클라우드 기반의 문서 관리 및 전자결재 시스템

팀 명 : Cloud Office

지 도 교 수 : 이병천 교수님

팀 장 : 최준영

팀 원 : 홍승현

류재원

박범주

강형욱

지영욱

2023. 11.

중부대학교 정보보호학과

목 차

1. 서 론
   1.1 연구배경 ····· 3
   1.2 연구필요성 ····· 3
   1.3 연구 목적 및 주제선정 ····· 3

2. 관련연구
   2.1 Python ····· 4
   2.2 Django ····· 4
   2.3 HTML ····· 4
   2.4 CSS ····· 4

3. 본 론
   3.1 개요 ····· 5
   3.2 동작 설계 ····· 5
   3.3 DB 설계 ····· 6
   3.4 메뉴 소개 ····· 6

4. 분 석
   4.1 활용 시나리오와 사례 제시, 효과 분석 ····· 10
   4.2 흐름도 ····· 10

5. 결 론
   5.1 결론 및 기대효과 ····· 11
   5.2 향후 과제 ····· 11

6. 별첨
   6.1 참고 문헌 ····· 11
   6.2 주소 ····· 11
   6.3 팀 소개 ····· 12
   6.4 발표자료 ····· 12

1. 서론

1.1 연구 배경

지속 가능한 환경 및 현대 비즈니스 환경의 변화에 대한 요구가 증가함에 따라, 웹 기반의 사무보조 시스템 웹사이트 개발은 중요한 주제가 되었다. 이러한 시스템은 다양한 이유로 필요성이 대두되고 있으며, 이에 대한 연구와 개발이 중요한 과제로 부상하고 있다. 환경 보호와 지속 가능한 자원 활용은 급속한 산업화와 도시화로 인해 중요성을 갖는 문제 중 하나다. 종이 문서 관리는 대량의 자원과 에너지를 소비하며, 폐기물로 이어지는 환경 오염을 초래한다. 웹 기반의 사무보조 시스템 웹사이트를 통해 종이 문서를 전자 문서로 대체함으로써, 환경 오염을 줄이고 지속 가능한 환경을 위한 노력을 지원한다.

1.2 연구 필요성

디지털 시대에는 비즈니스 환경이 급속하게 변화하고 있다. 기업은 민첩성을 유지하고 경쟁력을 확보하기 위해 신속한 의사결정과 업무 프로세스 최적화가 필요하다. 웹 기반의 사무보조 시스템 웹사이트는 업무 프로세스를 자동화하고 문서 공유를 용이하게 함으로써 기업이 이러한 빠른 환경 변화에 대응할 수 있도록 돕는 중요한 역할을 한다.

또한 기업 및 조직에서는 중요한 비즈니스 정보를 안전하게 보호해야 한다. 종이 문서를 관리할 때 발생할 수 있는 문서 유출 및 정보 누설 위험을 줄이기 위해 웹 기반의 사무보조 시스템 웹사이트는 내부망에 설치된 시스템을 통해 모든 문서를 통합하고 보안성을 강화한다.

이로써 중요한 비증서성 정보를 안전하게 관리할 수 있으며, 규정 및 규정 준수를 준수할 수 있다. 전자 문서 관리 시스템은 문서에 빠르게 액세스할 수 있는 편리한 방법을 제공한다. 종이 문서 관리 시스템은 물리적 위치로 인해 문서에 접근하기 어려운 경우가 많지만, 웹 기반의 사무보조 시스템 웹사이트를 사용하면 언제 어디서든 문서에 접근할 수 있어 업무 효율성을 크게 향상시킬 수 있다.

1.3 연구 목적 및 주제 선정

본 프로젝트의 목적은 디지털화된 전자 문서들을 데이터베이스에 안전하게 저장하고, 다양한 이용자 그룹과 직책에 따라 신속하게 문서에 접근할 수 있는 신뢰성 높은 단일 시스템을 구축하는 것이다. 이를 통해 문서관리 및 정보 공유의 효율성을 향상시키고, 비즈니스 프로세스의 최적화를 실현하는 것이 목표다.

