

# Project 3

2022년 2학기	
제출일	2022. 12. 11.
과목명	데이터베이스및데이터시각화
담당교수	이기훈

학과	컴퓨터정보공학부	학과	컴퓨터정보공학부
학번	2017202006	학번	2017202061
이름	박준용	이름	전승권
학과	컴퓨터정보공학부	학과	컴퓨터정보공학부
학번	2017202076	학번	2019202020
이름	이민석	이름	이세연

## KWANGWOON UNIVERSITY

### 1. 웹 애플리케이션에 대한 설명

국내 주요 증시 및 증권과 관련된 다양한 투자정보를 제공하는 서비스 웹 애플리케이션이다. 국내지수, 종목에 대한 상세 지표와 투자정보를 제공하며, 증권/금융과 관련된 당일 뉴스를 제공한다. 또한 사용자간의 종목에 대한 다양한 의견을 나눌 수 있는 소통의 장을 제공하며, MY 페이지를 통해 관심종목과 보유종목을 관리하여 자산현황을 한눈에 확인할 수 있다.

## 2. 주요 기능 설명

- 사용자
  - 로그인 및 회원가입

stockmarket 검색어를 입력하세요 로그인/회원가입

홈 공지사항 MY

### 사용자 로그인

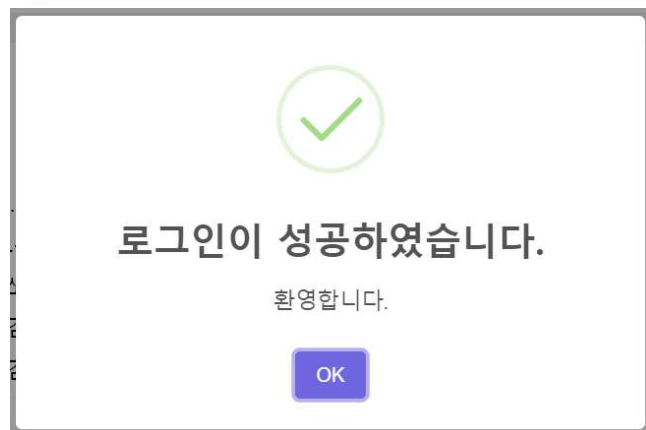
999999

\*\*\*\*\*

로그인

회원가입

관리자 이신가요?



웹에 접속하면 홈 네비게이션바 우측 상단에 로그인/회원가입 버튼이 존재한다. 해당 버튼을 클릭하면 로그인 창이 나타난다. 사용자 로그인이 기본으로 되어 있으며 하단에 회원가입 버튼과 관리자 로그인 전환 버튼이 존재한다. 사용자는 아이디와 비밀번호를 입력하여 로그인을 진행한다. 관리자는 ‘관리자이신가요?’ 버튼을 눌러 관리자 로그인 페이지로 이동해 로그인을 진행할 수 있다.

회원가입

아이디

5~15 영문 및 숫자를 포함

아이디 중복확인

비밀번호

문자,숫자를 포함 8자리 이상 입력

휴대폰번호

숫자만 입력 ex) 01012345678

이메일

id@site.com

이름

이름을 입력해주세요

나이

숫자로만 입력 ex) 23

이미 계정이 있습니다. [로그인 페이지](#)로 이동.

회원가입



해당ID는 사용 가능 합니다.

계속 회원가입을 진행해 주세요.

OK



회원가입에 실패했습니다.

비밀번호는 8~20자 최소 하나의 문자 및 하나의 숫자를 포함해야 합니다.

OK



회원가입에 성공하였습니다.

로그인을 진행해 주세요.

OK

사용자 로그인 페이지에서 회원가입 버튼을 클릭하면 사용자 회원가입 페이지로 이동한다. 아이디는 중복확인을 통해 사용 가능 여부를 알 수 있다. 아이디, 비밀번호, 휴대폰번호, 이메일, 이름, 나이를 모두 입력하여 회원가입 하도록 한다. 단, 각 조건을 만족하지 못하면 회원가입 버튼을 눌렀을 때 회원가입 실패와 그 이유를 alert 창으로 띄워준다. 사용자가 실수로 회원가입 버튼을 누르는 경우를 대비해 다시 로그인 페이지로 이동할 수 있도록 텍스트 버튼을 두었다.

#### - 개인정보 조회 및 수정

계좌

111122223333

▼

이세연

ggggg@naver.com

01077777777

수정

자산현황

817,500(-80.54 %)

총매입금액 : 4,200,000

당일실현손익 : 0.93%

총평가금액 : 817,500

내가 만든 토론

회원정보수정

개인정보 수정

휴대폰 번호

01077777777

이메일

ggggg@naver.com

나이

25

변경

비밀번호 수정

현재 비밀번호

새로운 비밀번호

비밀번호 확인

변경

로그인을 하면 홈 네비게이션 바에 있는 MY 페이지로 접속이 가능해진다. MY 페이지로 들어 오면 우측 상단에 내 이름과 이메일, 휴대폰번호의 정보가 뜨고 옆에 수정버튼을 누르면 본인의 정보를 조회 및 수정할 수 있다. 사용자가 원하는 파트만 수정할 수 있는 편의성을 고려하여 개인정보 수정과 비밀번호 수정을 구분해두었다.

## 회원정보수정

### 개인정보 수정

휴대폰 번호

01077777777

이메일

seyeon@naver.com

나이

변경사항이 적용되었습니다.

success

OK

변경

### 비밀번호 수정

현재 비밀번호

새로운 비밀번호

비밀번호 확인

변경

이세연

seyeon@naver.com

01077777777

수정

마이페이지에서 정보를 수정한 뒤 MY 페이지로 가보면 내 정보가 변경되어 있는 것을 확인할 수 있다.

- 보유주식 조회

계좌

111122223333  
111122223333  
123567890123

자산현황

817,500( -80.54 %)

총매입금액 : 4,200,000

당일실현손익 : 0.93%

총평가금액 : 817,500

보유주식

종목명	현재가격	구매가격	평가손익	전일대비	전일대비비율
현대차	163,500	840,000	-80.54%	-1,500	0.93%

관심종목

종목명	현재가격	등락률	거래량
SK하이닉스	78,200	-0.76%	3,769,573

이세연

seyeon@naver.com

01077777777

수정

내가 만든 토론

내가 남긴 댓글

MY 페이지에서 연결되어있는 자신의 계좌와, 자산현황, 보유하고 있는 주식, 관심종목, 내가 쓴 토론 게시판과 남긴 댓글 등의 정보를 확인할 수 있다.

- 토론게시판



## 기업소개

한국 및 DX부문 해외 9개 지역총괄과 DS부문 해외 5개 지역총괄, SDC, Harman 등 233개의 종속기업으로 구성된 글로벌 전자기업임.

## 주요주주현황

주요주주	소유주식수(주)	지분율
삼성생명보험 외 15인	1,241,176,035	20.79%
국민연금공단	458,637,667	7.68%
BlackRock Fund Advisors 외 15인	300,391,061	5.03%

## 종목토론실 [더보기 >](#)

삼성전자 좋아 [3]

stockmarket

검색어를 입력하세요



이세연 님 환영합니다

로그아웃

홈

공지사항

MY

## 삼성전자 토론게시판

제목	작성자	조회수	작성일자
삼성전자 좋아	aaaaa	16	2022-12-10

글쓰기

각 종목 페이지마다 하단에 종목토론실 탭이 있다. 해당 종목에 대한 게시판 글과 댓글 수를 미리보기 형식으로 확인할 수 있고, 더보기 버튼을 클릭하면 종목 토론게시판으로 이동한다. 게시글의 제목, 작성자 아이디, 조회수, 작성일자를 띄워주도록 했다. 그리고 하단의 글쓰기 버튼을 클릭하면 게시판에 글을 업로드할 수 있다.

삼전 오른다 ㅎㅎ

댓글

언제까지 올까요?

등록

드가자

kkkkk

22-12-10 03:22

다 거짓말이에요

aaaaa

22-12-10 15:09

다 장기투자도 잘해놓고더 먹으려다 매도시기를 놓친다 하  
수들은만바닥에서 사고 맨 꼭대기에서 파려니안되는것이지

aaaaa

22-12-10 15:19

목록

댓글

댓글 내용을 입력해주세요.

