1 페이지

안녕하십니까. COCONET 팀의 발표자 김현빈입니다.

저희 팀은 박경수 교수님께서 지도해 주셨으며,

팀원은 발표 및 프론트 엔드 개발 담당의 김현빈,

백 엔드 개발 담당의 정재훈,

백 엔드 개발 및 문서 정리, 발표 자료 제작을 맡은 김은비

총 3명으로 이루어져 있습니다.

2 페이지

먼저 발표는 프로젝트 개요와 동기 설명을 시작으로,

사용한 기술 스택과 프로젝트 구성도, 그리고 실제화면을 보며 주요 기능 여섯 가지에 대해 설명을 드리고, 마지막에는 향후 계획 설명 시간을 갖도록 하겠습니다.

3 페이지

첫 번째로, 저희 프로젝트를 간단히 소개해 드리도록 하겠습니다.

4 페이지

COCONET은 실제 기업에서 운영되고 있는 ●

사내 인트라넷에 보안을 접목해보면 어떨까 하는 생각에 시작되어 개발하게 된

사내 보안 인트라넷입니다.

5 페이지

잠시 실제 화면을 가져와 프리뷰 해드리자면,

보시는 바와 같이 출퇴근 현황 관리, 근태 기록과 같은 기본적인 인사 서비스부터, ●

기업의 기밀 유출 방지를 위한 모바일 단말 관리 서비스를

모두 하나의 인트라넷에서 관리할 수 있습니다.

6 페이지

다음으로, 저희 팀이 수많은 서비스들 중에 어째서 인트라넷을 선택하였는가에 대해서

이야기해보겠습니다.

주된 이유는 다음의 두 가지입니다.

첫 번째로, 원격 근무 문화가 한국에서도 널리 확산되면서,

중요 정보나 기술이 유출될 수 있다는 우려가 증가하고 있다는 것입니다. ●

그러나, 막상 이를 보호할 수 있는 정책인 BYOD를 도입하는 것은

불 필요하거나 비용이 효율적이지 않다고 여기는 기업이 다수입니다. ●

저희는, 이러한 문제점들을 페이지에 접근할 수 있는 자격을 제한하는 Authorization과,

등록된 기기만 사용할 수 있도록 하는 BYOD 정책,

그리고 모바일 장치의 기능을 제한하는 Mobile Device Management 서비스를

하나의 인트라넷 시스템에 통합하는 것으로 해결할 수 있다는 판단이 들어

사내 보안 인트라넷을 제작하게 되었습니다.

7 페이지

다음으로, 실제로 구현된 프로젝트의 내용에 대해서 살펴보겠습니다.

8 페이지

다음 화면은 이번 프로젝트에서 사용한 기술 스택을 정리한 것입니다.

우선 맨 왼쪽의 웹 클라이언트에서는 vs-code, html, css, java script,

그리고 React를 주로 이용하였으며, 상태관리를 위해 Redux, 빌드를 위해 Vite를 사용하였습니다.

웹에서 서버로 API 요청을 할 때에는 JWT와 Axios를 사용합니다.

서버는 웹에서 보낸 API 요청을 받아 처리하며,

이때 서버에서는 IntelliJ, Java, Spring Framework와 Spring Security, JPA Hibernate,

Rest Api, Lombok, coolSMS와과 같은 기술을 사용합니다.

추가로, 로깅 관련 라이브러리로는 보안 이슈가 있는 Log4j 대신 Logback을 사용하였으며,

서버에서 Repository를 통해 DB에 접근할 때에는 다시 Spring Security와 Bcrypt가 사용되고,

여기서 데이터베이스는 H2를 사용합니다. ●

추가로, SCM 형상 관리 프로그램으로는 git을,

API 문서를 정리하는 것은 Notion을 사용하였습니다.

9 페이지

해당 화면은 저희 프로젝트의 구성도입니다. ●

전체적으로는 화면과 같이 구성되어 있으며,

다음 슬라이드에서 확대하여 자세히 설명 드리도록 하겠습니다.

