이지.

Community: 글을 검색하고 볼 수 있고, 글을 작성할 수 있는 페이지.

CS: 고객센터 페이지.

MyPage: 사용자 페이지.

Login/Register: 로그인/회원가입 페이지.

## 2) Home Page

### Trend Properties

Trend is based on likes



The Trend Properties section displays four property cards. Each card features a main image, a price tag, a title, a short description, and key details like bedrooms, rooms, and area. The cards are: 1. Kingston Prestige (\$250,000, 2 bed, 4 rooms, 200 m2, Rent/Barter, 9 likes, 11 hearts). 2. Heaven Villas (\$150,000, 2 bed, 4 rooms, 100 m2, Rent, 11 likes, 11 hearts). 3. Prime Apartment (\$350,000, 3 bed, 5 rooms, 200 m2, Rent/Barter, 17 likes, 9 hearts). 4. Best Villa (\$120,000, 1 bed, 3 rooms, 75 m2, Barter, 6 likes, 6 hearts).

Trend Properties는 하루 종일 제일 조회수가 많이 보이는 매물 정보입니다. 백엔드에는 매일 밤 00:10 분에 실행하는 batch 로직이 있습니다.

### Popular Properties

Popularity is based on views

[See All Categories](#)

The Popular Properties section displays three property cards. Each card features a main image, a price tag, a title, a short description, and key details like bedrooms, rooms, and area. The cards are: 1. Prime Apartment (\$350,000, 3 bed, 5 rooms, 200 m2, Rent, 17 views). 2. Heaven Villas (\$150,000, 2 bed, 4 rooms, 100 m2, Rent, 11 views). 3. Kingston Prestige (\$250,000, 2 bed, 4 rooms, 200 m2, Rent, 9 views).



## Top Agents

Our Top Agents always ready to serve you

[See All Agents](#)



**Jonibek**  
AGENT



**Oscar**  
AGENT

그리고 Popular Properties 정보가 하루 종일 제일 “좋아요” 수가 많은 매물 정보입니다. Top Agents는 우리 웹사이트에서 제일 훌륭하고, 매물을 많이 등록하고, 판매한 중개사무소 정보입니다.

### 3) Properties Page

#### Find Your Home

What are you looking for?

**Location**

SEOUL  
 BUSAN  
 INCHEON  
 DAEGU

**Property Type**

APARTMENT  
 VILLA  
 HOUSE

**Rooms**

Any 1 2 3 4 5+

**Bedrooms**


Any 1 2 3 4 5+

**Options**

Barter  
 Rent

**Square meter**


Min 0 Max 500



**Busan Villa**  
Busan, BUSAN

2 bed 3 room 100 m2


Rent Barter 4 hearts 2



**Seoul THE SHARP Villa**  
L020L022L620, SEOUL

2 bed 5 room 150 m2


Rent Barter 8 hearts 6



**Prime Apartment**  
Seoul Prime 20-48, SEOUL

3 bed 5 room 200 m2


Rent Barter 17 hearts 9



**Winner Apartments**  
Seoul City Center 50, SEOUL

3 bed 5 room 300 m2


Rent Barter 8 hearts 6



**Heaven Villas**  
Busan Trading Center, BUSAN

2 bed 4 room 100 m2

Rent Barter 11 hearts 11



**Best Villa**  
Busan South Gate 20-58, BUSAN

1 bed 3 room 75 m2

Rent Barter 6 hearts 6



Bedroom 3 9.9 View Room 5 📅 Year Built 2024 🏠 Size 200 m2 🏠 Property Type APARTMENT


**Property Description**

This is our Prime offer for your best life!

**Property Details**

Price	\$350,000	Year Built	2024
Property Size	200 m2	Property Type	APARTMENT
Rooms	5	Property Options	For Barter
Bedrooms	3		Rent

**Get More Information**

 **Jonibek**  
📞 01048672909  
[View Listings](#)

**Name**

**Phone**

**Email**

Properties 페이지에서는 필터링을 통해 자기에게 맞는 방을 손쉽게 찾을 수 있습니다. 궁금한 방이 어떤 방인지, 어디에 있는지 등 자세한 정보를 한꺼번에 볼 수 있습니다. 그리고 방에 대한 의견도 리뷰로부터 확인할 수도 있습니다. 만약에 자기에게 방이 맞다면 중개사무소로 연락을 할 수 있습니다. 그리고 그 방을 다른 사람들에게 추천하고 싶을 때나 나중에 불생각이 있으면 "좋아요" 버튼을 누를 수도 있습니다.

