- ----------!!CSM17 관리자 화면 구현 관련 내용!!----------

작성자:원승현

빌드업 기간:2023/01/~2023/02

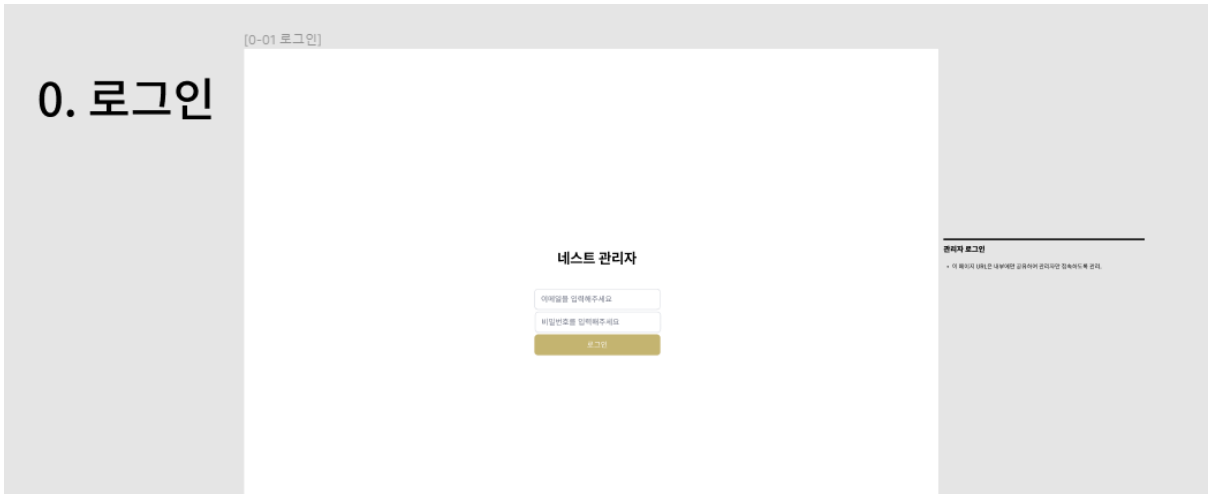
해당 파일은 레이어드 패턴형식으로 구성이 되어있다.

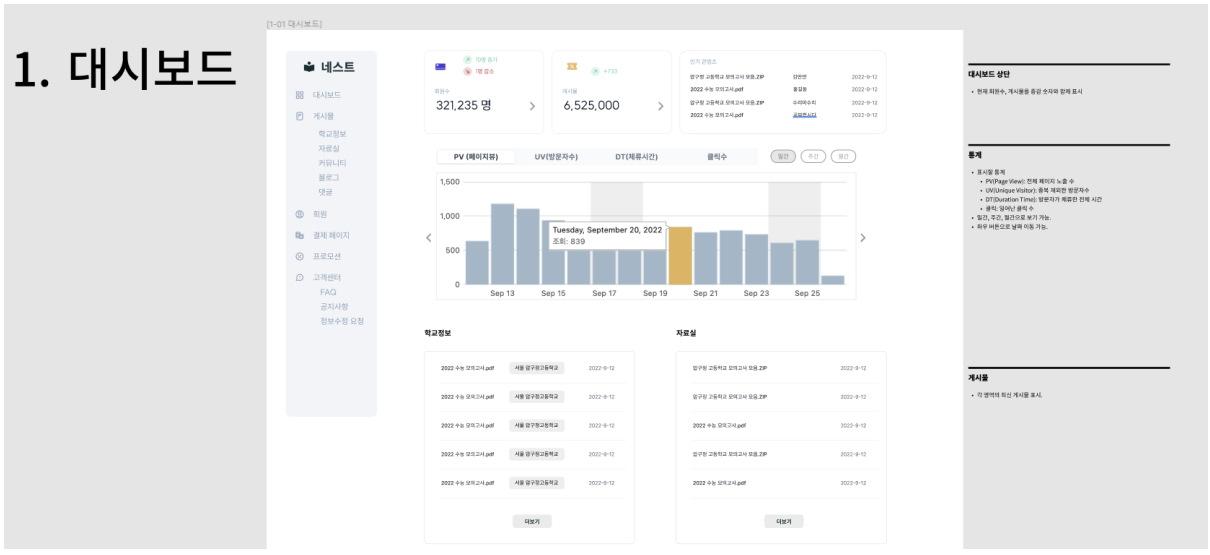
사용 스택은 Js , Nodejs , express ,npm ,mongodb