구체적으로, 웹 기반의 클라이언트와 안정적인 문서 저장 및 권한 관리 서버를 구축한다. 이를 통해 이용자는 어디서나 웹 브라우저를 통해 문서에 접근하고, 필요한 정보를 즉시 얻을 수 있다.

2. 관련 연구

2.1 Python

Phython은 웹 애플리케이션, 소프트웨어 개발, 데이터 과학, 기계 학습(ML)에 널리 사용되는 프로그래밍 언어이다. Python이 효율적이고 배우기 쉬운 것이 장점으로, 여러 플랫폼에서 실행될 수 있기 때문에 널리 Python을 사용한다. Phython 소프트웨어는 무료로 다운로드할 수 있고, Windows, macOS, Linux 및 Unix와 같은 다양한 컴퓨터 운영 체제 유형의 시스템과 원활하게 통합된다. 그렇기 때문에 개발자 커뮤니티에서 다양한 정보를 손쉽게 접할 수 있고, 문제 해결도 빠르게 받을 수 있다.

2.2 Django

장고 파이썬으로 2005년부터 시작된 Python의 오픈 소스 웹 프레임워크이고, 모델-템플릿-뷰(MTV) 패턴을 따르고 있다. 현재는 장고 소프트웨어 재단에 의해 관리되고 있다. 고도의 데이터베이스 기반 웹사이트를 작성하는 데 있어서 수고를 더는 것이 장고의 주된 목표이다. 그렇기 때문에 Python 기반 웹 프레임워크 중에서 가장 널리 퍼져 있기 때문에, 개발자가 손쉽게 원하는 정보를 찾을 수 있다는 장점이 있다. 장고는 컴포넌트의 재사용성과 플러그인화 가능성, 빠른 개발 등을 강조하고 있다. 또한, 중복 배제 원리를 따르고 있다.

2.3 HTML

HTML은 Hyper Text Markup Language의 약자로, 웹 페이지를 표시를 위해 개발된 마크업 언어이다. 인터넷 서비스의 하나인 WWW(World Wide Web)을 통해 볼 수 있는 문서를 만들 때 사용하는 기본적인 웹 언어의 한 종류를 말한다. HTML은 하이퍼텍스트를 작성하기 위해 개발되었고, 현재 인터넷에서 웹을 통해 접근되는 대부분의 웹 페이지들은 HTML로 작성된다.

HTML은 문서의 글자 크기, 글자색, 글자 모양, 그래픽, 문서이동(하이퍼링크) 등을 정의하는 명령어로서 홈페이지를 작성하는 데 쓰인다. HTML에서 사용하는 명령어는 태그(tag)라고 하는데 꺽쇠괄호(<>)를 사용하여 나타낸다. 이렇게 HTML로 작성된 문서를 웹 브라우저가 해석하여 이용자에게 보여주게 된다. HTML은 서버로부터 받은 정보대로 페이지를 구현하는 것에는 강하지만, 사용자의 입력에 반응해 페이지를 만드는 것에는 약하기 때문에 JavasScript 같은 각종 스크립트의 도움을 받아 이와 같은 약점을 보완하기도 한다.