## 4) Agents Page

Search for an agent

Sort by Recent

2 properties

Oscar Agent 2 1

5 properties

Jonibek Agent 2 2

Total 2 agents available

### Reviews

we are glad to see you again

+ 1 review

Kim  
12 October

good person

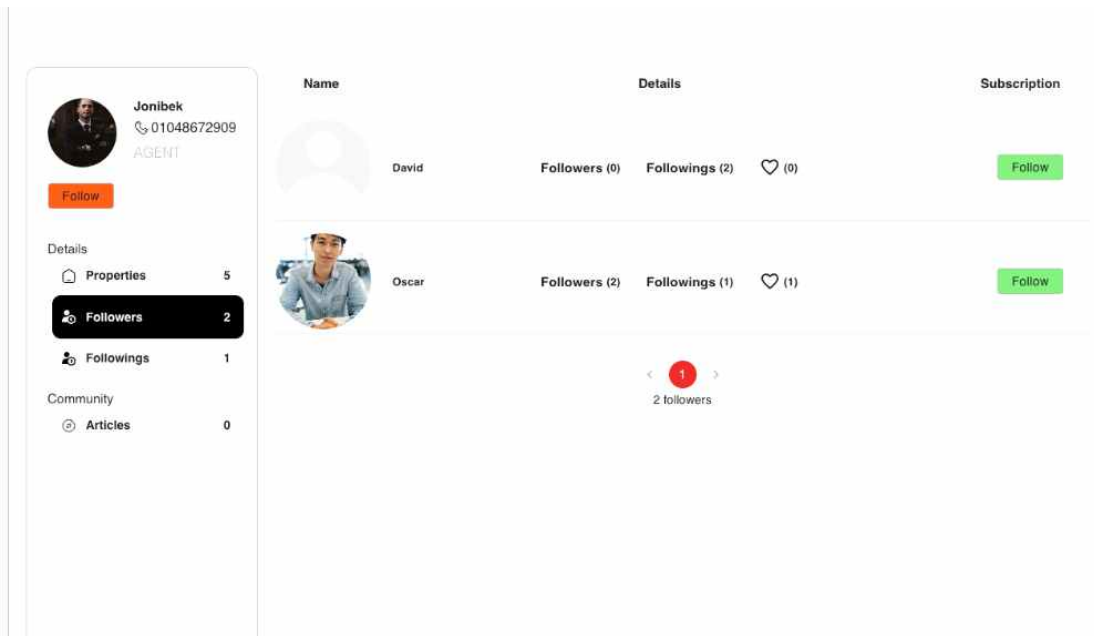
< 1 >

#### Leave A Review

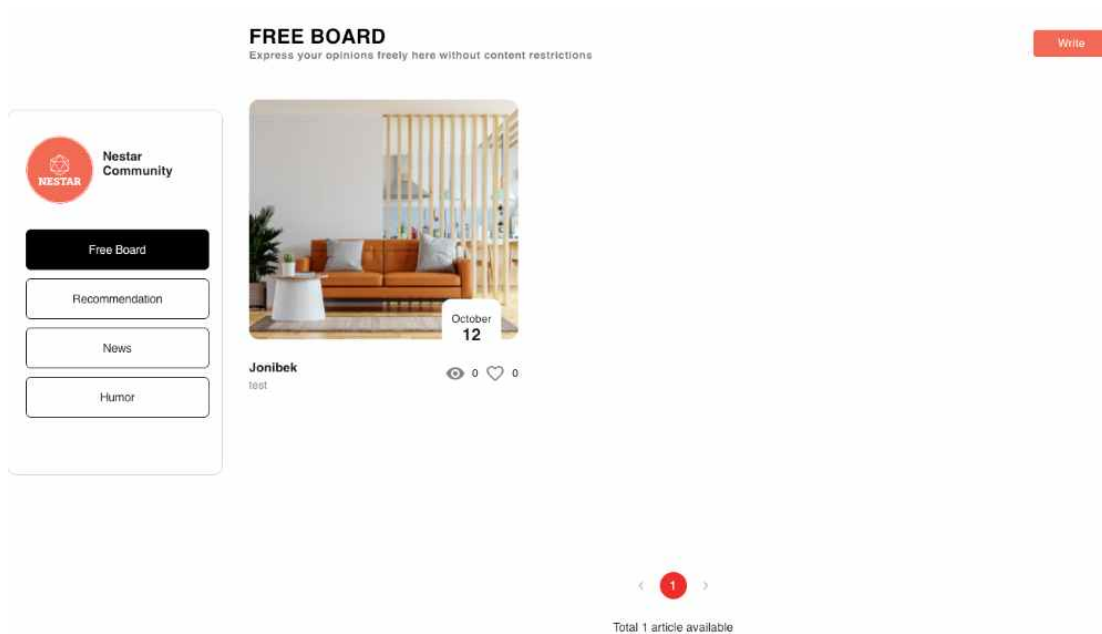
Review

Submit Review ↗

방만 아니라 중개사무소도 따로 찾을 수 있습니다. 궁금한 중개사무소의 방이나 정보를 볼 수도 있고, 다른 사람들이 그 중개사무소에 어떤 생각을 갖고 있는지도 리뷰를 통해 알 수 있을 것입니다. 만약에 중개사무소의 매물을 자주 보고 싶다면 follow 나 unfollow 기능을 사용할 수 있습니다. 좋아하는 중개사무소에 리뷰를 남길 수도 있고 “좋아요”를 누를 수도 있습니다.



## 5) Community Page




우리 사이트에서는 그뿐만 아니라 서로 다른 사람들하고 의사소통도 할 수 있습니다. 이런 기회를 바로 Community 페이지가 줍니다. Community 페이지에서는 글을 쓸 수도 있고, 다른 사람이 냈던 의견도 읽을 수 있습니다. 그리고 “좋아요” 할 수도 있고 코멘트를 통해 응원할 수도 있습니다.



## 5) My Page

### Add New Property

We are glad to see you again!



**Jonibek**  
01048672909  
AGENT

MANAGE LISTINGS

- Add Property
- My Properties
- My Favorites
- Recently Visited
- My Followers
- My Followings

Community

- Articles
- Write Article

MANAGE ACCOUNT

- My Profile
- Logout

**Title**

**Price**

**Select Type**

**Select Location**

**Address**

**Barter**

**Rent**

**Rooms**

**Bed**

**Square**


**Property Description**

Description

[ Add New Property ]

### My Properties

We are glad to see you again!



**Jonibek**  
01048672909  
AGENT

MANAGE LISTINGS

- Add Property
- My Properties
- My Favorites
- Recently Visited
- My Followers
- My Followings





Community

- Articles
- Write Article

MANAGE ACCOUNT


- My Profile
- Logout

On Sale    On Sold