10 페이지

우선 맨 왼쪽에 React로 제작된 웹 클라이언트와 중앙에 Spring Boot로 제작된 서버,

그리고 보시는 화면에는 보이지 않지만 오른쪽에 데이터베이스가 존재합니다.

JWT와 Spring Secuirty, Logback의 경우 이후 이어질 기능 소개 슬라이드에서

자세히 다룰 예정이므로, 해당 화면 에서는 웹과 서버, DB의 통신 과정을 메인으로 설명 드리겠습니다. ●

우선, 웹 클라이언트가 브라우저를 통해 서버로 Http Request로 API 요청을 보냅니다. ●

이때, SSL을 적용하여 통신을 암호화하며 ●

이후, Spring Security Authentication을 통해 사용자의 신원과 권한을 증명하게 됩니다. ●

해당 요청이 올바른 요청일 경우 Dispatcher Servlet이 요청에 해당하는 컨트롤러를 찾아

데이터를 처리합니다.

서버의 디자인패턴은 Spring MVC 패턴으로 이루어져 있습니다. ●

Controller에서 요청을 받게 되면, ●

Service에서 Repository를 통해 DB에서 해당하는 Entity의 데이터를 가져온 후

반환 받은 값을 다시 Controller에게 전달하고 ●

Controller는 원하는 데이터를 유저에게 돌려주게 됩니다.

11 페이지

다음으로 데이터베이스의 구성을 나타내는 ERD입니다.

회원정보 테이블을 중심으로, 릴레이션 된 테이블은

토큰 정보, 상태 정보, 권한 정보를 저장하는 테이블입니다.

그 외에는, 통신에 필요한 데이터들을 저장하는 테이블들로 구성이 되어있습니다.

12 페이지

COCONET의 회원가입은 사용자가 아닌 관리자 페이지에서 작업됩니다.

회사 조직에 신규 사원이 입사하였을 경우,

관리자 페이지의 신규 회원 추가 메뉴에서 사용자를 추가할 수 있습니다.

빈칸 없이 데이터를 입력한 뒤 ●

완료 버튼을 클릭하면 ●

서버로 데이터가 전송되며, ●

이때 클라이언트와 서버 간의 통신은 SSL로 암호화됩니다. ●

전송된 데이터가 올바를 경우 ●

DB에 사용자 권한을 가진 레코드가 추가됩니다. ●

여기서 비밀번호 암호화를 위해 해시를 사용하였는데, 단순하게 해싱만 수행했을 경우 발생될 수 있는 무차별 대입 공격 및 Rainbow table 공격을 방지하기 위해

회사에서 지정해둔 임시 비밀번호에 Salt를 더한 값이 해시로 암호화되어 저장됩니다.

DB에 레코드가 정상적으로 저장되면, ●

가입시 입력한 사용자의 전화로

회원가입 완료 알림과 임시 비밀번호가 문자로 전송됩니다.

13 페이지

다음은 로그인 로직입니다.

로그인 세션 유지 방식으로는 Token Based를 채용했으며, 그 중에서도 JWT를 사용하였습니다.

로그인 폼에 존재하는 사용자의 데이터를 입력한 후 로그인 버튼을 클릭하면 ●

서버로 데이터가 전송되고, ●

DB에 저장된 사용자의 상태가 '출근' 상태로 변경됩니다.

또한 사용자의 로그인 성공 로그가 남게 되는데, 이는 관리자 페이지에서 사용자 로그 조회를 통해 확인이 가능합니다.

마지막으로 Spring Security를 통해 획득한 정보를 기반으로

JWT 토큰이 발급되며, 이 토큰이 유효기간 만료 전에는 계속해서 세션이 유지되게 됩니다.

자세한 과정은 다음 슬라이드에서 설명 드리겠습니다.

14 페이지

여기서, 로그인에 사용한 JWT는, ●

다음과 같은 구조로 이루어져 있습니다.

