

주식 프로젝트

파이선 크롤링을 통해 MongoDB에 데이터 저장. 각각 Stocks, Predictions, Histories 저장. => 백엔드에서 엔티티(스키마)를 먼저 생성한 후, 자동 생성된 MongoDB 컬렉션에 파이썬으로 데이터 넣으면 됨

Stocks 컬렉션 (주식 데이터)

The screenshot shows the MongoDB Compass interface for the 'test.stocks' collection. The left sidebar lists databases and collections, with 'test' and 'stocks' highlighted. The main panel shows the 'Documents' tab, displaying a list of documents. Each document contains an '_id', a 'timestamp', and a 'data' array. The 'data' array contains an array of stock price objects with fields like 'Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', and 'Volume'.

Prediction 컬렉션 (예측 데이터)

The screenshot shows the MongoDB Compass interface for the 'test.predictions' collection. The left sidebar lists databases and collections, with 'test' and 'predictions' highlighted. The main panel shows the 'Documents' tab, displaying a list of documents. Each document contains an '_id', a 'timestamp', and a 'data' array. The 'data' array contains an array of prediction objects with fields like 'Date', 'Close', and 'Volume'.

Histories 컬렉션 (누적된 과거 데이터)

The screenshot shows the MongoDB Compass interface for the 'test.histories' collection. The left sidebar lists databases and collections, with 'test' and 'histories' highlighted. The main panel shows the 'Documents' tab, displaying a list of documents. Each document contains an '_id', a 'timestamp', and a 'data' array. The 'data' array contains an array of historical stock price objects with fields like 'Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', and 'Volume'.

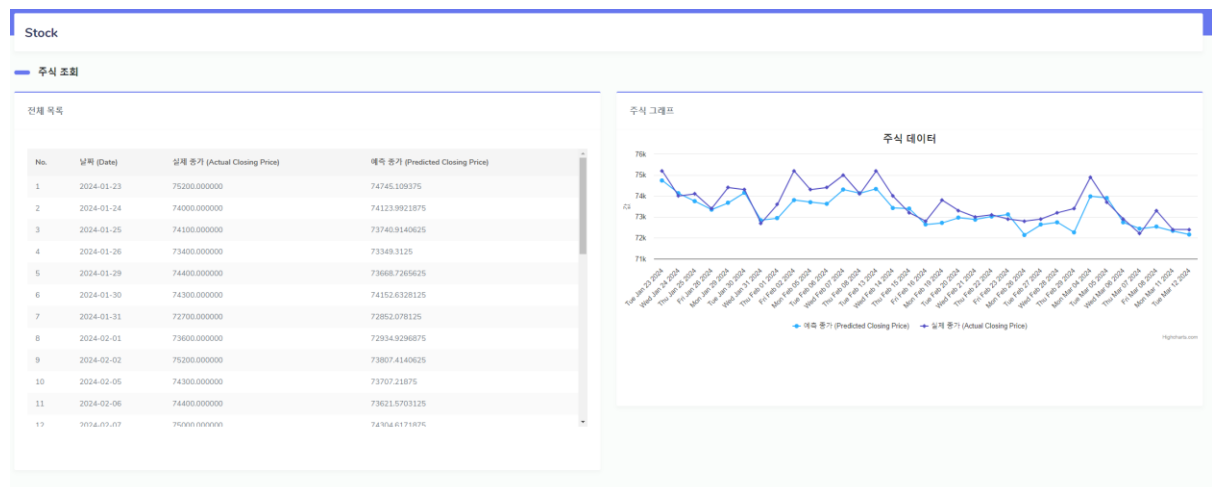
주식데이터 비교 로직

1. stocks와 timeStamp가 동일한 prediction의 데이터를 조회.
2. 해당 prediction과 timestamp가 동일한 histories의 데이터를 조회.
3. prediction과 histories의 날짜별 종가(Close)를 비교하는 그래프생성(High - chart 사용)

-주식 조회 목록

Stocks							
주식 조회							
전체 목록							
No.	파일명	파일 생성시간	시가 (Open)	고가 (High)	저가 (Low)	종가 (Close)	거래량 (Volume)
1	65f107c69a238b1cfef1252d	2024-03-13 19:56	73,700	74,100	73,500	73,900	5,749,770
2	65eff039580ece263a5f520d	2024-03-13 00:03	72,600	73,100	72,100	72,900	9,900,837
3	65efce425815a11f7d650329	2024-03-12 21:38	72,600	72,900	72,100	72,900	6,898,387
4	65efbdcf31264e3c5ab1e7c0	2024-03-12 20:28	72,600	72,700	72,100	72,500	5,231,626
5	65efbd92bb2d018bb807871d	2024-03-12 20:27	72,600	72,700	72,100	72,500	5,231,626
6	65efbce85214b58da606abbf	2024-03-12 20:24	72,600	72,700	72,100	72,500	5,207,947
7	65efb1728411c10eaa0c99f9	2024-03-12 19:35	72,600	72,700	72,100	72,300	3,464,487

위의 파일명을 클릭하면, 해당 timestamp를 파라미터로 전송하여 예측종가와 실제종가를 데이터 형태 및 그래프 형태로 표현 (High-Chart 사용)



백엔드 : Node.js – Nest.js 프레임워크

프론트엔드 : HTML, JavaScript, CSS

데이터베이스 : MongoDB

그 외 기능

1. 유저 회원가입, 2. 로그인, 3. 로그아웃, 4. 내정보 조회, 5. 회원정보 변경, 6. 회원 탈퇴
- 모든 데이터 조회 기능 API는 유저의 로그인상태 확인 후 가능, 즉 로그인된 상태의 유저만 서비스 사용 가능.
 - 유저의 정보 중 password는 보안성을 위해 hash처리하여 데이터베이스에 저장, 로그인 시 accessToken 및 refreshToken 발급. 로그아웃 시 refreshToken 제거