Listing title	Date Published	Status	View	Action
 <p><b>Seoul THE SHARP Villa</b> \$122,222</p>	11 September, 2024	ACTIVE	8	<span>✎</span> <span>🗑️</span>
 <p><b>Prime Apartment</b> Seoul Prime 20-48 \$350,000</p>	04 June, 2024	ACTIVE	17	<span>✎</span> <span>🗑️</span>
 <p><b>Winner Apartments</b> Seoul City Center 50 \$550,000</p>	19 May, 2024	ACTIVE	8	<span>✎</span> <span>🗑️</span>
 <p><b>Heaven Villas</b> Busan Trading Center \$150,000</p>	19 May, 2024	ACTIVE	11	<span>✎</span> <span>🗑️</span>

[ My Properties & Edit Property ]

## Followers



**Jonibek**  
01048672909  
**AGENT**

MANAGE LISTINGS





- [Add Property](#)
- [My Properties](#)
- [My Favorites](#)
- [Recently Visited](#)
- [My Followers](#)
- [My Followings](#)

Community

- [Articles](#)
- [Write Article](#)

MANAGE ACCOUNT

- [My Profile](#)
- [Logout](#)

Name	Details	Subscription
 David	Followers (0)   Followings (2)  (0)	<a href="#" style="background-color: green; color: white; padding: 5px;">Follow</a>
 Oscar	Followers (2)   Followings (1)  (1)	<i>Following</i> <a href="#" style="background-color: red; color: white; padding: 5px;">Unfollow</a>


< 1 >

2 followers

[ Followers & Followings ]

## My Profile

We are glad to see you again!



**Jonibek**  
01048672909  
**AGENT**

MANAGE LISTINGS

- [Add Property](#)
- [My Properties](#)
- [My Favorites](#)
- [Recently Visited](#)
- [My Followers](#)
- [My Followings](#)


Community

- [Articles](#)
- [Write Article](#)

MANAGE ACCOUNT

- [My Profile](#)
- [Logout](#)

### Photo




Upload Profile Image

A photo must be in JPG, JPEG or PNG format!