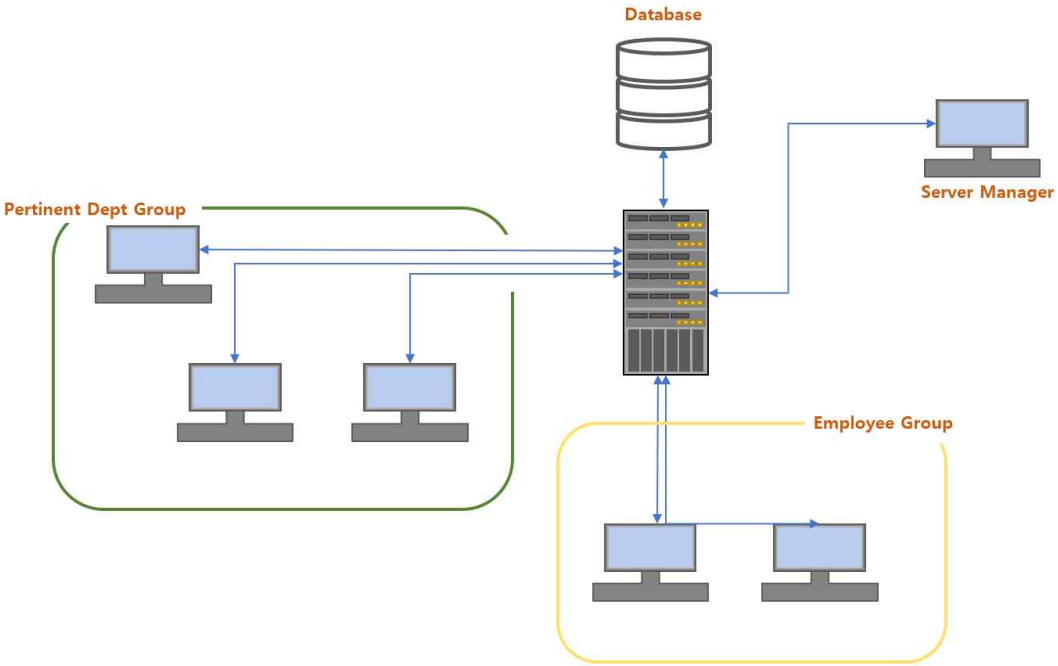
2.4 CSS

CSS는 HTML과 함께 웹을 구성하는 프로그래밍 요소이다. HTML은 틀을 구성하는 것이라면, CSS는 디자인 요소를 담당한다고 말할 수 있다. 그렇기에 HTML은 표와 텍스트, 이미지 같은 부분을 담당하기 때문에 이를 보완하기 위해 만들어진 것이 스타일 시트이고 스타일 시트의 표준안이 바로 CSS(Cascading Style Sheet)이다. CSS는 HTML로부터 디자인적인 요소를 분리해 사용하고, 세세한 스타일 지정의 필요를 줄어들게 하였다. HTML과 XHTML에 주로 쓰이며, 웹 문서의 내용과 상관없이 디자인만 바꾸거나

디자인은 그대로 두고 웹 문서의 내용 변경을 용이하게 해준다. 또 PC나 스마트폰, 태블릿 등 기기환경에 맞게 콘텐츠를 바꿀 수 있는 반응형 디자인이 가능하다.

3. 본론

3.1 시스템 개요



[그림 1. Cloud Office 시스템 개요]

3.2 동작 설계

3.2.1 회원 가입

django의 로그인 인증 시스템을 사용해서, 아무나 쉽게 가입할 수 없고 관리자 계정을 통해서 계정의 승인을 받아야만 사이트를 이용할 수 있다.