등록

드가자

kkkkk

22-12-10 03:22

다 거짓말이에요

aaaaa

22-12-10 15:09

다 장기투자도 잘해놓고더 먹으려다 매도시기를 놓친다 하  
수들은만바닥에서 사고 맨 꼭대기에서 파려니안되는것이지

aaaaa

22-12-10 15:19

언제까지 올까요?

ggggg

22-12-10 17:56

게시글을 클릭하면 자세한 내용을 확인할 수 있고, 댓글도 달 수 있다. 각 댓글마다 내용과 아이디, 작성 날짜와 시간을 출력하도록 했고, 목록 버튼을 누르면 종목 게시판으로 이동한다.



## 삼성전자 알잖아

aaaaa | 2022-12-10 03:20:43

삼전 무적권 오른다 ㄷㄷ

댓글

댓글 내용을 입력해주세요.

등록

드가자

kkkkk

22-12-10 03:22

목록

삭제하기

수정하기

게시글이 내가 작성한 것이라면 목록버튼 옆에 삭제하기 와 수정하기 버튼이 뜬다.

## 토론게시판 수정

제목

삼성전자 좋아

내용

삼성전자 매수하는 것이 바로 애국자의 삶

취소

수정



게시판 수정에 성공하였습니다.

게시판 페이지로 이동합니다.

OK

## 삼성전자 좋아

aaaaa | 2022-12-10 18:11:28

삼성전자 매수하는 것이 바로 애국자의 삶

댓글

댓글 내용을 입력해주세요.

등록

수정한 내용이 게시글에 반영되는 것을 확인할 수 있다.

## 삼성전자 토폰게시판

제목	작성자	조회수	작성일자
삼성전자 믿음시다 !	mmmmm	62	2022-12-10
삼성전자 신제품 나온단데??	eeeeee	103	2022-12-10
거의 보험인듯	fffff	73	2022-12-10

글쓰기

게시글 삭제에 성공하였습니다.

success

OK

삭제하기 버튼을 클릭하면 게시글 삭제에 성공했다는 alert창과 함께 게시판에서 글이 사라지는 것을 확인할 수 있다.

- 검색기능

stockmarket

삼성

Q

이세연 님 환영합니다!

로그아웃

홈

공지사항

MY

'삼성'에 대한 검색결과 입니다.

종목명	현재가격	거래량	전일대비	전일대비비율	현재변동률
삼성전자	58,700	12,779,480	500	-0.84%	0.00509
삼성SDI	709,000	292,713	-16,000	2.31%	-0.02119
삼성물산	116,500	402,429	1,000	-0.85%	0.01293
삼성생명	71,200	870,927	1,600	-2.20%	0.03116
삼성바이오로직스	828,000	83,790	-1,000	0.12%	-0.00481

홈 네비게이션바 상단에 검색창이 존재한다. 검색창에 키워드를 입력하고 엔터키를 누르거나 검색 버튼을 클릭하면 키워드를 포함하고 있는 종목의 세부정보들을 모두 출력해준다.

- 관리자
  - 회원관리

stockmarket

검색어를 입력하세요

Q

관리자님 환영합니다!

로그아웃

홈

공지사항

종목추가

회원관리

회원 관리

ID	이름	나이	전화번호	이메일	정지여부
aaaaa	홍길동	21	01011111111	aaaaa@naver.com	해제
bbbbb	최부자	35	01022222222	bbbbb@naver.com	해제
ccccc	박주식	42	01033333333	ccccc@naver.com	해제
ddddd	박준용	26	01044444444	ddddd@naver.com	해제
eeeee	전승권	51	01055555555	eeeee@naver.com	해제
fffff	이민석	33	01066666666	fffff@naver.com	해제
ggggg	이세연	25	01077777777	ggggg@naver.com	해제
hhhhh	이기업	27	01088888888	hhhhh@naver.com	해제
iiiii	김무식	39	01099999999	iiiii@naver.com	해제
jjjjj	신짱구	58	01000000000	jjjjj@naver.com	해제
kkkkk	우영우	26	01012121212	kkkkk@naver.com	해제
lllll	이수만	32	01013131313	lllll@naver.com	해제
mmmmm	양현석	25	01014141414	mmmmm@naver.com	해제

ID	이름	나이	전화번호	이메일	정지여부
aaaaa	홍길동	21	01011111111	aaaaa@naver.com	정지
bbbbbb	최부자	35	01022222222	bbbbbb@naver.com	해제
ccccc	박주식	42	01033333333	ccccc@naver.com	해제
dddddd	박준용	26	01044444444	dddddd@naver.com	해제

홈 네비게이션 바에서 회원관리 탭을 클릭하면 전체 가입자들의 정보를 확인할 수 있다. 그리고 정지여부 버튼을 클릭해 사용자를 정지시키거나 해제할 수 있다.

#### - 종목추가

stockmarket

검색어를 입력하세요

Q

관리자님 환영합니다!

로그아웃

홈

공지사항

종목추가

종목코드

종목 코드를 입력해주세요.

중복확인

종목명

종목 이름을 입력해주세요.

상장주식수

숫자만 입력해주세요.

기업이름

기업 이름을 입력해주세요.

주요주주 및 주식수

주주명을 입력해주세요.

보유주식수를 입력해주세요.

+

기업소개

기업 개요를 입력해주세요.

취소

등록

관리자로 로그인하면 홈 네비게이션 바에 종목추가 탭이 생성된다. 해당 탭을 클릭하면 종목 추가 페이지로 이동한다.

종목코드

123456

중복확인

종목명

상장주식수

기업이름

주요주주 및 주식

+

기업소개



해당 종목은 추가 가능 합니다.

계속 진행해 주세요.

OK

중복확인 버튼을 통해 종목코드를 체크함으로써 이미 존재하는 종목은 추가되지 않도록 했다.

종목추가

종목코드

123456

중복확인

종목명

이세연주식

상장주식수

1000

기업이름

이세연기업

주요주주 및 주식수

이세연

300

+

이민석

100

-

신함은할

120

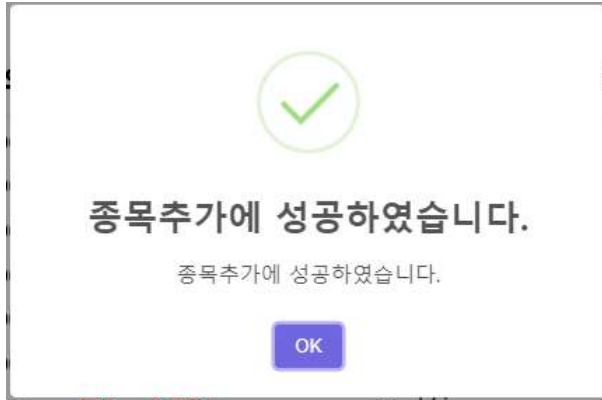
-

기업소개

이세연의 미래 보장을 위해 움직이는 기업입니다.

취소

등록

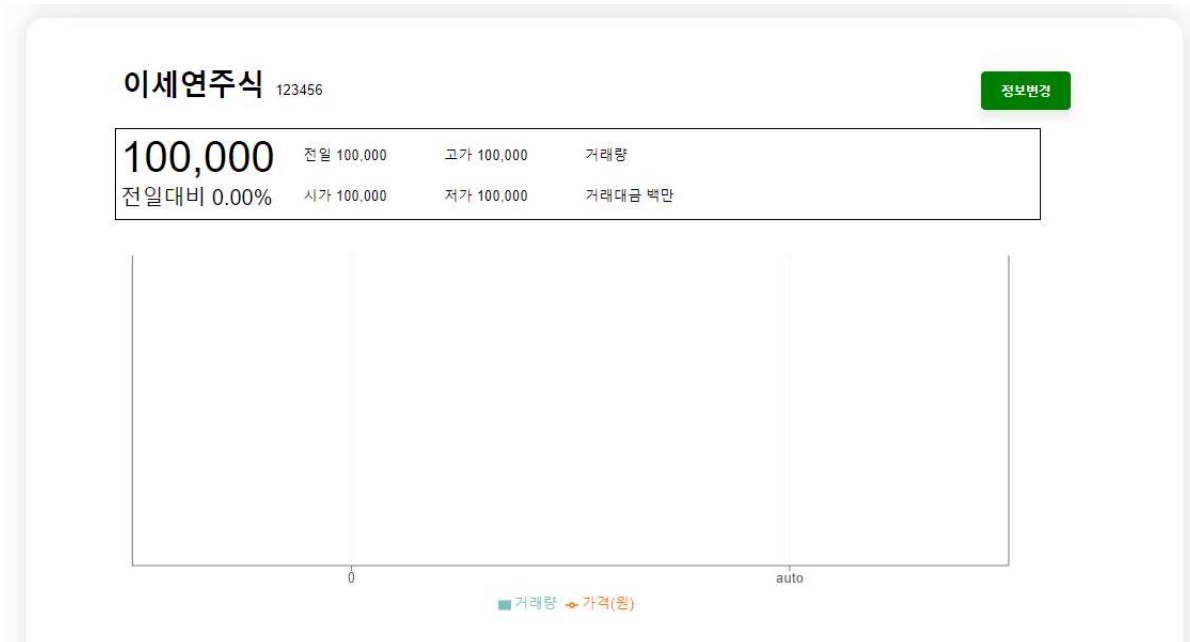


주요주주 및 주식수는 +, - 버튼을 통해 주주를 최대 5명까지 추가할 수 있다. 등록버튼을 누르면 추가가 완료된다. 가격은 default 값으로 10만원을 설정했고, 거래량과 변동량은 0으로 두었다.



