#1

안녕하세요. 저는 더클링을 만들고 있는 팀 고수익보장의 팀장 황승원입니다.

데이터 분석을 통한 주식거래 플랫폼 더클링, 발표를 시작하겠습니다.

#2

우선 저희 더클링에 대한 소개를 다시한번 하고, 현재 진행 상황과 완료된 부분, 향후 계획에 대해 말씀드리겠습니다.

#3

저희 서비스 이름인 '더클링'은 / 새끼오리 라는 뜻을 가진 영단어입니다. / 작은 오리가 / 물 위에 떠있기 위해 / 보이지 않는 물 속에서 / 오리발로 열심히 헤엄치는 모습이 / 마치, 보이지 않는 곳에서 데이터 분석을 하는 / 우리 서비스의 AI와 비슷하죠.

#4

저희 더클링은 기존에 복잡했던 주식거래 플랫폼을 간단하고 직관적으로 바꿉니다.

또한 AI를 활용한 수익성 있는 매매를 목표로 하고 있고요,

투자자들이 소통 할 수 있는 게시판도 제작할 것입니다.

#5

다음은 진행상황 입니다.

#6

지난 발표때와는 다르게, 저희 서비스의 구체적인 구조를 시각화 해보았습니다.

또한 깃허브 액션을 통해 배포를 더 쉽게 만들었습니다.

우선 가장 우측위를 보시면 저희 팀 개발자들이 각자의 로컬 머신에서 개발을 하게 됩니다.

각자 테스트 서버는 로컬에서 구동하게 되고요,

기능 개발과 테스트가 끝난 뒤 로컬 깃 레포지토리에 커밋을 하게 됩니다.

개발이 끝나면 각자 리모트 레포지토리에 개발 내용을 푸시하고, 풀 리퀘스트를 올리게 되면,

팀장인 제가 코드를 검토하여 main 브랜치에 병합합니다.

그러면 자동으로 깃허브 액션이 푸시를 감지하고 도커 이미지를 빌드합니다.

빌드 된 이미지는 각 aws 서버로 보내져 기존 컨테이너를 지우고 실행되게 되고, 이런 방식으로 업데이트가 이루어지게 됩니다.  
이 서비스 구조도는 깃허브에 올려놓도록 하겠습니다.

#7

다음은 각 파트 별 세부 현황입니다. 프론트, AI, 백엔드 순서로 말씀드리겠습니다.

#8

우선 프론트는 가장 먼저 도커와 깃허브 액션을 도입했습니다.

왼쪽이 배포된 실제 서버 모습이고요, 오른쪽이 로컬에서 돌아가는 서버 모습 입니다.

지금 사파리 개발자도구 아래쪽을 보시면 배포서버는 용량이 1.14 KB 인데 반해,

로컬 서버는 1.73 MB 인것을 보실 수 있습니다. 이는 배포서버에서는 빌드 과정이 포함되기 때문인데요,

#9

왼쪽이 배포, 오른쪽이 로컬 서버입니다.

NodeJS 빌드를 진행한 배포 서버는 용량도 적고 로드 속도도 절반수준으로 빠른것을 볼 수 있습니다.

그러나 개발 시 우리 팀원들이 코드 수정을 하고 깃허브에 올려 풀리퀘스트를 올리고, 병합이 되기까지 시간이 너무 오래 걸립니다.

그래서 각자 로컬 환경에서 동일한 코드로 동작하는 개발 서버를 두었습니다.

이 개발 서버는 코드 파일을 저장하면 실시간으로 타입스크립트 파일이 컴파일 되어 실행됩니다.

이로 인해 개발 편의성을 크게 높였습니다.

#10

다음은 프론트엔드 레이아웃 현황입니다.

저희 웹사이트의 모든 페이지를 목록으로 만들었고, 그에 따라 피그마로 페이지를 그려보았습니다.

#11

다음으로는 AI 입니다.

AI 에서는 주가 데이터를 예측하는 모델을 만들고 있습니다.

그러려면 우선 뉴스 문자열 데이터를 수집하여 인코딩합니다.

저희 팀은 해외 주식의 예측을 목표로 하기때문에 영문 기사를 수집합니다.

그후 시계열 데이터를 받는 모델에 넣고 돌리게됩니다.

#12

뉴스 데이터는 임베딩되어 LSTM에 의해 긍정인지 부정인지 분석되게 되고, 이 긍부정 아웃풋 값은 주가 예측 모델에 인풋값으로 주어지게 됩니다.