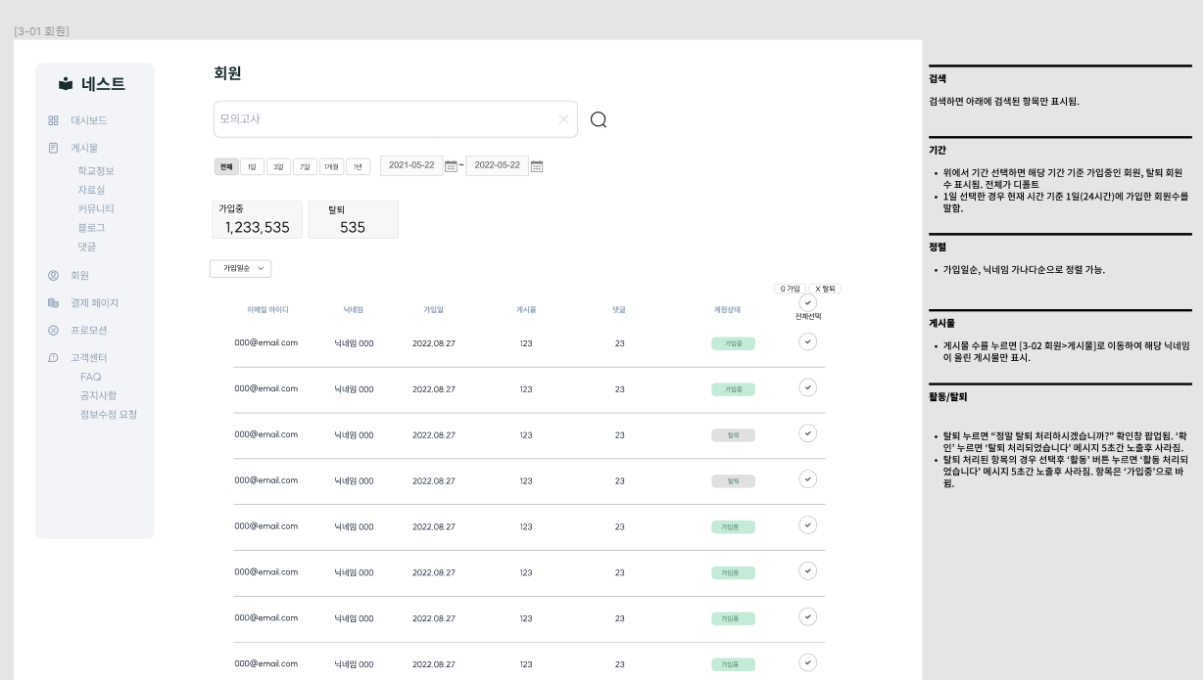
flow는 router - controller - service - model순으로 흘러간다

서버를 동작하는 파일은 app.js ,server.js로 분리 되어있다.

node버전 12.22.12.에 최적화 되어있는 코드이다. node버전이 낮거나 높으면 프레임워크 버전 오류로 서버가 돌아가지 않을수 있다.







기능 구현

처음 기획당시 총 4개의 화면을 담당했지만, 중간중간 다른 화면 기능구현 시도(결제 모듈 , adminJs 도입 시도) 로 인하여 시간이 지체되어 2가지 화면에 대해 구현을 완성했다.

0번째 관리자 로그인 화면이다.

화면에는 없는 기능이지만 엔드포인트 /auth/admin/signin으로 신규 관리자 등록 api도 구현을 시켜둔 상태이다. 가입시 인증이 필요한 adminCode는 .env 파일에서 따로 관리 하고 있다.

테스트로 본인이 사용하는 mongodb에 postman으로 회원가입을 하고 ,

엔드 포인트 /auth/admin/signup으로 화면에 나와있는 관리자 로그인을 시도하면 정상적으로 토큰이 발행되면서 정상 로그인이 되는 것을 확인 할 수 있다.

- 원칙적으로 관리자 암호 또한 hasing을 한 후 db에 저장을 해야 하지만 기능구현이 정상적으로 되는지 확인을 하기위해 hasing을 생략한 상태로 회원가입이 된다. 추후 해당 부분은 비밀번호 hasing 로직을 추가 해주면 된다.

-2023/01/31 관리자 비밀번호 암호화 완료-

1.대시보드 화면

대시보드 화면의 엔드 포인트는 /auth/admin/dashboard

1.왼쪽 상단에 회원수 박스내용

- 왼쪽의 상단 유저의 총 회원수 및 변동 숫자에 대한 값은 userDao에서 db에서 값을 추출해 온다. 테스트 환경이라 직접적인 유저가 없기 때문에 임시적으로 관리자를 다수 생성 후

관리자의 숫자의 변화로 정상적인 기능을 수행하는지 테스트 했다. 추후 해당부분은 관리자 숫자 카운팅에서 유저 숫자를 변화를 나타내도록 관리자에서- 유저로 코드만 살짝 수정해주면 된다. 유저의 수 차이의 기준은 1일전이랑 비교한다. 1일말고 다른 기준으로 수정을 원한다면 62번째 라인에 마지막 1000000부분을 원하는 기준에 따라 수정해주면, 오늘의 총 회원수와 설정한 날짜의 총 회원수의 차이의 변화를 증가 또는 감소로 함께 추출해 준다.