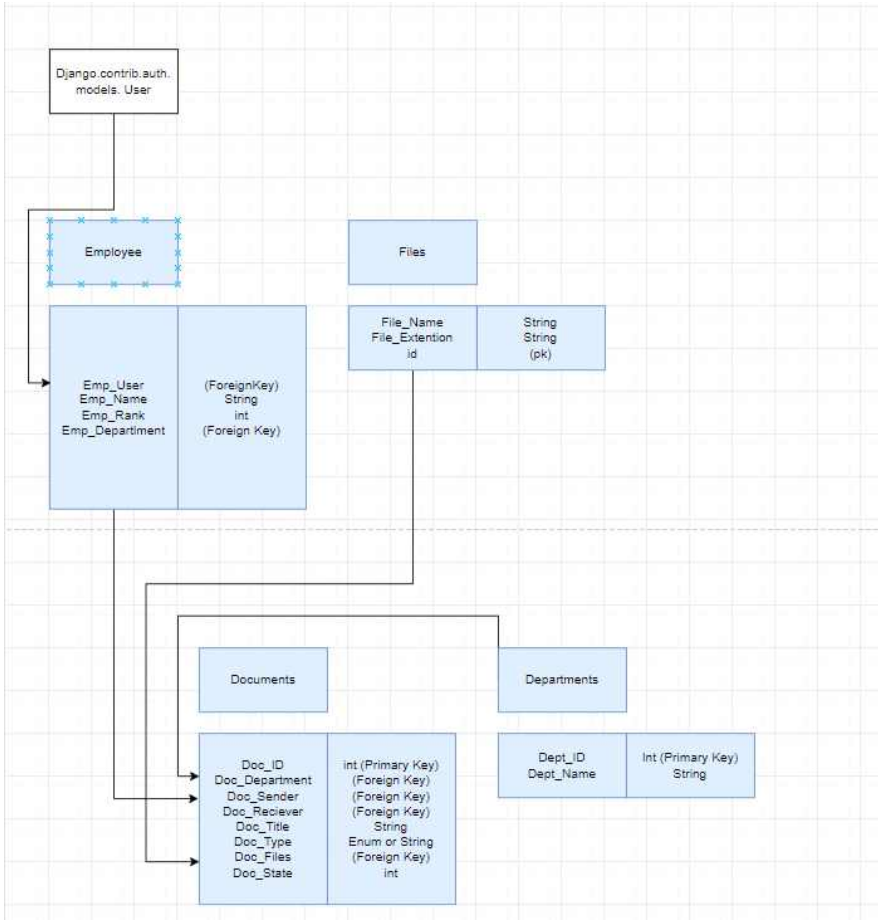
3.2.2 메인 페이지

메인 페이지에서는 자신이 받은 문서와 받은 메일, 처리 대기 중인 문서를 확인할 수 있고 배너를 통해 각각의 페이지를 자유롭게 오고 갈 수 있다.

3.2.3 문서 시스템

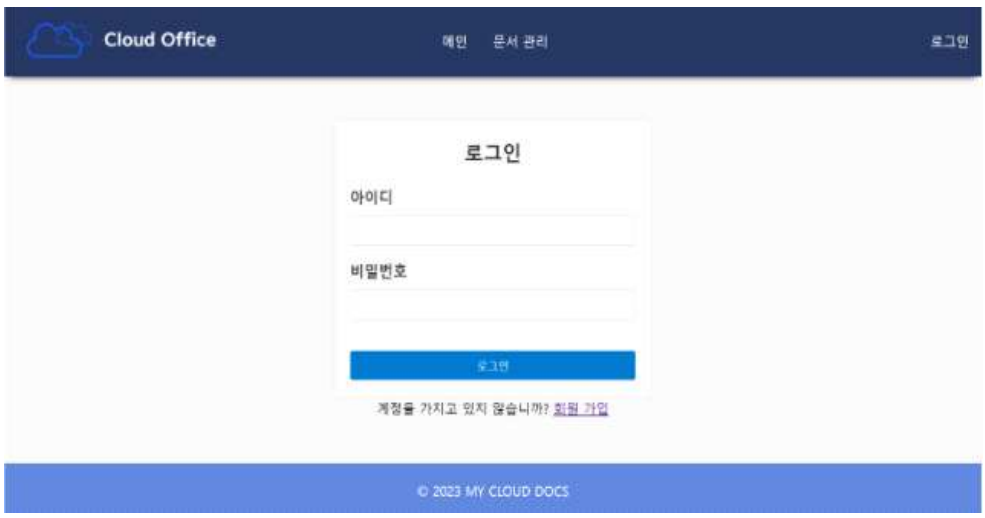
문서 시스템을 통해 자유롭게 .docx, .pptx의 파일을 보낼 수 있으며, 문서를 받은 사람은 발신자의 이름과 문서 번호, 승인 상태와 보고 일자, 발신부서 및 승인자를 확인할 수 있고 한 줄 평을 통해 피드백을 남길 수 있다.

