종합설계 프로젝트 수행 보고서

프로젝트명	**Crepe:** 원격 저장소를 활용한 그래픽 레이어 관리 플랫폼
팀번호	**S1-1**
문서제목	수행계획서**( )** **2**차발표 중간보고서**(O)** **3**차발표 중간보고서**( )** 최종결과보고서**( )**

2023.02.27

팀원 **:** 김유림 **(**팀장**)**

김주연

김태양

나은서

지도교수 **:** 전광일 교수

지도교수 **:** 정의훈 교수

문서 수정 내역

작성일	대표작성자	버전**(Revision)**	수정내용	
2022.12.27	김유림	1.0	수행계획서	최초작성
	김유림	1.1	주제 변경	
	김유림	1.2	계획서 내용 추가	
2023.02.27	김유림, 나은서	2.0	2차발표자료	양식에 맞게 기존 보고서를 토대로 작성

문서 구성

진행단계	프로젝트 계획서 발표	중간발표**1** **(3**월**)**	중간발표**2** **(5**월**)**	학기말발표 **(6**월**)**	최종발표 **(10**월**)**
기본양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식	계획서 양식
포함되는 내용	Ⅰ. 서론 (1~6) Ⅱ. 본론 (1~3) 참고자료	Ⅰ. 서론 (1~6) Ⅱ. 본론 (1~4) 참고자료	Ⅰ. 서론 (1~6) Ⅱ. 본론 (1~5) 참고자료	Ⅰ. 서론 (1~6) Ⅱ. 본론 (1~7) 참고자료	Ⅰ Ⅱ Ⅲ

이 문서는 한국산업기술대학교 컴퓨터공학부의

“종합설계**”**교과목에서 프로젝트**“Crepe:** 원격 저장소를 활용한

그래픽 레이어 관리 플랫폼**”**을 수행하는

(S1-1, 김유림**,** 김주연**,** 김태양**,** 나은서**)**들이 작성한 것으로 사용하기

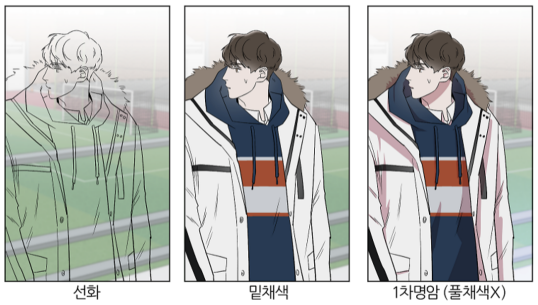
위해서는 팀원들의 허락이 필요합니다**.**

목 차

I. 서 론

1. 작품선정 배경 및 필요성

그림 한 장을 작업하는 사람들



웹툰채색 예

- 캐릭터 담당 작업자
- 배경 담당 작업자
- 이펙트 담당 작업자
- 말풍선 담당 작업자
- 피드백 담당자

- 그래픽 작업물이 완성되기까지 필요한 과정
  1. 최종본이 나오기까지 상당한 양의 파일을 로컬 컴퓨터에서 직접 백업한다.
  2. 담당 작업 별 반복적인 피드백을 반영한다.
     2-1. 피드백을 받을 때마다 여러 번 작업물 전달한다.
  3. 담장 작업 별 완성된 작업물을 합쳐서 최종본을 생성한다.
     3-1. 합친 후, 최종본이 되기까지 계속 피드백을 반영한다.
  4. 이전 작업물과 비교하여 최종본을 선택한다.
     4-1. 이전 작업물이 괜찮다면, 해당 작업물에 필요한 리소스 직접 찾아야 한다.
     4-2. 파일이 많거나 파일 이름이 겹칠 경우 파일 관리가 힘들어진다.
     4-3. 원하는 파일이 손상되었을 수 있다.

- 위 과정을 편리하게 만들어주기 위해 선정하였다.

2. 기존 연구/ 기술 동향 분석

가. Git

- 기능
  - 원격 저장소를 사용하여 코드를 관리한다.
  - 언제 어디서든 접근 권한만 있다면 코드를 다운받아 사용할 수 있다.
  - 이전 작업으로 돌아가거나 매 작업당 수정한 내용을 확인하기 용이하다.
  - 전체 코드를 한 번에 작성하지 않고 기능별로 나누어 구현하여 체계적으로 코드를 작성할 수 있다.

- 단점
  - CLI를 사용하여 진입장벽이 높다.
  - 다양한 GUI 툴이 많지만 너무 많은 기능이 내장되어 있어 툴을 익히기까지 시간이 오래 걸린다.
  - UI/UX가 친숙하지 않다.

- 개선 사항
  - 처음부터 gui만 사용하게 하여 cli 명령어로 관리하는 부분을 없앤다.
  - 실시간으로 바뀌는 부분과 최종 적용된 버전을 미리보기로 보여준다.
  - 튜토리얼 페이지를 제공한다.
  - ui/ux를 모든 사용자 친화적으로 개선한다.