검색창에 추가한 종목을 검색해보면 세부 데이터가 뜨는 것을 통해 추가 되었음을 확인할 수 있다.

#### - 종목정보 변경



종목수정

종목코드

123456

종목명

이세연주식

상장주식수

1000

기업이름

이세연기업

기업소개

이세연의 미래 보장을 위해 움직이는 기업입니다.

취소

등록

종목 페이지의 우측 상단에 정보변경 버튼이 존재한다. 해당 버튼을 클릭하면 종목 정보를 변경할 수 있다.

#### - 게시물 관리

stockmarket

검색어를 입력하세요

Q

관리자님 환영합니다!

로그아웃

홈

공지사항

종목추가

공지사항

번호	제목	조회수	작성일자
1	공지사항입니다.	140	2022-12-09
2	탈퇴 회원 명단	50	2022-12-10
3	신규 종목 출시 예정	100	2022-12-10
4	EVENT #1 출시!	328	2022-12-10
5	EVENT #1 당첨자 발표	548	2022-12-11

공지 등록

관리자는 공지사항 탭에서 글을 작성, 수정, 그리고 삭제할 수 있다.

### 3. 특별 기능 설명

#### - 관심 종목 조회

계좌 111122223333

### 자산현황

817,500(-80.54 %)

총매입금액 : 4,200,000      당일실현손익 : 0.93%      증평가금액 : 817,500

### 보유주식

종목명	현재가격	구매가격	평가손익	전일대비	전일대비비율
현대차	163,500	840,000	-80.54%	-1,500	0.93%

### 관심종목

종목명	현재가격	등락률	거래량
SK하이닉스	78,200	-0.76%	3,769,573

### 이세연

seyeon@naver.com

01077777777

수정

### 내가 만든 토론

### 내가 남긴 댓글

언제까지 올까요?

MY 페이지 하단의 관심종목 탭에서 내가 관심있는 종목을 조회할 수 있다.

### ☆ 삼성전자 005930

주문하기 >

58,700

전일대비 -500 -0.84%

전일 59,200

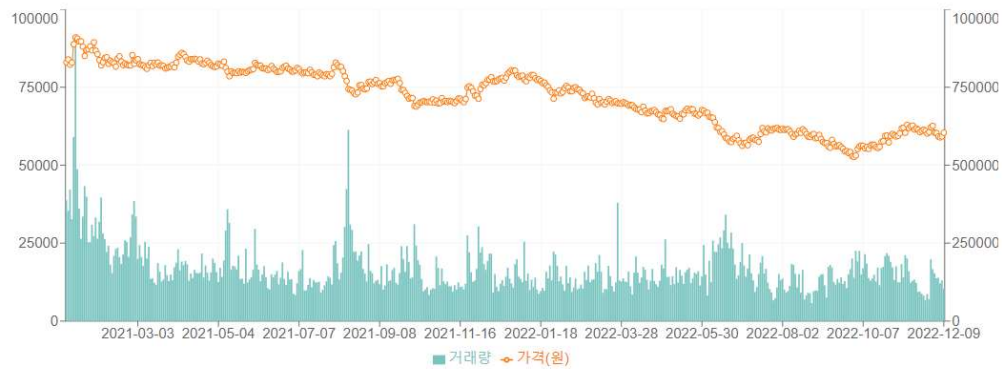
고가 59,200

거래량 12,779,480

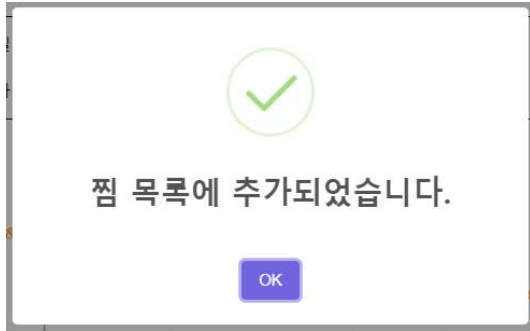
시가 58,700

저가 58,600

거래대금 750,155.476 백만







★ 삼성전자 005930

주문하기 >

58,700

전일 59,200

고가 59,200

거래량 12,779,480

전일대비 -500 -0.84%

시가 58,700

저가 58,600

거래대금 750,155.476 백만

## 관심종목

종목명	현재가격	등락률	거래량
SK하이닉스	78,200	-0.76%	3,769,573
삼성전자	58,700	-0.84%	12,779,480

종목 페이지에서 종목 이름 앞에 있는 ☆ 버튼을 누르면 내 관심 종목에 추가가 된다.

### - 최근 본 종목 조회



홈 화면 우측 상단에는 최근 본 종목을 볼 수 있는 탭이 있다. 로그인을 안했을 때는 로그인을 해달라는 문구와 버튼이 뜨고, 버튼을 누르면 로그인 페이지로 이동한다.



로그인을 하면 최근 검색한 종목을 최대 5개까지 확인할 수 있다. 종목 이름과 현재 가격, 전일대비 퍼센트를 띄워주고, 종목 이름을 클릭하면 해당 종목 페이지로 이동해 상세한 정보 또한 확인할 수 있다.

#### - 로그인 정보 암호화

웹사이트 계정 보안을 위해 JWT(JSON Web Token)를 이용하여 백엔트 코드를 작성하였다. JWT는 인증에 필요한 정보들을 암호화한 토큰을 말한다. 따라서 Access Token(JWT Token)을 HTTP 헤더에 실어 서버에 전송한다. 토큰은 임의로 생성된 비밀번호처럼 동작하게 되고, 제한된 수명을 가지고 있다. 따라서 새로 만든 토큰의 기간이 만료되면 새로 생성시켜줘야 한다.

```
export default {
  secretKey: "DbGaJoAy0", // 원하는 시크릿 키
  option: {
    algorithm: "HS256", // 해싱 알고리즘
    expiresIn: "30m", // 토큰 유효 기간
    issuer: "issuer", // 발행자
  },
};
```

토큰의 유효 기간은 30분으로 설정해주었고, HS256해싱 알고리즘을 사용하였다.

```
token: jwt.sign(payload, secretKey, options),
```

사용자가 로그인을 시도하게 되면 위와 같은 함수를 통해 id, pw 정보가 들어간 payload와 secretKey, option을 넣고 token을 생성하게 된다.

```
export const handleLogin = async (id, pw) => {
  const res = await axios.post("http://localhost:4000/login", {
    id: id,
    pw: pw,
  });
  if (res.data) {
    console.log("RES DATA : " + res.data);
    if (res.data === "isBanned") {
      return "isBanned";
    }
    console.log("로그인 성공!");
    setTokenToCookie(res.data.token); // cookie에 refresh_token 저장
    setInfoToCookie(res.data); // cookie에 refresh_token 저장
    return true;
  } else {
    console.log("로그인 실패");
    return false;
  }
};
```

프론트엔드 단에서는 사용자가 로그인에 성공하게 되었을 때 발급된 JWT Token을 받아서 setTokenToCookie 함수를 통해 Cookie에 저장하게 된다. 이제 Cookie에 저장된 JWT Token을 이용해서 마이페이지 등 개인정보가 그대로 노출이 되는 부분을 처리할 때 get 방식으로 id, pw를 그대로 넣어서 보내는 것이 아니라 백엔드로 암호화된 Token을 보내주면 안전하게 처리가 되는 것이다.