먼저, 토큰의 타입과 해시 알고리즘을 지정할 수 있는 헤더와

토큰에 담을 정보를 저장하는 내용,

그리고 유효성 검증을 할 때 사용하는 서명,

세 부분으로 나누어져 있으며, ●

COCONET에서 발급되는 토큰 또한 이와 같은 구조로 이루어져 있는 것을 확인하실 수 있습니다.

우측 화면에서 볼 수 있는 것처럼 COCONET Access Token은 HMAC using SHA-512으로 암호화 되어있으며, 페이로드에는 사용자의 이메일, 권한, 만료일을 담아두었습니다.

하지만, 이 Access Token만으로 인증을 수행하기에는 ●

제3자에게 탈취당할 경우에 해당 토큰만으로 서비스에 접근이 가능하기 때문에 보안에 취약합니다. 그래서 도입하게 된 것이 Refresh Token입니다.

15 페이지

Access Token과 함께 발급되는 Refresh Token은 이전과 같은 형태의 JWT로,

페이로드에는 개인정보가 아닌, 유효기간이라는 무의미한 값이 들어 있습니다.

이 토큰은 인증에 사용되지 않으며, 오직 Access Token이 만료되었을 경우 Token을 새로 발급해 주는 용도로만 사용됩니다. 이때, Refresh Token의 유효기간은 2주 정도로 길게 설정하고,

Access Token의 유효기간을 1시간 정도로 설정했다고 가정하면

탈취를 당하더라도 짧은 유효기간을 가진 Access Token은 이미 사용할 수 없는 상태가 되어

비교적 안전하다고 할 수 있습니다.

16 페이지

이러한 구조를 가진 JWT는 다음과 같은 동작 원리를 기반으로 작동합니다. ●

먼저, 웹 클라이언트가 Axios 요청을 통해 서버로 로그인 요청을 보내면, ●

서버는 요청을 받아 해당 사용자의 유저 정보, 권한, 만료일자로 Access Token을 생성하고. ●

만료일자로만 생성된 Refresh Token은 DB에 저장합니다. ●

그 뒤, 응답으로 생성된 Access Token과 Refresh Token을 클라이언트에 알려주면 ●

클라이언트는 발급받은 토큰 중 Refresh 토큰을 session storage에 저장 ●

유저 정보는 redux에 저장하여 상태를 관리합니다. ●

이후 API 요청 시, Request Header에 Access Token 정보를 담아 요청하면 ●

서버에서 해당 토큰을 validation하여 ●

유효한 토큰일 경우에만 응답을 돌려줍니다.

Access 토큰을 새로 발급받아야 하거나 기간이 만료된 경우 Session Storage에 저장된 Refresh 토큰을 이용하여 재 발급받는 것으로 세션을 유지하며 Refresh 토큰의 유효성 검증에 실패할 경우 로그아웃 처리됩니다.

17 페이지

회원가입과 로그인 기능에 적용 되어있는

Spring Security는 우측과 같은 구조를 가지고 있는 프레임 워크입니다.

Spring Security는 크게 Authentication과 Authorization의 구현에 도움을 주고 있습니다.

Authentication의 경우, coconet의 모든 페이지에 접근할 때

인증이 되지 않은 사용자가 보낸 요청에 대해서 접근을 통제합니다.

Authorization의 경우, 회원가입 시 사용자의 계정에 적절한 권한을 부여하며

이후 사용자가 권한이 필요한 특정 페이지, 혹은 리소스에 접근을 하면

접근할 수 있는 권한을 소유하고 있는지 판단합니다.

18 페이지

해당 화면은 로그인 페이지에서 이동할 수 있는 비밀번호 초기화 화면입니다.

비밀번호를 잊어버렸을 경우, 해당 페이지에서 존재하는 사용자의 정보를 입력하고 ●

코드 발급을 이벤트를 발생시키면, ●

해당 사용자의 휴대폰으로 인증번호 관련 메시지가 전송됩니다. ●

인증번호를 올바르게 입력하고, 인증하기 버튼을 클릭하면, ●

이어지는 비밀번호 재설정 페이지에서

8글자 이상이라는 규칙을 갖고 있는 비밀번호를 새로 설정할 수 있습니다.