2.중간 상단에 게시물 박스내용

-중간 상단 게시물 총 개수 및 변동 숫자에 대한 값 또한 userDao 에서 db에서 값을 추출 해온다. 이 또한 테스트 환경이라 임의로 db에 게시물을 무작위로 다수 생성하여 진행 하였다. 회원수 증가 감소 모듈 로직이랑 동일하게 비교해야 할 기준은 1일전날이다. 이 기준을 변경하고 싶으면 82번째 라인 1000000을 다른 기준으로 수정 해주면 정상적으로 오늘의 게시물 숫자와 해당 기준의 날짜의 게시물 숫자를 증가또는 감소로 수치로 나타내서 뽑아준다.

3.오른쪽 상단의 인기 컨텐츠 박스 내용

-인기 컨텐츠의 기준은 조회수로 잡았다. 물론 댓글 수 , 좋아요 수로 변경을 원한다면 변경 또한 가능하다. 테스트 환경이기 때문에 유저의 실질적은 조회수 카운팅이 없기 때문에 각 게시물이 생성될때 랜덤적으로 조회수 값을 갖도록 랜덤 함수를 만든 후 조회수를 비교하여 내림차순으로 상위 5개를 추출하게 해준다.

현재 기준은 내림차순이지만 오름차순으로 변경을 원한다면 ,95번째 라인의 view:-1의 부분을 view:1로 수정해주면 오름차순으로 추출이 가능하다. 또한 5개 보다 더 많은 데이터를 보고 싶으면 95번째 라인 limit(5)부분을 원하는 게시물 숫자에 따라 limit(원하는 숫자)로 변경해주면 된다.

조회수 기준이 아닌 예를들어 좋아요 순 또는 댓글순으로 정렬을 하고 싶으면 95번째 라인 view:-1의 부분을 like:-1 또는 review:-1로 수정을 해주면 된다.

4. 왼쪽 하단 학교정보

-왼쪽 하단 학교정보 박스의 출력 내용을 보면 파일 , 학교명 , 생성날짜 가 추출된다 하지만 기준이 없기 때문에 생성순서 (즉 id값 기준) 으로 5개 추출된다 102번째 라인에서 수정 가능하다

5.자료실

이 부분에서 고민을 많이 하였다. 시험관련 pdf파일이다 보니 , 몽고디비에서 저장하기엔 너무 파일이 헤비 하다고 판단하였다. 따라서 구글 드라이브와 연동하여 , 업로드 시 자동으로 구글 드라이브로 업로드가 되어, 실질적으로 백 부분에서 파일 원본을 db에 저장하여 트래픽 낭비를 발생 시키는거보다

해당 자료에 대한 구글 드라이브 Url만 mongodb에 저장이 되도록 하여 트래픽 발생을 줄이는 방향으로 짰다.

108번째 라인에서 수정이 가능하다

2.회원 관리 화면

검색창에 찾고 싶은 유저의 닉네임을 입력시 해당 글자를 포함하는 닉네임을 가지고 있는 모든 유저를 불러온다. 쿼리 파라미터로 작성되어있으며 앤드포인트는

/auth/admin/userstatus?userNickName=“검색하고싶은거”

해당 유저를 검색이 된다.

또한 밑에 현재 가입중인 유저의 전체 수 , 탈퇴한 유저의 수 또한 출력이 된다.

유저의 상태를 변경 하는 기능의 앤드 포인트는

/admin/userstatus/changestatus/:userId/:statusId이다

유저를 영구탈퇴 및 삭제시키는 기능은

엔드포인트 "/admin/userstatus/delete/:userId 이다

- ETC:

해당 로직의 db의 주소는 .env에서 관리하기 때문에 코드상에는 나와있지 않다. 정상적으로 구동이 가능한지 테스트 하기 위해선 작성자의 디비로 테스트 하는게 편하다. 테스트를 해보려 db가 필요하다면 작성자 연락처 010-7604-0858로 연락주시면 db에 접근을 할수 있다.

나머지 화면 구현 또한 시간만 충분했다면 가능했지만 마무리를 깔끔하게 하지 못해 아쉬움이 많은 프로젝트 였다. 마지막으로 화면 중간에 그래프는 프론트딴에서 구현하는 grahp.js로 구현이 가능하기 때문에 생략하고 넘어갔다.