3.3 DB 설계



[그림 2. Cloud Office DB 설계]

3.4.1 로그인&회원가입 페이지



[그림 3. Cloud Office 로그인 페이지]

[그림 4. Cloud Office 회원가입 페이지]

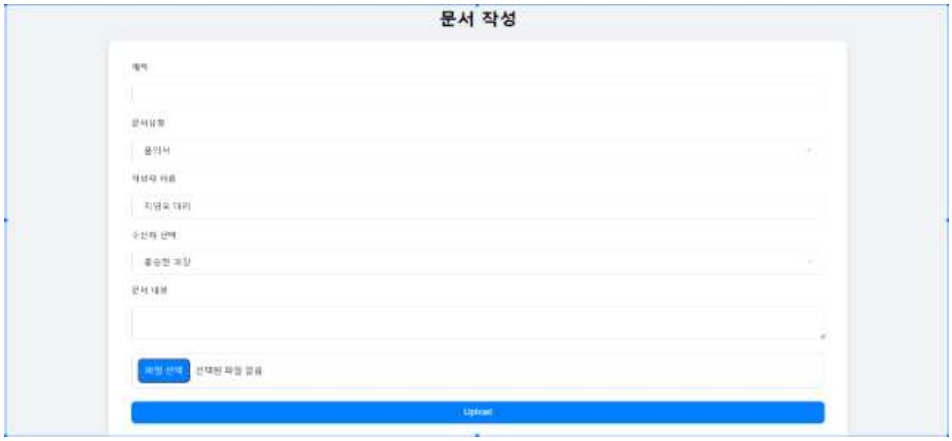
회원가입 페이지를 통해 아이디와 사원 이름, 비밀번호 등을 입력해 회원가입을 진행할 수 있으며, 관리자 페이지에서 관리자가 가입을 허용하고 직책을 부여하게 되면 로그인 페이지에서 로그인을 진행할 수 있다.

3.4.2 메인페이지

[그림 5. Cloud Office 메인 페이지]

로그인을 하면 로그인한 사람의 이름과 직책이 뜨게 되고, 향후 문서 처리 업무에 따라 처리 대기 문서, 반려된 문서, 결재 완료된 문서에 문서 목록이 표출된다.

3.4.3 문서 작성 페이지



[그림 6. Cloud Office 문서 작성 페이지]

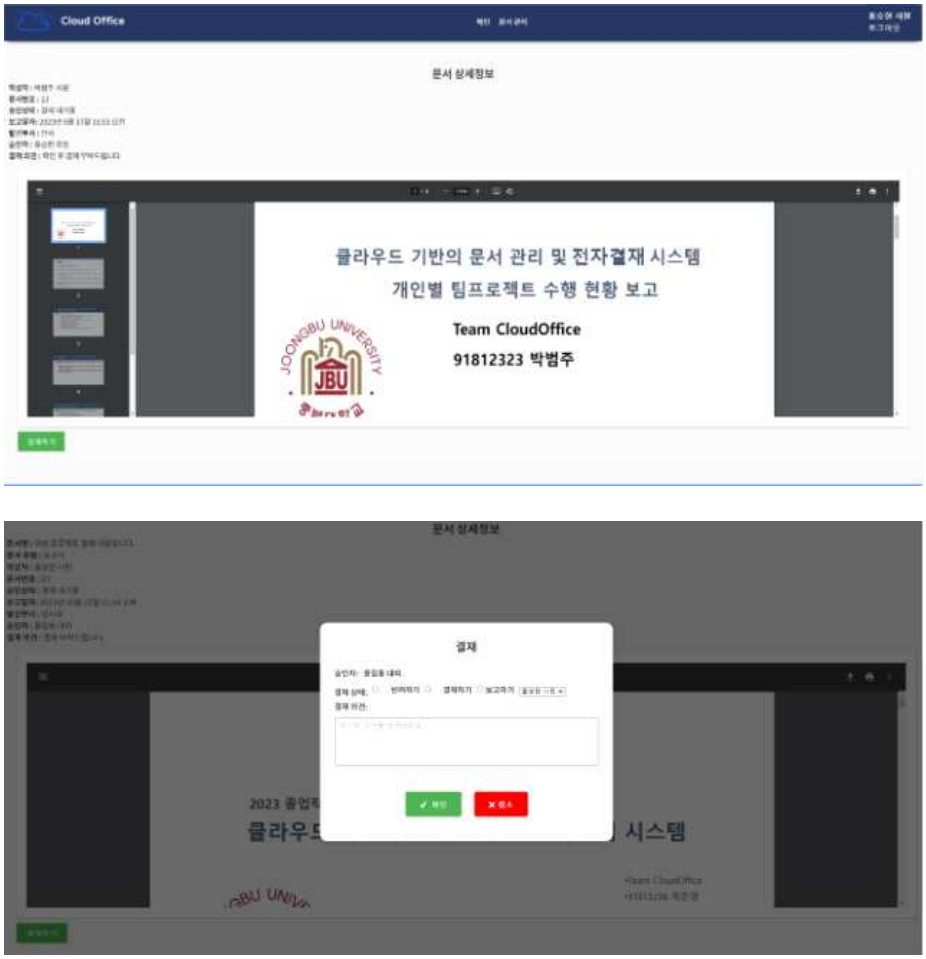
제목 란에 문서의 제목을 입력하고, 문서 유형과 수신자를 선택해 파일을 첨부해서 업로드하면, 수신자에게 pdf 형태로 문서가 수신된다.