## - 수익률 계산

계좌
100245593010

홍길동

aaaaa@naver.com

01011111111

수정

자산현황

3,911,000(-37.63 %)

총매입금액 : 6,270,500

당일실현손익 : -0.79%

총평가금액 : 3,911,000

보유주식

종목명	현재가격	구매가격	평가손익	전일대비	전일대비비율
SK하이닉스	78,200	84,300	-7.24%	-600	-0.76%
삼성전자	58,700	166,000	-64.64%	-500	-0.84%

관심종목

종목명	현재가격	등락률	거래량
SK하이닉스	78,200	-0.76%	3,769,573

내가 만든 토론

내가 남긴 댓글

ㅇㅇ 물랐어?

알았어 알았어

자산현황 탭을 통해 내가 총 매입한 금액, 당일수익률, 총평가금액 등의 정보를 통해 총 수익률을 계산해주었다. 그리고 보유주식 탭에서는 보유 주식들의 구매가 대비 현재가를 통해 총

목별로 수익률을 계산해 띄워주었다.

## - 랭킹 시각화

### 상승률 상위

1. 삼성SDI	709,000	+16,000	+2.26%
2. 한국조선해양	74,300	+1,500	+2.02%
3. 현대모비스	210,000	+4,000	+1.90%
4. LG화학	682,000	+12,000	+1.76%
5. POSCO홀딩스	290,000	+3,500	+1.21%
6. 현대차	163,500	+1,500	+0.92%
7. 기아	65,200	+500	+0.77%
8. 넥슨게임즈	13,600	+100	+0.74%
9. 카카오	55,900	+400	+0.72%
10. 한화	27,700	+150	+0.54%

### 하락률 상위

1. 삼성생명	71,200	-1,600	-2.25%
2. 삼성물산	116,500	-1,000	-0.86%
3. 삼성전자	58,700	-500	-0.85%
4. NAVER	184,000	-1,500	-0.82%
5. SK하이닉스	78,200	-600	-0.77%
6. 셀트리온	175,500	-1,000	-0.57%
7. LG	79,200	-400	-0.51%
8. 엔씨소프트	464,500	-1,500	-0.32%
9. 삼성바이오로직스	828,000	+1,000	+0.12%
10. SK텔레콤	50,300	+100	+0.20%

### 거래량 상위

1. 삼성전자	58,700	-500	-0.85%
2. SK하이닉스	78,200	-600	-0.77%
3. KB금융	50,500	+100	+0.20%
4. 카카오	55,900	+400	+0.72%
5. 기아	65,200	+500	+0.77%
6. 삼성생명	71,200	-1,600	-2.25%
7. SK텔레콤	50,300	+100	+0.20%
8. NAVER	184,000	-1,500	-0.82%
9. 현대차	163,500	+1,500	+0.92%
10. POSCO홀딩스	290,000	+3,500	+1.21%

### 시가총액 상위

1. 삼성전자	58,700	-500	-0.85%
2. 삼성바이오로직스	828,000	+1,000	+0.12%
3. SK하이닉스	78,200	-600	-0.77%
4. 삼성SDI	709,000	+16,000	+2.26%
5. LG화학	682,000	+12,000	+1.76%
6. 현대차	163,500	+1,500	+0.92%
7. NAVER	184,000	-1,500	-0.82%
8. 기아	65,200	+500	+0.77%
9. 카카오	55,900	+400	+0.72%
10. 셀트리온	175,500	-1,000	-0.57%

홈 화면에서는 하루동안 가격의 상승률과 하락률이 가장 큰 순서대로 최대 10개씩 랭킹을 시각화해서 보여준다. 또한 하루동안 거래량이 가장 많은 종목별로 최대 10개씩 랭킹을 시각화해서 보여준다. 시가총액도 상위 10개씩 랭킹을 띄워주도록 했다. 색상 또한 상승 시에는 빨간색, 하락 시에는 파란색, 변동이 없을 시에는 검정색으로 두어 사용자가 한눈에 파악하기 쉽도록 하였다. 그리고 각 랭킹은 링크로 두어 종목을 선택하면 종목 상세 페이지로 이동한다.

## - 연령대별 주식 보유 현황 시각화

### 거래량 상위

1. 삼성전자	58,700	-500	-0.85%
2. SK하이닉스	78,200	-600	-0.77%
3. KB금융	50,500	+100	+0.20%
4. 카카오	55,900	+400	+0.72%
5. 기아	65,200	+500	+0.77%
6. 삼성생명	71,200	-1,600	-2.25%
7. SK텔레콤	50,300	+100	+0.20%
8. NAVER	184,000	-1,500	-0.82%
9. 현대차	163,500	+1,500	+0.92%
10. POSCO홀딩스	290,000	+3,500	+1.21%

### 시가총액 상위

1. 삼성전자	58,700	-500	-0.85%
2. 삼성바이오로직스	828,000	+1,000	+0.12%
3. SK하이닉스	78,200	-600	-0.77%
4. 삼성SDI	709,000	+16,000	+2.26%
5. LG화학	682,000	+12,000	+1.76%
6. 현대차	163,500	+1,500	+0.92%
7. NAVER	184,000	-1,500	-0.82%
8. 기아	65,200	+500	+0.77%
9. 카카오	55,900	+400	+0.72%
10. 셀트리온	175,500	-1,000	-0.57%

### 연령별 보유량 상위

	20대	30대	40대	50대
1. SK하이닉스			78,200	-600 -0.76%
2. 삼성전자			58,700	-500 -0.84%
3. 현대차			163,500	+1,500 +0.93%
4. 삼성바이오로직스			828,000	+1,000 +0.12%
5. LG화학			682,000	+12,000 +1.79%

### 연령별 보유량 상위

	20대	30대	40대	50대
1. 현대차			163,500	+1,500 +0.93%
2. LG화학			682,000	+12,000 +1.79%
3. 삼성바이오로직스			828,000	+1,000 +0.12%

홈 화면에서는 연령대 별 보유한 종목을 상위 최대 5개까지 띄워준다. 각 연령대는 탭으로 구성되어 클릭하면 데이터에 따라 랭킹이 변동되는 것을 확인할 수 있다.

### - 오늘의 뉴스

stockmarket

검색어를 입력하세요

로그인/회원가입

홈

공지사항

MY

오늘의 뉴스

TSMC '400억불 투자'에 삼성 '1년에 1공장' 맞불... 설비경쟁 후끈 [워클리반도체]

매일경제

삼성전자, 네이버와 AI 손잡은 이유

비즈니스위치

[단독] "삼성 디자인에 힘준다"...노태문 사장 "디자인경영센터장" 겸직

머니투데이

LG화학, 아시아 최초 '식물성 원료 기반' 친환경 ABS 출시

해럴드경제

배터리·소재 강자끼리 뿔쳤다' LG화학·고려아연 주식 맞교환

한경비즈니스

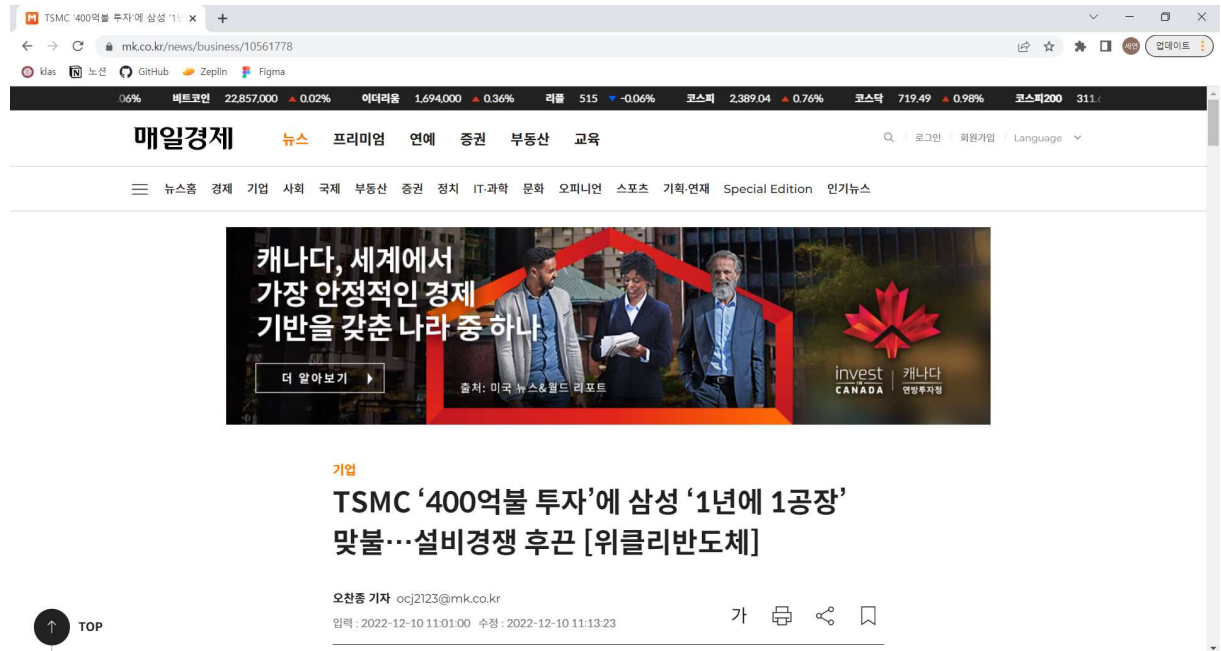
최근 검색

로그인하시면, 나의 최근 검색을 볼 수 있습니다.