Username

Phone

Address

Update Profile 



## [ My Profile ]

My Page에는 다양한 기능을 사용할 수 있습니다. 개인 정보 변경, 좋아하는 매물 정보, 팔로우/팔로잉, my article, 새로운 글을 작성할 수 있습니다. 만약에 중개사무소 사용자로 로그인하면 중개사무소 사용자가 방을 광고로 올릴 수 있습니다. 방을 올릴 때 필요한 정보를 자세하게 작성하고 올립니다. 그리고 올린 방을 수정할 수도 있고, 비공개나 공개로 설정할 수도 있습니다. 제일 중요한 것은 이익도 받을 수 있을 것입니다.

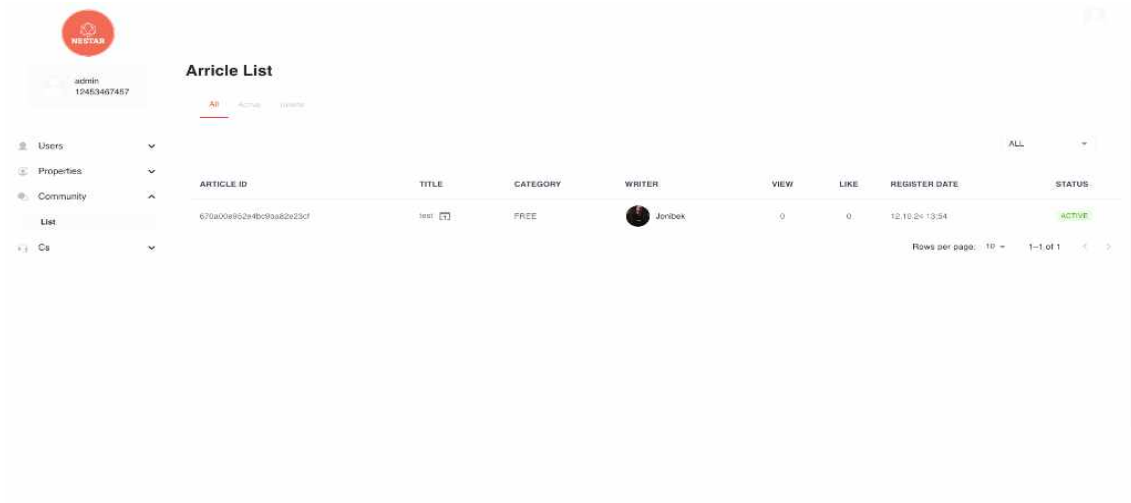
## 6) Admin Page

MB ID	NICK NAME	FULL NAME	PHONE NUM	MEMBER TYPE	WARNING	BLOCK CRIMES	STATE
610991132e40c9ca8262389	Nam	-	0102233485	USER	0	0	ACTIVE
66f041532e44c9ca8262115	Oscar	-	01012131213	AGENT	0	0	ACTIVE
66f0cca32e40c9ca82621a7	David	-	01098887766	USER	0	0	ACTIVE
66f0c0852e40c9ca8262199	Jimbok	-	0104922809	AGENT	0	0	ACTIVE
66f049252e40c9ca826216f	admin	-	12453467457	ADMIN	0	0	ACTIVE

## [Admin Members Control]

MB ID	TITLE	PRICE	AGENT	LOCATION	TYPE	STATUS
66e4494044cc11b112258	Busan Villa	500000	Oscar	BUSAN	VILLA	ACTIVE
66e59887944cc11b111156	Star	43	Oscar	BUSAN	VILLA	BLD
66e0a20244cc11b111178	Seoul THE SHARP Villa	122222	Jimbok	SEOUL	APARTMENT	ACTIVE
66e0e872044cc11b1111440	Globalcity	100000	Oscar	SEOUL	APARTMENT	BLD
66e0448244cc11b111114f	Chilchon Uy	100000	Jimbok	BUSAN	APARTMENT	BLD
66ecc7a22e40c9ca82621d5	Castle	260000	Jimbok	BUSAN	HOUSE	BLD
667d51b0c981226e18129	DALJON DCLAN LESION	300000	Oscar	DALJON	APARTMENT	BLD
665eac910aa0ff17c1bc93f	Prime Apartment	350000	Jimbok	SEOUL	APARTMENT	ACTIVE
66452ac0304025542117233a5	Winter Apartments	580000	Jimbok	SEOUL	APARTMENT	ACTIVE
6649286794025542117233a0	Grander House	80000	Oscar	DAEGU	HOUSE	BLD