3.4.4 문서 관리 페이지

[그림 7. Cloud Office 문서 관리 페이지]

문서 관리 페이지에서 자신에게 수신된 문서를 한눈에 볼 수 있으며, 문서 번호와 보고 일자 결재 여부 등을 볼 수 있다. 문서 명을 클릭하면 바로 문서 뷰어 페이지로 넘어가 문서를 열람할 수 있다.

3.4.5 문서 뷰어 페이지



[그림 8, 9. Cloud Office 문서 뷰어 페이지]

문서 뷰어 페이지에서 자신에게 할당된 문서를 PDF 형태로 열람할 수 있다. 뷰어 페이지에는 발신자와 문서의 정보 등이 포함되며, 결재 메뉴를 통해 반려와 결재 그리고 간단한 결재 의견 등을 작성할 수 있다.

3.4.6 결재 완료 페이지

[그림 10. Cloud Office 결재 완료 페이지]

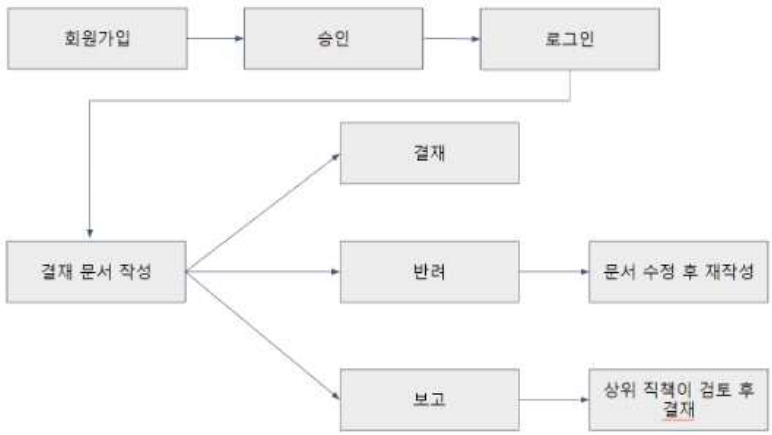
결재 완료 페이지에서 다시 수신된 문서와 함께 결재 의견과 결재 상태를 알 수 있다.

4. 활용 및 분석

4.1 프로젝트의 활용 시나리오와 사례 제시, 효과 분석

클라우드 오피스는 기업에서 사용할 전자 문서 관리 시스템이다. 종이 문서 대신 전자 문서를 사용해 업무 시간 절약 및 비용을 절감할 수 있으며, 서버에 수많은 결재 문서들을 안전하게 서버에 보관하고 관리할 수 있다. 또 투명하게 결재 기록이 남기 때문에 문제가 발생 시 기록을 통해 해결할 수 있다. 회사 내부망을 통해야만 접속이 가능하기 때문에 높은 보안성과 신뢰성을 유지할 수 있고, 손쉽게 문서를 공유할 수 있어 부서 간의 협업이 쉽게 이루어질 수 있다.