로그인

홈 화면에서는 오늘의 뉴스 5개를 띄워준다. 뉴스 제목, 언론사를 보여주고 해당 뉴스를 클릭하면 url로 넘어가 뉴스 전체를 읽을 수 있다.





#### 4. SQL 설명

##### 로그인(login.js)

```
"SELECT id, name, ban FROM user WHERE id = ? and pw = ?";
```

로그인에 사용되는 SQL이다. Id와 pw를 입력 받아 user Table에서 data를 받는다. Data가 있다면 로그인이 정상적으로 완료될 수 있고, data가 없다면 로그인이 되지 않는다.

##### 관리자 로그인(adminLogin.js)

```
"SELECT id FROM admin WHERE id = ? and pw = ?";
```

관리자 로그인에 사용되는 SQL이다. Id와 pw를 입력 받아 admin Table에서 data를 받는다. Data가 있다면 로그인이 정상적으로 완료될 수 있고, data가 없다면 로그인이 되지 않는다.

##### 회원가입(register.js)

```
"SELECT * FROM user WHERE id = ?";
```

회원가입시 아이디가 중복인지 확인하는 SQL이다. ID를 입력으로 받아 user Table에서 data를 받는데 data가 있다면 아이디가 중복인 것이다.

```
"INSERT into user(id, pw, name, phone, email, age) values(?,?,?,?,?,?)";
```

이후 회원가입 폼의 내용을 다 입력한 후 회원가입 버튼을 누르면 값들이 user table에 insert되어 회원가입이 완료된다.

##### 사용자 보유주식 현황 정보 (calculatestock.js)

```

SELECT code,
stock_name 종목명,
n_price 현재가격,
sum(stock_num*stock_price)/sum(stock_num) 구매가격,
(n_price-(sum(stock_num*stock_price)/sum(stock_num)))/(sum(stock_num*stock_price)/sum(stock_num))*100 평가손익,
n_price-e_price 전일대비, (n_price-e_price)/e_price*100 전일대비비율
FROM STOCK_U natural join STOCK_P natural join STOCK
WHERE ID=? and date='2022-12-08'
group by code

```

사용자 보유주식 테이블인 STOCK\_U, 주식 가격정보를 가지고 있는 STOCK\_P, 종목 코드와 회사이름을 가지고 있는 STOCK 테이블은 natural join으로 join

사용자가 가지고 있는 종목의 주식 수 (stock\_num) \* 해당 주식의 구매당시의 가격 (stock\_price)를 sum한 후 해당 종목의 보유 주식 수로 나누어 해당 주식에 대한 평균 구매 가격 확인

위에서 확인한 구매가격을 현재 가격에서 빼 후 구매가격으로 나누어 %를 취해 해당 종목에 대한 평가손익 확인

종목의 현재가격(n\_price)에 전날 종가(e\_price)를 빼 전일대비 가격변동 확인

위 전일 대비 가격 변동을 증가로 나눈 후 %를 취해 전일대비 가격 변동 비율 확인

### 사용자 정보 비교 (checkuser.js)

```

SELECT t_id FROM u_board WHERE id = ?

```

종목 토론게시판에서 사용자가 글을 작성하였을 때 해당 글에 대해 작성한 사용자만 수정 혹은 삭제 기능을 사용할 수 있도록 한다.

위 SQL문을 통해 게시글의 primary key인 t\_id 값을 게시글 정보 테이블인 u\_board에서 유저 정보인 id를 통해 확인 후 api에서 param으로 받은 값과 비교하여 사용자 정보를 확인

### 게시글, 공지사항, 댓글 삭제 (checkuser.js, deleteboard.js, deletecom.js)

```

DELETE FROM D_BOARD WHERE t_id = ?
DELETE FROM U_BOARD WHERE t_id = ?
DELETE FROM A_BOARD WHERE postnum = ?

```

공지사항 테이블 A\_BOARD, 게시글 테이블 U\_BOARD, 댓글 테이블 D\_BOARD에 대한 삭제 기능을 수행하는 SQL이다. 공지사항은 관리자만 삭제 가능하며 게시글은 작성자만 삭제 가능하다.

### 조회수 업데이트 (updateboard.js, updatedisview.js)

```

UPDATE u_board SET num = ? WHERE t_id = ?
UPDATE a_board SET num = ? WHERE postnum = ?

```

각각 공지사항과 종목토론 게시글의 조회수 기능을 지원하는 SQL이다. Num 값을 기존 num + 1로 업데이트 하면 해당 글을 확인할 때 마다 값이 1씩 늘어나게 된다.

### 게시글, 공지사항 수정 (modifydis.js, modifyboard.js)

```

UPDATE U_BOARD SET title=?, text=?, time = ? WHERE t_id = ?
UPDATE A_BOARD SET title=?, text=?, date = ? WHERE postnum = ?

```

각각 공지사항과 종목토론 게시글의 수정하는 SQL이다. 사용자가 수정 정보를 입력하면 해당 정보를 post를 통해 api로 가져온 후 해당 값으로 update한다. Update가 되었다면 true를 return 아닐 경우 false를 return한다.

### 게시글, 댓글, 공지사항 출력 (dis.js, getaboard.js, getcommentbytid.js)

```
SELECT * FROM U_BOARD WHERE code = ?
```

```
SELECT * FROM A_BOARD
```

```
SELECT * FROM D_BOARD WHERE t_id = ?
```

각각 공지사항, 종목게시판, 댓글을 출력하기 위해 해당 정보를 요청하는 SQL이다. 공지사항의 경우 글을 나누지 않고 일괄 출력하기 때문에 where 조건을 주지 않았다. 하지만 게시글은 주식별로 나누어 출력, 댓글은 게시글 별로 나누어 출력하기 때문에 각각 where 조건을 주어 정보를 요청한다.

게시글, 댓글 공지사항 추가 (uploaddis.js, uploadcomment.js, uploadaboard.js)

```
INSERT into A_BOARD(title, text, date) values(?,?,?)
```

```
INSERT into D_BOARD(id, t_id, text, time) values(?,?,?,?)
```

```
INSERT into U_BOARD(id, code, title, text, time) values(?,?,?,?,?)
```

각각 공지사항, 종목게시판, 댓글을 추가하는 SQL이다. 사용자 혹은 관리자가 입력한 정보를 post를 통해 데이터 베이스에 값을 insert하여 성공하였을 때 true 반환 실패하였을 때 false를 반환한다.

검색 기능 (searchstock.js)

```
SELECT * FROM stock natural join stock_p WHERE date = '2022-12-08' and stock_name like "searchWord"
```

메뉴 바 상단에서 주식 정보에 검색할 때 사용하는 SQL이다. 현재 날짜와, 검색어와 like를 통해 기업 이름을 비교하여 해당 조건에 맞는 주식 정보를 반환한다. 해당 정보를 search 페이지에 반환하여 search 페이지를 구성한다.

사용자 정보 조회 및 수정 (userUpdate.js, usergetdata.js)

```
select * from user where id = ?
```

```
UPDATE user SET email = ?, phone = ?, age = ? WHERE id = ?
```

```
SELECT pw FROM user WHERE id = ?
```

```
UPDATE user SET pw = ? WHERE id = ?
```

사용자 id 정보를 토큰으로 읽어와 사용자 정보를 출력한다. 위 출력한 사용자 정보에 대한 수정 기능을 UPDATE를 통해 수행한다.

사용자의 비밀번호의 경우 재설정할 때 위 select 문을 통해 기존 비밀번호를 확인한 후 새로 입력하는 비밀번호로 업데이트 한다.