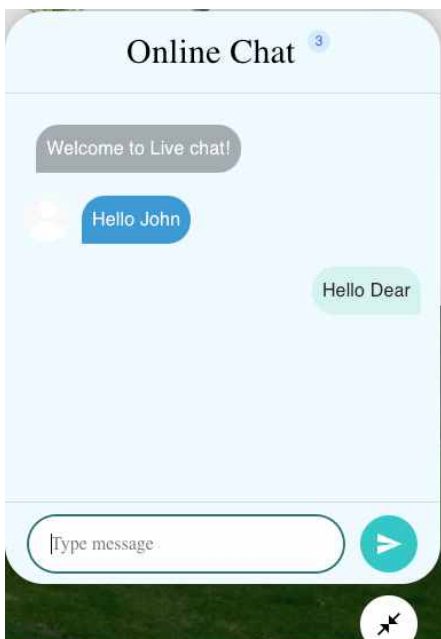
## [Admin Properties Control]



## [Admin Articles Control]

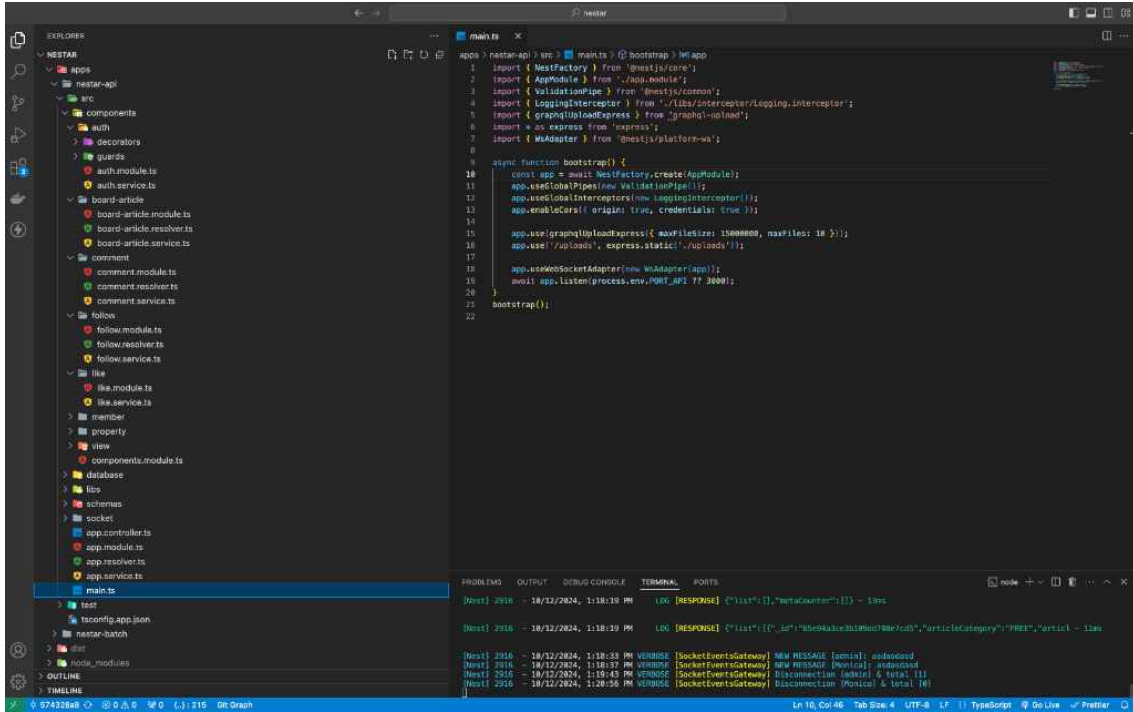
관리자 페이지를 살펴보면 관리자가 수많은 기능을 갖고 있습니다. 특히 사용자들이나 올린 매물을 관리할 수 있습니다. 만약에 어떤 사람이 사기를 했고, 그 사람이 사기를 했다는 신고를 받으면 바로 그 사용자를 Block 합니다. 그리고 어떤 사람이 일반 사용자부터 중개사무소 사용자로 활동하고 싶다고 요청하면 관리자가 일반사용자를 AGENT 사용자로 설정할 수도 있습니다.