19 페이지

다음은 구현된 페이지 내용을 살펴보도록 하겠습니다.

먼저, 사용자 페이지입니다. ●

COCONET에 사용자 계정으로 로그인 시 처음 나타나게 되는 메인 화면입니다. ●

좌측에 보이는 '오늘의 업무'는 Todo List 기능으로,

할 일 추가 및 삭제, 완료 처리가 가능합니다. ●

그 옆의 공지사항에서는 관리자가 작성한 공지사항을 확인할 수 있으며, ●

더보기를 클릭하면 연도별로 게시글의 내용을 조회할 수 있습니다.

다시, 메인 화면으로 돌아오면, ●

우측 부분에서 알림을 확인할 수 있습니다. ●

상단의 업무 시작과 직원 출근 퍼센트 관련 로그는

관리자가 지정한 출근 시간에 자동으로 생성되는 로그이며, ●

그 외 로그는 사용자 로그 중 출퇴근 관련 로그를

최신순으로 읽어와 보여주게 됩니다.

20 페이지

다음은, 근무 현황 페이지입니다. ●

이곳에서는 차트 하나를 확인할 수 있는데,

해당 차트는 출근시간 기준, 사원들의 현재 근무 현황 통계를 확인할 수 있게끔

시각적으로 나타낸 것입니다. ●

결재는 사원들의 휴가, 외근, 출장과 관련된 결재 처리 사항을 확인할 수 있는 메뉴로,

서류가 처리된 날짜를 확인할 수 있습니다. ●

출퇴근 현황에서는 각 부서별 사용자들의 출퇴근 관련 로그, 즉 시간을 확인할 수 있습니다.

21 페이지

이번엔 기기 관리 페이지입니다. ●

사용자가 보유한 모바일 디바이스, 기기의 어떤 기능이 차단되어 있는지

확인할 수 있는 페이지로, ●

기능 제어 관련 로그를 우측의 알림 부분에 가져와

기기별로 기능이 차단, 허가된 시간을 확인할 수 있도록 하였습니다.

22 페이지

인트라넷 화면의 우측 하단 아이콘을 클릭하여 진입할 수 있는

사용자 정보 수정 페이지입니다.

기본 정보 메뉴에서는 ●

이름과 전화번호를 변경할 수 있으며,

변경 시 사용자 로그에 관련 로그가 남게 됩니다. ●

조직 정보 메뉴에서는 현재 내가 소속된 부서와 직급 정보를 확인할 수 있습니다.

추후, 이 부분에 부서별 계급도를 추가하겠다는 계획이 있습니다. ●

비밀번호 변경 메뉴에서는 사용자의 비밀번호를 변경할 수 있습니다.

먼저, 현재 비밀번호를 입력한 후 ●

일치 여부를 확인하는데, ●

이때 Bcrypt를 통해 '입력한 비밀번호의 해시 값과 – 기존 비밀번호의 해시 값' 일치 여부를 확인하고, 동일하게 입력했을 시에만 8자리 이상의 새로운 비밀번호로 변경할 수 있게 합니다.

23 페이지

이제, 관리자 권한을 가진 계정으로 로그인 된 사용자에게만 보여지는

관리자 전용 페이지를 살펴보도록 하겠습니다. ●

먼저, 출퇴근 시간 변경 메뉴입니다.

관리자가 근무일과 출퇴근시간을 설정하면 ●

변경된 내용이 CRON 표현식에 맞춰 서버의 스케줄러에 등록되며, ●

스케줄러는 해당 시간에 출근한 사용자의 통계와, 업무 시작을 알리는 로그를 남깁니다.

24 페이지

다음으로, 디바이스 제어 페이지입니다.

관리자가 기기를 제어할 사용자를 선택하면 ●

해당 사용자가 보유한 디바이스의 기기명과

현재 차단된 기능의 목록을 확인할 수 있습니다.

여기서 제어할 기능을 선택하고 적용 버튼을 클릭하면 ●

제어된 내용이 로그로 남게 되며,

사용자는 해당 디바이스에서 더 이상 차단된 기능을 사용할 수 없게 됩니다.