자산현황 및 계좌조회 (usermoneydata.js, getbankacc.js)

```
select
```

```
(sum(stock_num*n_price)-sum(stock_num*e_price))/sum(stock_num*e_price)*100 당일실현손익,  
sum(stock_num*stock_price) 총매입금액, sum(stock_num*n_price)총평가금액
```

```
from stock_u natural join stock_p natural join stock
```

```
where id=? and date='2022-12-08'
```

```
group by id
```

사용자가 현재 가지고 있는 주식에 대하여 sum(종목별 주식수(stock\_num)\*해당 주식 개당 현재 가격(n\_price)) / sum(종목별 주식수(stock\_num)\*해당 주식 전일 종가(e\_price) \* 100 을 통해 사용자가 보유한 전체 주식의 당일 실현 손익을 출력한다.

사용자가 종목별 가진 주식 수량 \* 해당 종목 구입가격을 통해 총 매입 금액을 출력한다.

사용자의 종목별 가진 주식 수량 \* 해당 종목의 현재가격을 통해 총 평가금액을 확인한다.

위 정보는 사용자 보유주식 테이블인 (stock\_u)와 주식 현재가격(stock\_p) 테이블 주식회사정보(stock)을 natural join하여 확인하며 사용 시간을 2022-12-08일로 가정 사용자 id는 토큰



을 통해 확인한다.

```
SELECT * FROM USER_A WHERE id = ?
```

위 SQL을 통해 사용자 계좌정보를 확인한다.

관심종목 정보조회 (favoritestock.js)

```
select code,
       stock_name,
       n_price, (n_price-e_price)/e_price*100 전일대비비율, price_count
from stock_uz natural join stock natural join stock_p natural join company
where id = ? and date='2022-12-08' group by code
```

사용자가 관심설정된 주식에 대한 정보를 출력하는 SQL이다. 해당 주식 정보는 주식 코드, 회사 이름, 주식 현재가, 거래량, 현재가(n\_price)-전일종가(n\_price)/전일종가(n\_price)의 비율을 확인한다. 사용자 정보인 id와 사용 날짜인 2022-12-08를 조건으로 데이터를 가져온다.

뉴스정보 출력 (getnews.js)

```
SELECT * FROM N_BOARD
```

위 select 문을 통해 뉴스 정보 테이블인 N\_BOARD를 확인한다.

최근검색 (getRecentStock.js, updateRecentStock.js)

```
"SELECT * FROM STOCK_UC WHERE id = ?;"
```

주식 상세 페이지에 들어갈 때 실행하는 SQL이다. STOCK\_UC table은 사용자가 어떤 주식에 언제 방문했는지 저장하고 있는 table이다. 어떤 주식에 방문할 때 위와 같은 SQL을 실행해 데이터가 있으면 방문한 적이 있는 주식이고 그렇지 않다면 처음 방문하는 주식이다.

```
"DELETE FROM STOCK_UC WHERE id = ? AND code = ?;"
```

이전에 방문했던 주식이라면 이전에 방문했던 기록은 삭제한다.

```
"INSERT into STOCK_UC(id, code, time) VALUES(?,?,?);"
```

그리고 새로 방문한 기록의 data를 추가한다.

```
14 • select code, stock_name name , n_price price, (n_price-e_price)/e_price*100 rate
15 from STOCK_UC natural join stock_p natural join stock
16 where id = 'aaaaa' and date='2022-12-08'
17 order by time DESC LIMIT 0, 5;
```

code	name	price	rate
006400	삼성SDI	709000	2.3088
123123	123123	100000	0.0000
032830	삼성생명	71200	-2.1978
028260	삼성물산	116500	-0.8511
012330	현대모비스	210000	1.9417

User id를 입력 받아 최근 검색한 주식을 찾는다. Stock\_uc에 사용자가 주식에 방문 했을 때 시간 데이터가 있어 시간을 내림차순 해 최근 검색한 주식을 찾을 수 있다. Stock\_p와 Stock table과 join하여 주식 이름, 주식 가격, 변동률 등 데이터를 가져온다. Limit을 통해 5개의 데이터만 가져온다.

### 하락률 상위 랭킹(fallingRate.js)

```
178 • SELECT code, stock_name name, n_price price, n_price - e_price difference, (n_price - e_price)/n_price * 100 rate
179 FROM STOCK_P NATURAL JOIN STOCK
180 WHERE date = "2022-12-08"
181 ORDER BY (n_price - e_price)/n_price
182 LIMIT 0,10;
183
```

code	name	price	difference	rate
032830	삼성생명	71200	-1600	-2.2472
028260	삼성물산	116500	-1000	-0.8584
005930	삼성전자	58700	-500	-0.8518
035420	NAVER	184000	-1500	-0.8152
000660	SK하이닉스	78200	-600	-0.7673
068270	셀트리온	175500	-1000	-0.5698
003550	LG	79200	-400	-0.5051
036570	엔씨소프트	464500	-1500	-0.3229
123123	123123	100000	0	0.0000
207940	삼성바이오로직스	828000	1000	0.1208

(시가-종가)/시가 를 통해 주식 가격의 변동률을 계산한다. 이렇게 계산한 변동률을 통해 오름 차순 하면 하락률 상위 랭킹 data를 가져올 수 있다. Limit를 통해 10개의 데이터만 가져온다.

### 상승률 상위 랭킹(risingRate.js)

```
178 • SELECT code, stock_name name, n_price price, n_price - e_price difference, (n_price - e_price)/n_price * 100 rate
179 FROM STOCK_P NATURAL JOIN STOCK
180 WHERE date = "2022-12-08"
181 ORDER BY (n_price - e_price)/n_price DESC
```

code	name	price	difference	rate
006400	삼성SDI	709000	16000	2.2567
009540	한국조선해양	74300	1500	2.0188
012330	현대모비스	210000	4000	1.9048
051910	LG화학	682000	12000	1.7595
005490	POSCO홀딩스	290000	3500	1.2069
005380	현대차	163500	1500	0.9174
000270	기아	65200	500	0.7669
225570	넥슨게임즈	13600	100	0.7353
035720	카카오	55900	400	0.7156
000880	한화	27700	150	0.5415

(시가-종가)/시가 를 통해 주식 가격의 변동률을 계산한다. 이렇게 계산한 변동률을 통해 내림 차순 하면 상승률 상위 랭킹 data를 가져올 수 있다. Limit를 통해 10개의 데이터만 가져온다.

### 거래량 상위 랭킹(volume.js)

```
9 • SELECT code, stock_name name, n_price price, n_price - e_price difference, (n_price - e_price)/n_price * 100 rate, price_count
0 FROM STOCK_P NATURAL JOIN STOCK
1 WHERE date = '2022-12-08'
2 ORDER BY price_count
3 DESC LIMIT 0,10;
```

code	name	price	difference	rate	price_count
005930	삼성전자	58700	-500	-0.8518	12779480
000660	SK하이닉스	78200	-600	-0.7673	3769573
105560	KB금융	50500	100	0.1980	1869444
035720	카카오	55900	400	0.7156	1364132
000270	기아	65200	500	0.7669	1262664
032830	삼성생명	71200	-1600	-2.2472	870927
017670	SK물류	50300	100	0.1988	746856
035420	NAVER	184000	-1500	-0.8152	706472
005380	현대차	163500	1500	0.9174	701414
005490	POSCO홀딩스	290000	3500	1.2069	576109

Stock\_p의 price\_count인 거래량을 정렬하여 거래량 상위 랭킹을 보여준다.

## 시가총액 랭킹(marketCap.js)

```
5 • SELECT code, stock_name name, n_price price, n_price - e_price difference, (n_price - e_price)/n_price * 100 rate , stock_count*n_price
6 FROM STOCK_P NATURAL JOIN STOCK NATURAL JOIN COMPANY
7 WHERE date = '2022-12-08'
8 ORDER BY stock_count*n_price DESC
9 LIMIT 0,10;
```

code	name	price	difference	rate	stock_count*n_price
005930	삼성전자	58700	-500	-0.8518	350426235685000
207940	삼성바이오로직스	828000	1000	0.1208	5893207200000
000660	SK하이닉스	78200	-600	-0.7673	5692978455200
006400	삼성SDI	709000	16000	2.2567	4875405177000
051910	LG화학	682000	12000	1.7595	4814397588000
005380	현대차	163500	1500	0.9174	3493474743000
035420	NAVER	184000	-1500	-0.8152	3018503072000
000270	기아	65200	500	0.7669	2642968976800
035720	카카오	55900	400	0.7156	2489289496200
068270	셀트리온	175500	-1000	-0.5698	2470897143000

시가총액은 상장주식수 \* 시가를 통해 구할 수 있다. 이를 내림차순으로 정렬하여 시가총액 랭킹을 구할 수 있다.