4.2 흐름도



[그림 11. Cloud Office 결재 흐름도]

5. 결론

5.1 결론 및 기대효과

클라우드 오피스 사용을 통해서, 종이 문서 사용에 필요한 자원과 비용을 절감할 수 있으며 전자 문서 이용을 통해 빠르고 간편한 문서 관리 및 정리가 가능할 것으로 기대한다. 또한 다수의 문서를 하나의 시스템에서 일괄적으로 암호화 및 백업 관리함으로써 높은 보안성과 신뢰성을 유지할 수 있고 손쉽게 문서를 공유하고 접근할 수 있어 협업 또한 용이해질 것으로 기대한다.

5.2 향후 과제

5.2.1 사용자 경험 개선

현재의 클라우드 기반 문서 관리 시스템의 사용자 인터페이스와 사용성을 평가하고, 사용자의 피드백을 바탕으로 개선할 부분을 찾아내기 위한 연구를 진행할 필요가 있다.

5.2.2 지속적인 백업 및 복구 시스템 개선

클라우드 저장소의 안정성을 위해 지속적인 백업 및 복구 시스템의 최적화 작업을 진행한다.

5.2.3 모바일 접근성 확대

스마트폰 및 태블릿과 같은 모바일 기기를 사용하는 사용자들의 접근성을 높이기 위한 모바일 앱 개발 및 최적화 작업을 진행한다.

5.2.4 다양한 파일 형식 지원

현재 지원되는 파일 형식 외에도 다양한 형식의 파일들을 지원하도록 확장하는 작업을 진행한다.

6. 별첨

6.1 참고문헌

6.1.1 Django 한 그릇 뚝딱 - 3가지 프로젝트로 파이썬 웹 프로그래밍 단련하기

6.1.2 파이썬 웹 프로그래밍 실전편 - Django(장고)를 활용한 쉽고 빠른 웹 개발, 개정판

6.1.3 이한영의 Django(장고) 입문

6.2 주소

6.1.1 깃허브 주소 : https://github.com/Marustar/CloudOffice

6.1.2 웹 서비스 주소 : www.clouldoffice.site/

6.1.3 앱 주소 : www.clouldoffice.site/

6.3 팀 소개

6.3.1 팀원 깃허브 주소

최준영 : https://github.com/Marustar/
강형욱 : https://github.com/C0mm4
홍승현 : https://github.com/honk0128
지영욱 : https://github.com/KooSinB
류재원 : https://github.com/MustHaveLove
박범주 : https://github.com/bjpark21

6.3 발표 자료

Part. 1 Outline

CloudOffice

개발 배경

보안성

격리된 네트워크를 통해 문서 유출 방지

접근성

다수의 문서에 네트워크를 통한 간편한 접근

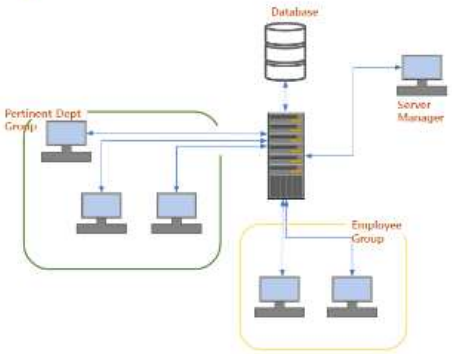
효율성

결제 기한을 정해 업무 기한의 효율적인 관리

Part. 1 Outline

CloudOffice

개발 목표



- Python, Django를 사용한 서버 구축
- 관리자 계정을 통한 안전한 계정 관리
- 전자 문서를 사용해 종이 문서보다 효율적인 결재 시스템 구현

Part. 2 Development plan

CloudOffice

타임라인

3월(~2주차)	4월	5월(~3주차)	6월(~3주차)
소프트웨어 구조 기획	웹 인터페이스 설계	데이터베이스 설계 및 구현	보안성 검토

3월(~4주차)	4월	5월(~5주차)	7~10월(~1주차)
웹 페이지 디자인	플로우 차트 작성	웹 클라이언트 개발	지속적인 개선 및 최종 발표 준비