#13

실시간으로 받아오는 시계열 데이터를 방금 보았던 감성분석 아웃풋 값과 합쳐서 PROPHET 이라는 모델에 넣고 주가를 예측합니다.

#14

다음은 백엔드 입니다.

우선 보시는것과 같이 마리아 DB를 활용하여 사용자와 각 사용자가 가진 계좌들을 테이블로 정의하였습니다.

#15

이후 스프링 부트 프레임워크에서 회원가입, 세션값을 활용한 로그인을 구현했습니다.

가장 왼쪽부터 로그인 화면, 회원가입 화면, 로그인 하면 보이는 홈 화면 입니다.

DB에 저장된 id를 활용한 세션값은 30분이 지나거나 로그아웃을 하면 파기되어 다시 로그인 화면이 표시되게 됩니다.

각 유저는 여러 계좌를 가질 수 있으며 위에 있는 계좌 추가버튼으로 계좌를 등록 할 수 있습니다. 지금은 모의투자 계좌 하나만 등록해 놓은 모습입니다.

각 계좌별로 토큰 발급, 계좌 조회, 주식 주문 기능을 만들어놓았습니다.

여기서 접근 토큰 발급을 누르면 한국투자증권 OpenAPI 에서 발급받은 키값을 활용하여 API 서버 접근 토큰을 발급받게 됩니다. 이 토큰은 24시간이 지나면 파기되어 재발급을 받아야 합니다.

다음으로 계좌 조회를 누르면

#16

이렇게 국내 주식과 해외주식으로 구분지어 잔고를 보여줍니다.

거래 내역 링크를 누르면 해당 계좌의 거래 내역을 보여줍니다.

지금은 제이슨 로우 데이터를 그대로 보여주지만 프론트엔드에 필요한 데이터만 가공하여 송신 할 계획입니다.

#17

홈 화면에서 이번엔 주식주문 버튼을 눌러보면

각 주식을 종목코드로 조회할 수 있습니다.

국내, 해외 주식으로 구분지어 현재가는 얼마인지, 시가와 종가는 얼마인지, 기업의 재무재표는 어떠한지 등을 가져옵니다.

#18

다음은 종목 조회 밑에 있는 주식 주문입니다.

호가창을 구현하진 않았지만 위에서 얻은 현재가 정보를 활용해

시장가, 지정가 / 매수, 매도 / 국내, 해외 세가지 유형을 입력한 뒤 종목코드와 주문 가격, 수량을 입력하면 실제 주문이 가능합니다.

사진은 삼성전자 세주를 모의투자 계좌에서 시가로 매수 주문을 보내는 모습입니다.

아까 보았던 잔고 화면에 삼성전자 주식이 3주 들어와 있는것을 보실 수 있습니다.

#19

마지막으로 향후 계획을 말씀드리겠습니다.

#20

우선 프론트엔드는 레이아웃 작업이 끝난 상황으로, React 를 활용한 본격적인 개발에 착수 할 예정입니다. 마이페이지와 실시간 주가창 등을 구현 할 예정이고요, 뉴스와 게시판 페이지도 제작합니다. 마지막으로는 껍데기를 백엔드와 연동하여 실 서비스를 구현 할 예정입니다.

#21

AI 파트는 데이터 분석과 전처리가 어느정도 진행 된 상황입니다.

LSTM과 Transfomer 중 더 나은 모델로 감성분석을 진행 할 예정이고요, 시계열 데이터와 합쳐서 데이터를 학습하고 실제 주가를 예측하는 부분이 남아있습니다.

#22

백엔드입니다. 백엔드는 실시간 체결가, 호가창등 아직 구현되지 못한 API 기능을 구현하고,

한투에서 받아온 각종 데이터들을 프론트에 보내주기 위해 API 명세서 작성 등 데이터를 정리할것입니다. 또한 프론트처럼 도커와 깃허브 액션을 적용하여 서버를 배포하고, html 페이지를 모두 지워 백엔드는 프론트만 접근이 가능하도록 aws 보안 설정도 할 계획입니다.

마지막으로는 실제 돈을 실전투자 계좌에 넣어두고 거래 안정성을 검증 할 계획입니다.

#23

다음은 팀 전체 향후계획입니다.

주식거래 서비스는 백엔드에서 어느정도 개발이 끝났습니다.

4월, 5월에 AI 모델과 프론트 개발에 집중하여 실제 웹 페이지를 론칭 할 예정입니다.

#24

이상으로 데이터 분석을 통한 주식거래 플랫폼 더클링을 개발하는 팀 고수익보장이었습니다. 감사합니다.

질문 있으시면 손을 들어주세요.