## 연령별 주식 보유 현황(getAgeStock.js)

```
160 • WITH AGE AS (
161     SELECT ID AS ID,
162     ( CASE
163         WHEN age >= 20 AND age <= 29 THEN '20'
164         WHEN age >= 30 AND age <= 39 THEN '30'
165         WHEN age >= 40 AND age <= 49 THEN '40'
166         WHEN age >= 50 AND age <= 59 THEN '50'
167         WHEN age >= 60 THEN '60'
168         ELSE '10'
169     END) AS age
170 FROM user )
171 SELECT code, stock_name name, n_price price, n_price-e_price dif, (n_price-e_price)/e_price*100 rate
172 FROM AGE natural join STOCK_U natural join STOCK natural join STOCK_P
173 where age = '20'
174 group by age, code
175 order by sum(stock_num) DESC
176 LIMIT 0, 5;
```

code	name	price	dif	rate
000660	SK하이닉스	78200	-600	-0.7614
005930	삼성전자	58700	-500	-0.8446
005380	현대차	163500	1500	0.9259
207940	삼성바이오로직스	828000	1000	0.1209
051910	LG화학	682000	12000	1.7910

CASE를 사용해 연령별로 구분한다. 그리고 WITH를 사용해 STOCK\_U TABLE, STOCK, STOCK\_P Table과 join한다. Age와 code로 그룹하고 stock\_num의 합으로 내림차순 정렬하여 연령별로 많이 가지고 있는 주식을 뽑는다. Limit를 0, 5로 하여 5개만 뽑을 수 있다. 그렇게 뽑은 data 중 주식코드, 주식이름, 가격, 전날대미가격, 변동률을 사용한다.

## 주식 상세 정보(getStockInfo.js)

```
"SELECT * FROM STOCK_P natural join STOCK WHERE code= ? and date='2022-12-08';",
```

주식 상세 페이지에서 주식 정보를 가져오는 SQL이다. Code와 data를 입력 받아 code에 맞는 주식에서 date에 해당하는 주식의 상세 정보들을 가져온다.

## 기업 개요(getCompanyInfo.js)

```
"SELECT * FROM COMPANY WHERE code= ?;"
```

주식 상세 페이지에서 해당 주식의 회사에 대한 개요를 가져오는 SQL이다. Code를 입력받아 COMPANY Table에서 data를 가져온다.

#### 주주 정보(getShareholderInfo.js)

```
"SELECT * FROM stock_o WHERE code= ? order by stock_num DESC;"
```

주식 상세 페이지에서 해당 주식에 대한 주주 정보를 가져오는 SQL이다. Code를 입력 받아 stock\_o Table에서 주주 정보를 가져온다. Stock\_num 정렬을 통해 주주 data를 보유 주식 순서로 정렬할 수 있다.

#### 종목 토론실 데이터 연결(getDisInfo.js)

```
11 • select title, count(*) num
12 from u_board join D_BOARD on u_board.t_id = D_BOARD.t_id
13 where code = '005930'
14 group by u_board.t_id
15 order by u_board.time DESC;
```

sult Grid		Filter Rows:	Exports:	Wrap Cell Content:
title	num			
거의 보험인듯	1			
삼성전자 신제품 나온다던데??	1			
삼성전자 믿읍시다!	3			
삼성전자 알잖아	1			

주식 상세 페이지에서 해당 주식에 대한 종목 토론실 데이터를 가져오는 SQL이다. Code를 입력 받아 U\_Board Table에서 해당 code에 대한 종목 토론 게시판 글 data를 가져온다. 이를 D\_Board Table와 join해서 각 글에 대한 댓글 데이터도 가져온다. 종목 토론 게시판의 글의 id로 그룹 하여 댓글의 개수를 count 하고, 글 제목과 함께 가져온다. Data를 작성시간 순서로 정렬하여 최근 글 순서로 가져온다.

#### 매도/매수(stockOrder.js)

##### 매수

```
"SELECT stock_num FROM STOCK_U WHERE id=? and code=? and stock_price=?"
```

Id, code, stock\_price를 입력 받아 구매 주식 table에 데이터가 있는지 확인한다.  
데이터가 있다면,

```
"UPDATE STOCK_U SET stock_num=? WHERE id=? and code=? and stock_price=?;"
```

Update를 통해 구매 주식 table에 데이터에서 수량만 update 해주고,  
데이터가 없다면,

```
"INSERT into STOCK_U(id, code, stock_num, stock_price) values(?,?,?,?);" ,
```

Insert를 통해 구매 주식 table에 data를 넣어준다.

##### 매도



```

17 • SELECT code , sum(stock_num) num, sum(stock_num*stock_price)/sum(stock_num) price
18 FROM STOCK_U natural join STOCK
19 WHERE ID='aaaaa' and code='005930'
20 group by code;

```

code	num	price
005930	20	166000.0000

ID와 매도 하려는 주식의 code를 입력 받아 보유하고 있는 주식의 총 개수와 그것들의 구매 가격의 평균을 구한다. 매도하기 위해선 주식을 1개이상 갖고 있어야 하기 때문에 data가 없다면 매도할 수 없다.

```
"DELETE FROM STOCK_U WHERE id = ? AND code = ?";
```

그 후 구매 주식 table에 id와 code에 해당하는 data를 모두 지우고

```
"INSERT into STOCK_U(id, code, stock_num, stock_price) values(?,?,?,?);"
```

Id와 code, 그리고 매도하고 남은 주식의 개수와 갖고 있던 주식들의 평균 구매 가격을 INSERT한다.

#### 종목 추가(addstock.js)

```
"SELECT * FROM STOCK WHERE code = ?";
```

Code 중복 검사를 할 때 SELECT문을 통해 code가 있는지 확인한다.  
이후 중복되는 code가 없으면

```
"INSERT into STOCK(code, stock_name) values(?,?)",
```

```
"INSERT into stock_p(code) values(?)",
```

```
"INSERT into COMPANY(code, company_name, stock_count, company_info) values(?,?,?,?)",
```

```
"INSERT into stock_O(code, stock_owner, stock_num, stock_p) values(?,?,?,?)",
```

를 차례로 한다. STOCK에 INSERT를 먼저 해야 다른 테이블에서 code를 FK로 받을 수 있기 때문에 STOCK에 data를 먼저 INSERT 하지 않으면 오류가 난다.

#### 종목 정보 수정(modifystock.js)

```
"UPDATE COMPANY SET company_name=?, stock_count=?, company_info = ? WHERE code = ?";
```

기업 이름, 상장 주식 수, 기업 정보, 주식 code를 입력 받아 code에 해당하는 COMPANY TABLE의 data를 update 한다.

```
"UPDATE STOCK SET stock_name=? WHERE code = ?";
```

주식 이름도 입력 받아 STOCK TABLE을 수정한다.

#### 회원관리(getMemberList.js, banControl.js)

```
"SELECT * FROM user";
```

User Table의 정보를 가져와 모든 사용자의 정보를 출력한다.

```
"UPDATE user SET ban = ? WHERE id = ?";
```

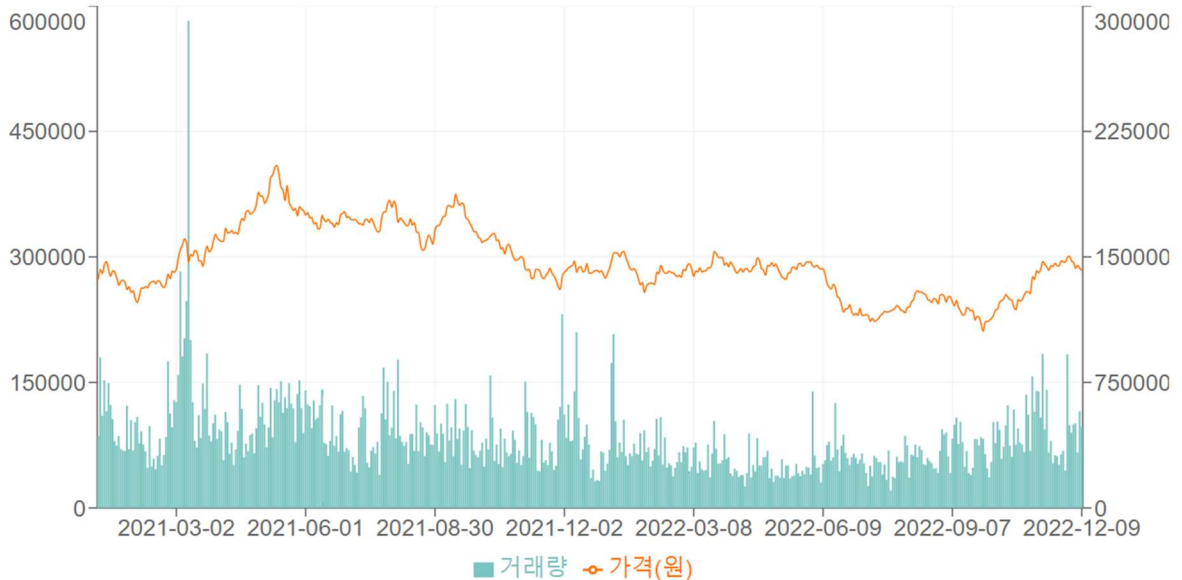
Ban과 id를 통해 사용자 ban 정보를 update한다.