## 7) Group Chatting

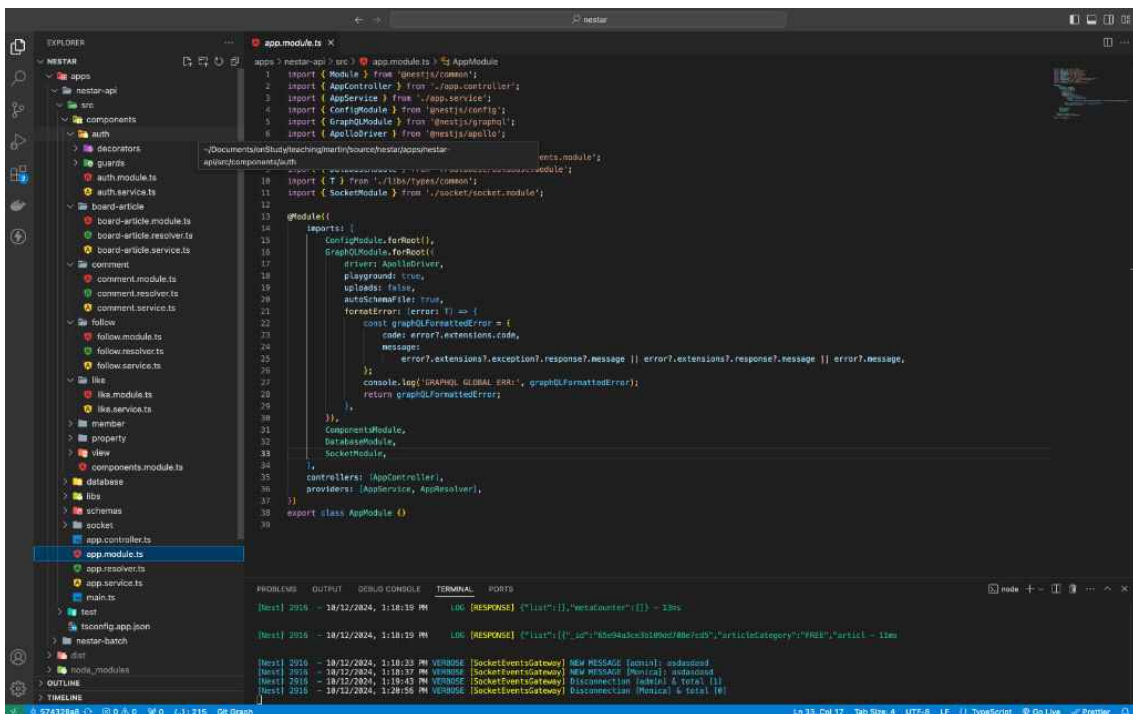


사이트를 이용하는 사용자들이 모두가 그룹 채팅을 할 수 있습니다. 서로 의사소통하고 즐거운 시간을 보내는 공간입니다.

### 3.3 백엔드 구성



[ 파일 구조 & main 파일 ]



[Global Integration]

```

@WebSocketGateway({ transports: ['websocket'], secure: false })
export class SocketGateway implements WebSocketGateway {
  private logger: Logger = new Logger('SocketEventsGateway');
  private summaryClient: Member = {};
  private clientsSetMap = new Map<Socket, Member>();
  private messagesList: MessagePayload[] = [];

  constructor(private authService: AuthService) {}

  @WebSocketServer()
  server: Server;

  public afterInit(server: Server) {
    this.logger.verbose('Websocket Server Initialized & total is {this.summaryClient}');
  }

  private async retrieveAuth(req: any): Promise<Member> {
    try {
      const parsedUrl = url.parse(req.url, true);
      const { token } = parsedUrl.query;
      return await this.authService.verifyToken(token as string);
    } catch (err) {
      return null;
    }
  }

  public async handleConnection(client: WebSocket, req: any) {
    const authMember = await this.retrieveAuth(req);
    this.summaryClient++;
    this.clientsSetMap.set(client, authMember);

    const clientKick: string = authMember?.memberNick ?? 'Guest';
    this.logger.verbose('Connection is {clientKick} & total is {this.summaryClient}');

    const infoMsg: InfoPayload = {
      events: 'Info',
      totalClients: this.summaryClient,
      memberData: authMember,
      action: 'joined',
    };
    this.emitMessage(infoMsg);
  }
}