Part. 1

Outline

CloudOffice

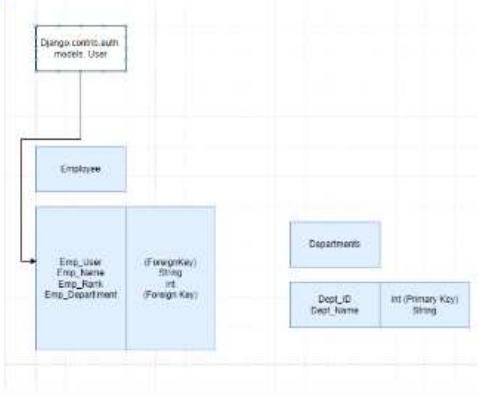
사용 기술

Part. 1

Outline

CloudOffice

DB 구조



1.사원 DB

사원 아이디, 이름, 직급 및 부서 데이터 지정(Foreign Key)

2.부서 DB

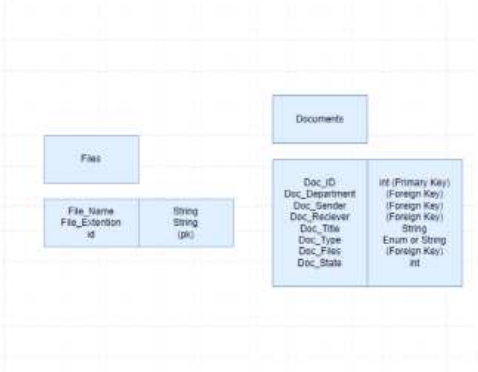
부서 데이터 저장

Part. 1

Outline

CloudOffice

DB 구조



1.파일 DB

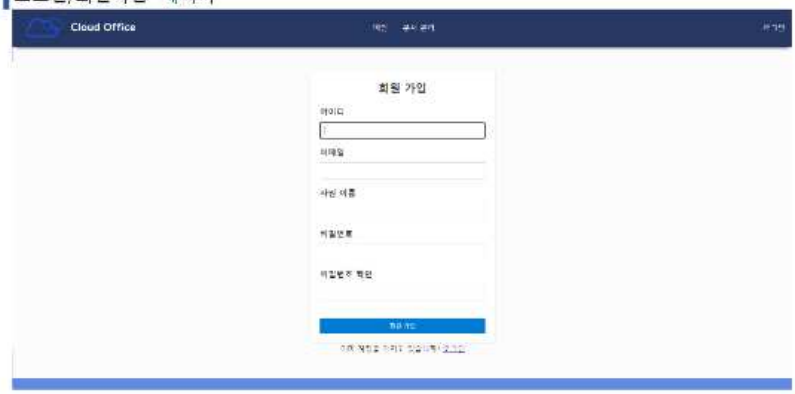
파일 이름과 번호 데이터 저장

2.문서 DB

문서 번호 및 발신지, 수신지, 문서 제목, 유형, 업로드 파일, 결재 상태 데이터 저장

Part. 2 Project Elements
CloudOffice

로그인/회원가입 페이지



Part. 2 Project Elements
CloudOffice

로그인 시도

Part. 2 Project Elements
CloudOffice

관리자 계정 승인

Part. 2

Project Elements

CloudOffice

메인 페이지

Part. 2

Project Elements

CloudOffice

문서 작성 페이지

Part. 2

Project Elements

CloudOffice

문서 관리 페이지

Part. 2 Project Elements
CloudOffice

문서 뷰어 페이지

문서 정보

작성자와 승인상태 외곤 등을 표시

Part. 2 Project Elements
CloudOffice

문서 뷰어 페이지

Part. 2 Project Elements
CloudOffice

결재 완료 페이지

Part. 2

Project Elements

CloudOffice

결재 완료 페이지

Part. 3

Future Improvements

CloudOffice

개선 사항

1. 보안성 향상을 위해 사용자의 반응이 없으면 자동 로그아웃 기능 추가

2. 웹메일 기능 구현

3. 지속적인 UI 개선 (사용자 프로필 사진 기능..etc)

4. 문서 작성의 수신자 목록에서 수신자의 부서 표시 및 정렬 기능 추가

Part. 3

Q&A

CloudOffice

Q&A