### 5. 데이터 시각화

주어진 주식 별 가격 현황, 거래량, 종가, 시가, 고가, 저가 등의 데이터를 시각화 하기위해 Rechart 라이브러리를 선택하였습니다.

# Recharts

Rechart는 비교적 코드가 단순하여 적용하기 편리한 장점이 있습니다.



위 그래프는 데이터 시각화 결과 예시입니다. 주식의 가격 변동과 거래량 변동을 동시에 표현하기 위해 콤비네이션 차트를 사용하였으며, 주황색 그래프는 해당 주식의 가격 변동현황을 나타내며 가격은 오른쪽 Y축을 통해 확인할 수 있습니다. 초록색 막대 그래프는 해당 주식의 거래량 변동현황을 나타내며 왼쪽 Y축을 통해 확인할 수 있습니다. 아래 X 축은 두 그래프 공통인 날짜입니다.

## 6. YouTube 및 GitHub 링크

Github Repository URL: [https://github.com/lllminsuk/DB\\_Project](https://github.com/lllminsuk/DB_Project)

YouTube Video URL: <https://www.youtube.com/watch?v=RWGqRKc-8mY>

## 7. 고찰

이름	역할
박준용	데이터 시각화 차트 제작, Database 설계 및 수정, ER diagram 설계, DB query 문 작성, 관심 종목 정보API 제작, 보유 주식 정보 API 제작, 사용자별 총 자산평가 API 제작, 회원 정보 출력 및 수정 API 제작, 내가 쓴 글 및 댓글 확인, 작성, 수정, 삭제 API 제작, 관리자 공지사항 확인, 작성, 수정 삭제 API 제작, 검색기능 API, 뉴스 출력 및 연동 API, MY 페이지 자산평가, 보유주식평가, 관심종목, 회원정보, 내가 쓴 글, 내가 쓴 댓글 데이터 연동, 종목토론게시판 UX/UI 제작 및 데이터 연동, 공지사항 게시판 페이지 UX/UI 제작 및 데이터 연동, 검색페이지 제작 및 데이터 연동, 뉴스 테이블 제작 및 데이터 연동, 회원정보 수정 페이지 UX/UI 제작 및 데이터 연동, 발표 YouTube 영상 제작
전승권	테이블 별 테스트 데이터 생성 및 삽입

이민석	Database 설계 및 수정, DB query문 작성, Test Data 생성 및 수정, 로그인 API 구현, 회원가입 API 제작, 최근검색 API 제작, 상승률 상위 랭킹 API 제작, 하락률 상위 랭킹 API 제작, 거래량 상위 랭킹 API 제작, 시가 총액 상위 랭킹 API 제작, 연령별 주식 보유 현황 랭킹 API 제작 및 UI/UX 제작, 종목 정보 페이지 UI/UX 제작 및 데이터 연동 API 제작, 매도/매수 UI/UX 제작 및 데이터 연동 API 제작, 관리자 종목 추가 API 제작, 종목 정보 수정 페이지 UI/UX 제작 및 데이터 연동 API 제작, 회원 관리 페이지 UI/UX 제작 및 데이터 연동 API 제작, 페이지 Defect 테스트
이세연	사용자 회원가입 페이지 제작 및 연동, 관리자 로그인 페이지 제작 및 연동, 네비게이션 바 UI/UX, 검색 버튼 기능 구현, 홈 화면 UI/UX, 최근 검색 기능 UI/UX, 마이페이지 디자인, 토론게시판 및 공지사항 페이지 디자인, 종목 추가 페이지 제작 및 디자인, 전체적인 페이지 디자인 수정, 페이지 Defect 테스트, 웹 디자인 설계

**박준용** - 이전에 SpringBoot를 이용한 개인 웹프로젝트의 경험은 있지만, 팀 웹프로젝트는 처음 이었다. 또한 react 기반 프로젝트도 처음이라 기본기가 많이 부족하여 공부하며 프로젝트를 진행해야 하는 어려움이 있었다. 또한 깃허브를 이용하여 프로젝트를 진행하였는데 이에 대한 중요성을 다시한번 알고 부족했던 부분도 제대로 알게 되었다. 이번 프로젝트에서는 프론트 엔드 웹 디자인 부분 공부가 미흡한 점이 많아 다음 프로젝트는 이에 대해 준비 후 한다면 더 좋은 결과의 프로젝트를 할 수 있을 것 같다. 이번 과제를 통해 DB query 문, API 설계, 백 엔드 서버 데이터 연동 등 많은 작업을 진행하여 한층 더 웹 및 DB에 대해 잘 알게 된 좋은 기회였다.

**전승권** - 이번 프로젝트는 4인으로 진행했던 프로젝트여서 버전 관리가 상당히 중요했다. 그렇기 때문에 다양한 깃 명령어를 사용해 작업을 했어야 했고 이번 기회에 사용해볼 수 있어서 좋았다. 또한 개인적인 사유로 인해 프로젝트에 많이 참여하지 못해서 팀원들에게 많이 고마웠고, 작게나마 도움이 될 수 있어서 다행이었다. 비록 맡은 일은 테스트 데이터 전처리와 테이블 데이터 삽입 정도의 일이었지만 쿼리 작성을 해볼 수 있어서 좋은 경험이 되었다고 생각한다.

**이민석** - 소프트웨어공학 수업 때 웹 개발 팀 프로젝트 이후로 두 번째 웹 개발 팀 프로젝트였다. 소프트웨어공학 수업에선 팀 협업에 중요한 깃허브를 중요하게 생각하며 웹 개발을 진행했다면 이번 데이터베이스 및 데이터 시각화 수업에서는 DB에 중점적으로 여겨 웹 개발을 진행한 것 같다. 하지만 소프트웨어공학 수업에서 배웠던 깃허브의 중요성을 알고 이번 프로젝트도 팀원들과 열심히 깃 허브를 사용해 프로젝트를 진행했다. 소프트웨어공학 프로젝트에서는 팀원이 백 엔드와 프론트 엔드로 나뉘어 개발을 진행했지만 이번 프로젝트에선 모든 팀원이 DB를 해볼 수 있게 백 엔드와 프론트 엔드로 나누지 않고 가능한 기능별로 파트를 나누어 프로젝트를 진행했다. 데이터베이스설계와 query문, API 작성을 하면서 PK, FK에 대한 문제가 있었지만 팀원들과 상의하고 수정하면서 프로젝트를 끝마칠 수 있었다. 데이터베이스설계에 대한 아쉬움이 있지만 이번 프로젝트를 경험으로 다음에 데이터베이스를 설계하게 된다면 더 잘할 수 있을 것 같다.

**이세연** - 두 번째 웹 개발 팀 프로젝트인데 확실히 이전보다 깃허브에 형상관리를 쉽게 할 수 있었고, 더욱 중요성을 느낄 수 있었다. 프론트 엔드 위주로 담당했는데 백 엔드에서 데이터를 연동하면 바로 웹 페이지에 띄울 수 있도록 UI를 만드는 것의 중요성을 느꼈고, 또 그렇게 하기 위해 노력을 많이 했다. 가져올 정보가 단순한 페이지는 백 엔드 API를 만들어 쿼리문도 작성해봤는데 수업을 통해 배운 것을 실습할 수 있어 좋았고, 아직까지는 마냥 어렵게만 느껴지는 것 같다. 조금 더 공부해서 프론트 엔드와 백 엔드 모두 잘 해보고 싶다.