```

```

[11:18:33 PM] [SocketEventsGateway] NEW MESSAGE [admin]: apasssads
[11:18:37 PM] [SocketEventsGateway] NEW MESSAGE [admin]: apasssads
[11:18:43 PM] [SocketEventsGateway] Disconnection [admin] & total is 1
[11:18:58 PM] [SocketEventsGateway] Disconnection [Merica] & total is 0

```

[ Socket GateWay ]

```

import { Injectable } from '@nestjs/common';
import * as bcrypt from 'bcrypt';
import { Member } from '.../libs/ota/member/member';
import { T } from '.../libs/types/common';
import { JwtService } from '@nestjs/jwt';
import { ShapeInfoMongoObjectID } from '.../libs/config';

@Injectable()
export class AuthService {
  constructor(private jwtService: JwtService) {}

  public async hashPassword(memberPassword: string): Promise<string> {
    const salt = await bcrypt.genSalt();
    return await bcrypt.hash(memberPassword, salt);
  }

  public async comparePasswords(passwords: string, hashedPassword: string): Promise<boolean> {
    return await bcrypt.compare(passwords, hashedPassword);
  }

  public async createToken(member: Member): Promise<string> {
    const payload: T = {
      object: { member: { _doc: member }, member },
      member: { _doc: member },
    };
    delete payload.memberPassword;
    return await this.jwtService.signAsync(payload);
  }

  public async verifyToken(token: string): Promise<Member> {
    const member = await this.jwtService.verifyAsync(token);
    member._id = ShapeInfoMongoObjectID(member._id);
    return member;
  }
}

```

```

[11:18:33 PM] [SocketEventsGateway] NEW MESSAGE [admin]: apasssads
[11:18:37 PM] [SocketEventsGateway] NEW MESSAGE [Merica]: indosands
[11:18:43 PM] [SocketEventsGateway] Disconnection [admin] & total is 1
[11:18:58 PM] [SocketEventsGateway] Disconnection [Merica] & total is 0

```

[ Authentication Model ]

```

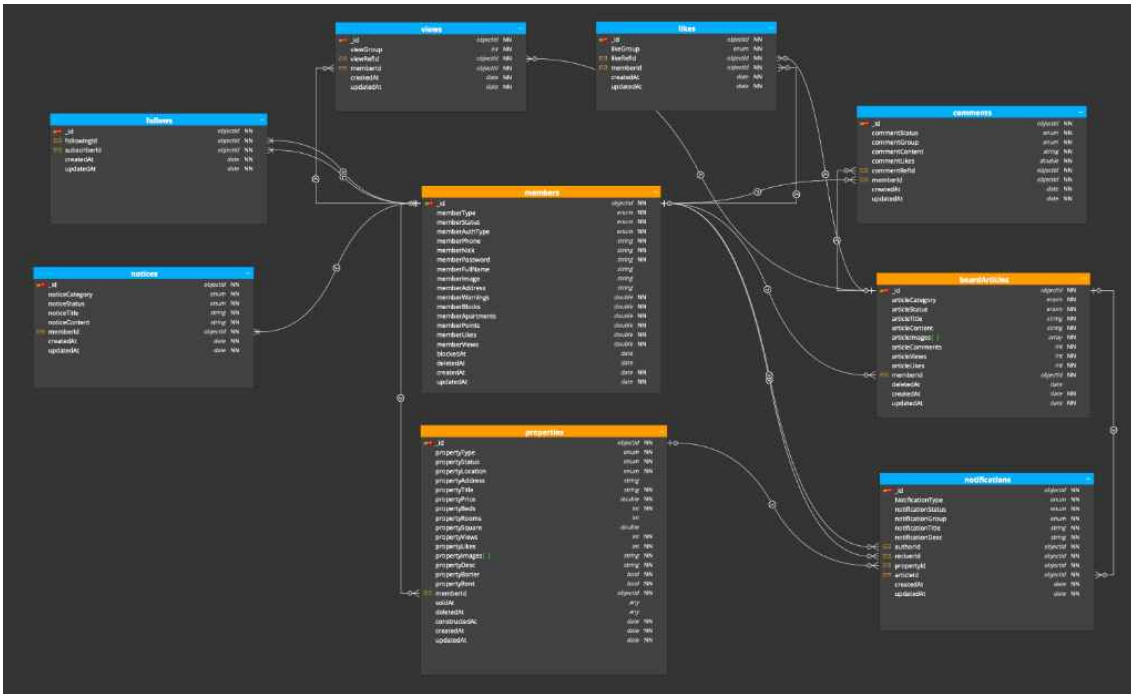
export class PropertyService {
  public async getProperties(memberId: ObjectId, input: PropertiesInquiry): Promise<Properties> {
    const result = await this.propertyModel
      .exec();
    if (!result.length) throw new InternalServerErrorException(Message_ML_DATA_EMPTY);
    return result[0];
  }

  private shapeMatchQuery(match: T, input: PropertiesInquiry): void {
    const {
      memberId,
      locationList,
      roomList,
      bedList,
      typeList,
      periodRange,
      priceRange,
      squareRange,
      options,
      text,
    } = input.search;
    if (memberId) match.memberId = shapeInfoRangeOfId(memberId);
    if (locationList && locationList.length) match.propertyLocation = ($in: locationList);
    if (roomList && roomList.length) match.propertyRooms = ($in: roomList);
    if (bedList && bedList.length) match.propertyBeds = ($in: bedList);
    if (typeList && typeList.length) match.propertyType = ($in: typeList);
    if (priceRange) match.propertyPrice = ($gte: priceRange.start, $lte: priceRange.end);
    if (periodRange) match.createdat = ($gte: periodRange.start, $lte: periodRange.end);
    if (squareRange) match.propertySquare = ($gte: squareRange.start, $lte: squareRange.end);
    if (text) match.propertyTitle = ($regex: new RegExp(text, ''));
    if (options) {
      match.$or = [options.map((ele) => {
        return { $ele: true; }
      })];
    }
  }

  public async getFavorites(memberId: ObjectId, input: OrdinaryInquiry): Promise<Properties> {
    return await this.likeService.getFavoriteProperties(memberId, input);
  }
}