로그인

이메일 (E-mail)

비밀번호

계정이 없으신가요? [회원가입](#)

로그인

회원가입

이메일 (E-mail)

비밀번호 (Password)

비밀번호 확인 (Confirm Password)

닉네임 (Nickname)

소속 (Organization)

☒ 관리자 회원 (Admin)
☐ 일반 회원 (Guest)

이미 회원가입 하셨나요? [로그인](#)

회원가입

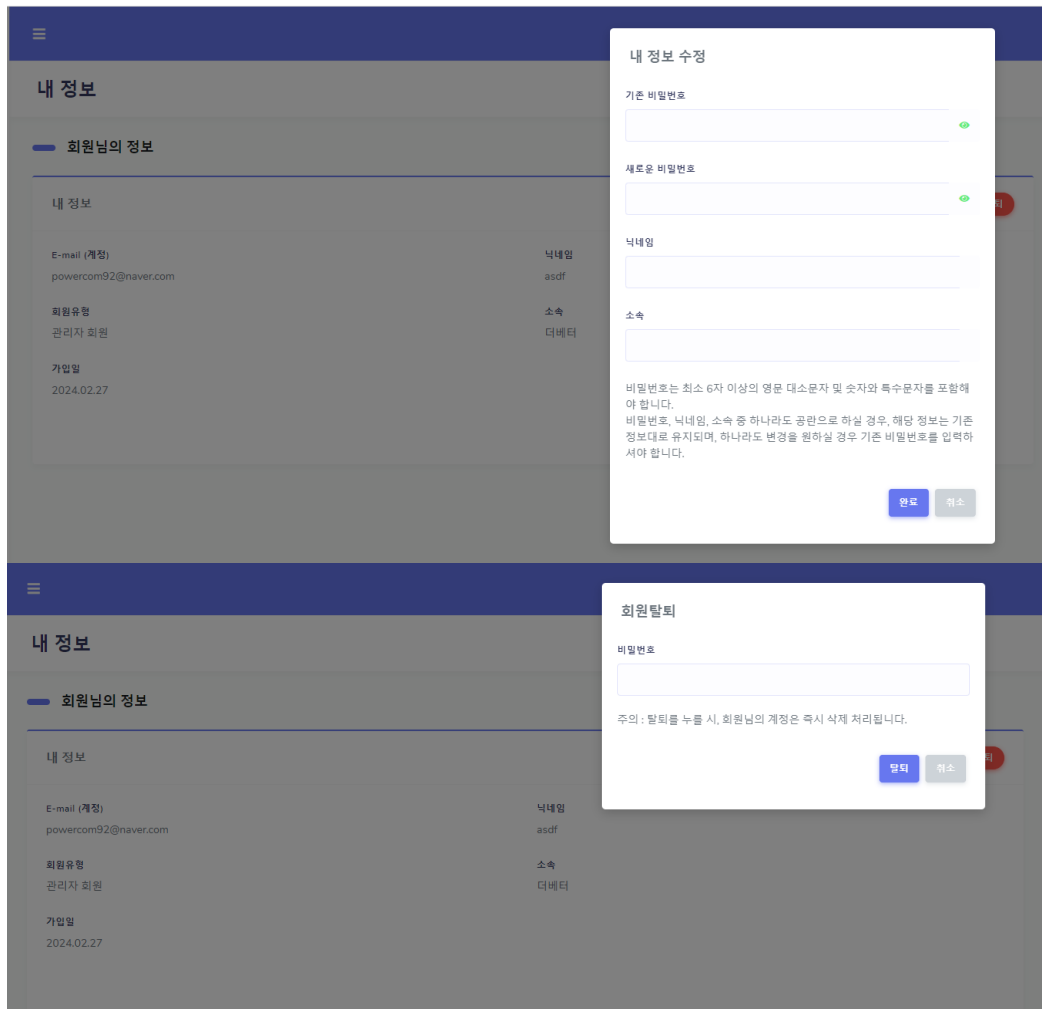
내 정보

회원님의 정보

내 정보 수정

회원탈퇴

내 정보	E-mail (계정)	powercom92@naver.com	닉네임	asdf
회원유형	관리자 회원	소속	더베터	
가입일	2024.02.27			



내 정보 수정 및 회원탈퇴는 모달창을 이용.

```
_id: ObjectId('65c034ffce2266d7e114572c')
email: "thebetter@gmail.com"
password: "$2b$10$/13oxpRt4vS83IXTESAiT.SfN5eefq9q39E68qC4DDv1vGdtmmZDW"
nickname: "jake1"
organization: "thebetter"
role: "NORMAL"
createdAt: 2024-02-05T01:08:13.383+00:00
updatedAt: 2024-02-06T08:50:18.790+00:00
__v: 0
refreshToken: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFPbCI6InRoZWJdHRlckBnbWFPb_"
```

```
_id: ObjectId('65c1da37c578376b8427545c')
email: "thebetter@thebetter.com"
password: "$2b$10$Jue0QG67cB7Eq.yrHtpxAU.wYb5Bh3iKjqYV7epXLUprfsTYF1DD1."
nickname: "Betterhaha"
organization: "The-Better"
role: "NORMAL"
createdAt: 2024-02-06T07:03:14.665+00:00
updatedAt: 2024-02-07T08:12:41.977+00:00
__v: 0
refreshToken: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFPbCI6InRoZWJdHRlckB0aGV1Z_"
```