25 페이지

사용자 로그 조회와 관리자 로그 조회 페이지를 보여 드리기 전에,

COCONET은 어째서 로그를 남기는가, 어떻게 관리하는가 이야기 드리고자 합니다.

COCONET에는 크게 ●

사용자 로그와 관리자 로그가 존재합니다.

먼저, 사용자 로그를 남기는 이유에 대해서 설명 드리겠습니다. ●

사용자 로그를 남기는 이유로는 다음의 세 가지를 들 수 있습니다.

첫 번째, 사용자 계정에서 발생하는 부적절한 작업을 감시하여, 사용자에게 위험을 알리기 위해,

두 번째, 계정 해킹 등의 문제가 발생할 경우 해당 원인을 탐색하여 같은 일의 반복을 방지하기 위해,

마지막으로, 근무 관리 페이지에 있는 차트의 데이터로 사용하기 위해서 로그를 남기게 됩니다.

이러한 로그는 외부 API를 통해서 DB에 접근해 CRUD 이벤트를 발생시키거나

로그인, 로그아웃 시에 남게 됩니다. ●

다음으로, 관리자 로그를 남기는 이유와 시점에 대해서 설명 드리겠습니다.

관리자 로그를 남기는 데에는 다음의 세 가지 이유가 존재합니다.

첫 번째, 관리자가 수행하지 않은 작업이 발생했을 때, 그를 인지하고, 대처하기 위해.

두 번째, 로그 파일을 참고하여 시스템을 유연하게 유지보수 하기 위해,

마지막으로, 개인정보 관련 법 조항을 준수하기 위해 관리자 로그를 남기고 있습니다.

관리자 로그는 관리자가 사용자 정보에 접근하거나, 정보를 변경할 때,

출퇴근 시간 변경과 같이 시스템에 관련된 내용을 변경할 때,

관리자가 사용자 기기의 기능을 제어할 때 남게 됩니다. ●

로그에 남게 되는 내용은 개인정보보호법의 접속 기록의 보관 및 점검에 관한 제 2조에 따라

관리자의 계정과, 접속일시, 접속지 IP, 처리한 정보주체 정보, 그리고 수행업무를 로그로 남기게 됩니다. ●

보관 및 삭제 시점은 정보통신망법 접속기록의 위 · 변조 방지에 관해 제5조에 따라

로그는 별도의 물리 장치에 저장하여 보관하도록 하였으며, ●

정보통신망법 제5조와 개인정보 보호법 개인정보의 안전성 확보 조치 기준 제8조에 따라

로그 파일의 만료 기간을 1년으로 설정하였고,

하루 단위로 파일이 백업되도록 롤링 정책을 설정하였습니다.

26 페이지

이렇게 남게 된 로그들은 각 메뉴에서 조회할 수 있게 됩니다.

먼저, 사용자 로그 메뉴에서는 ●

상단에서 조회할 사용자를 선택하여

해당 사용자의 계정에 관련된 로그 내용과 시간을 확인할 수 있습니다.

27 페이지

다음으로, 관리자 로그 조회 화면입니다. ●

이 페이지에서는 화면과 같이 관리자 활동에 관련된 로그들을 확인할 수 있습니다. ● (2회 클릭)

28 페이지

다음은 시연 영상을 보여드리겠습니다.

29 페이지

마지막으로 COCONET의 향후 계획에 대해 말씀드리겠습니다.

30 페이지

우선, 수집하고 있는 로그를 활용하여

~~~~~~~~~~~~..... 수 있는 방안을 마련하고 싶습니다.

그리고, 안드로이드 플랫폼의 경우 안드로이드 차단기능 구현 과정에서 문제가 발생하여

이번 발표에서 다루지 못하게 된 점이 아쉽습니다.

따라서, 발표회 이후에 추가~~~~. ◇

31 페이지

이상으로 발표를 마치도록 하겠습니다 coconet이였습니다. 감사합니다.
~~~~~~~~~~~~