```

[ Property Service & Model ]



[ ER Modeling ]

## 4. 서비스 안내

### 4.1 웹서비스 구성

- 웹서비스 주소: <http://62.72.1.166:6004/>

## 5. 결론

### 5.1 결론

이번 연구를 통해 우리는 **중양아시아 부동산 시장**에 적합한 디지털 플랫폼을 설계하고 개발했습니다. 이 플랫폼은 사용자의 **시간 절약**과 **효율성**을 목표로 하여, 부동산 검색 및 거래 과정을 보다 간편하게 만들었습니다. 특히, 중개사무소와 일반 사용자가 상호작용을 할 수 있는 기능을 중심으로, 매물 등록, 수정, 필터링 검색, 리뷰 및 커뮤니티 기능 등 다양한 기능을 구현했습니다.

본 연구의 핵심은 중양아시아 시장의 특수성을 고려한 사용자 경험을 제공하는 것입니다. 한국과 같은 선진국에서 이미 성공적으로 자리 잡은 부동산 플랫폼의 장점들을 활용하면서도, 현지의 필요와 요구 사항에 맞춘 맞춤형 솔루션을 제공했습니다. 이를 통해 중양아시아에서 부동산 거래의 디지털 전환을 촉진하고, 이 지역 시장에서 선도적인 플랫폼으로 자리 잡을 수 있는 가능성을 제시했습니다.

### 5.2 기대효과

사용자는 매물을 쉽게 검색할 수 있어, 오프라인 탐색에 소요되는 시간을 줄일 수 있습니다. 중개사무소는 매물 관리의 효율성을 높여 광고 비용을 절감할 수 있습니다.

리뷰와 평가 시스템을 통해 신뢰할 수 있는 매물 선택이 가능해지며, 중개사무소의 서비스 품질 향상에도 기여합니다. 직관적이고 사용자 친화적인 인터페이스를 통해 매물 검색과 거래가 간편해지고, 커뮤니티 기능을 통해 사용자 간 소통이 활성화됩니다.

또한 플랫폼을 통한 거래 증가로 중양아시아 부동산 시장이 활성화되며, 중소 중개사무소에 새로운 기회를 제공할 것입니다.

## 6. 별첨

### 6.1 팀원 소개 & 소스 코드

- 조니백(팀장)  
역할: 디자인, 백엔드, 프론트엔드, 배포, 발표  
개인 깃허브 주소: <https://github.com/jonik2909>
  
- 프로젝트 영상: <https://www.youtube.com/watch?v=h9v8BRwc3sE>