나. 이전 기수 졸업작품 : 그림 깃

- 차이점 및 개선 사항
  - 레이어 단위로 수정이 가능하다.
  - 그림 리소스에 대해 레이어 별로 관리하기 때문에 (리소스 간의 순서 관리) 더 구체적인 데이터를 저장할 수 있다.
  - 그림에만 국한되어 있지 않고, 사진 편집, 움직이는 사진 프레임 저장(애니메이션) 등 다양한 디지털 아트 분야에서 활용이 가능하다.

3. 개발 목표

- 그래픽 작업물을 쉽게 관리할 수 있는 그래픽 레이어 관리 플랫폼 개발

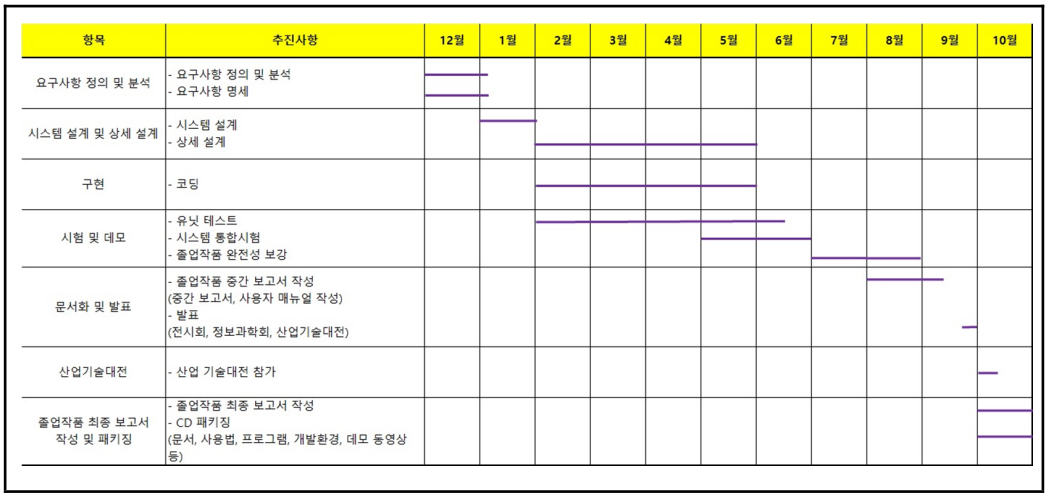
- 그래픽 파일을 원격 저장소에서 관리
  - usb 등이 없어도 언제 어디서든 개인의 작업물에 접근이 가능하다.
  - 지속적인 피드백을 받을 때 원격 저장소 접근 허용만 해주면 매번 작업물을 전달해 주지 않아도 된다.
  - 로컬에서 파일이 손상되어도 원격 저장소에서 가져와서 사용하면 된다.

- 작업 로그를 남겨 언제 어떤 부분을 작업했는지 **DB**에 기록
  - 작업 로그를 기반으로 원하는 파일 쉽게 찾을 수 있다.
  - 이전 작업으로 돌아가기 수월하다.

4. 팀 역할 분담

이름	주요 역할
김유림	Backend, DevOps(Infra)
김주연	Frontend, UI/UX
김태양	Backend, Spring Security
나은서	Frontend, UI/UX

5. 개발 일정



6. 개발 환경

- 개발언어: JAVA, CSS, HTML, JavaScript
- 사용 프레임워크: React, Spring Boot
- 개발방법론(옵션): 애자일
- 주요 라이브러리: boto3
- 서버 프로세서 : AWS EC2, AWS CloudFront
- DB : MySQL, Redis, MongoDB

II. 본 론

1. 개발 내용

- 그래픽 리소스 원격 저장소에서 관리
  - 원격 저장소에 그래픽 리소스를 백업한다.
  - 백업할때마다 작업물을 저장한다.
    - 저장할 때 저장 기록을 남긴다. 저장 기록은 저장된 작업물과 시간, 작업한 내용이 저장된다.
    - 저장 기록을 통해 언제든지 이전 작업물로 돌아갈 수 있다.
  - 메일이나 usb로 그래픽 리소스를 백업하거나 전달하지 않아도 된다.
    - 어떤 컴퓨터에서든지 원격 저장소에 로그인한다면 자신이 작업했던 내역을 다운받을 수 있다.
    - 원격 저장소 접근 권한을 가진 사람이라면 누구든지 작업 내역을 다운받을 수 있다.

- 편리한 협업 및 피드백 반영
  - 한 작업물에 대한 메인 작업물이 존재한다.
  - 한 작업물을 만들기 위해 여러 사람이 독립적으로 작업을 수행한다.
    - 메인 작업물을 다운받고 자신의 아이디로 업로드를 한다.
  - 독립적으로 수행한 작업물이 괜찮으면 메인 작업물과 합친다.
    - 이때 합쳐진 작업물을 미리 볼 수 있다.
    - 이때 합쳐진 작업물이 괜찮은지는 피드백 하는 사람이 원격 저장소에서 작업물을 다운받아 확인한다.

- 원격저장소를 쉽게 이해하고 사용
  - 모든 기능은 GUI로 제공되기 때문에 쉽게 이용 가능하다.
  - 튜토리얼 페이지가 제공된다.

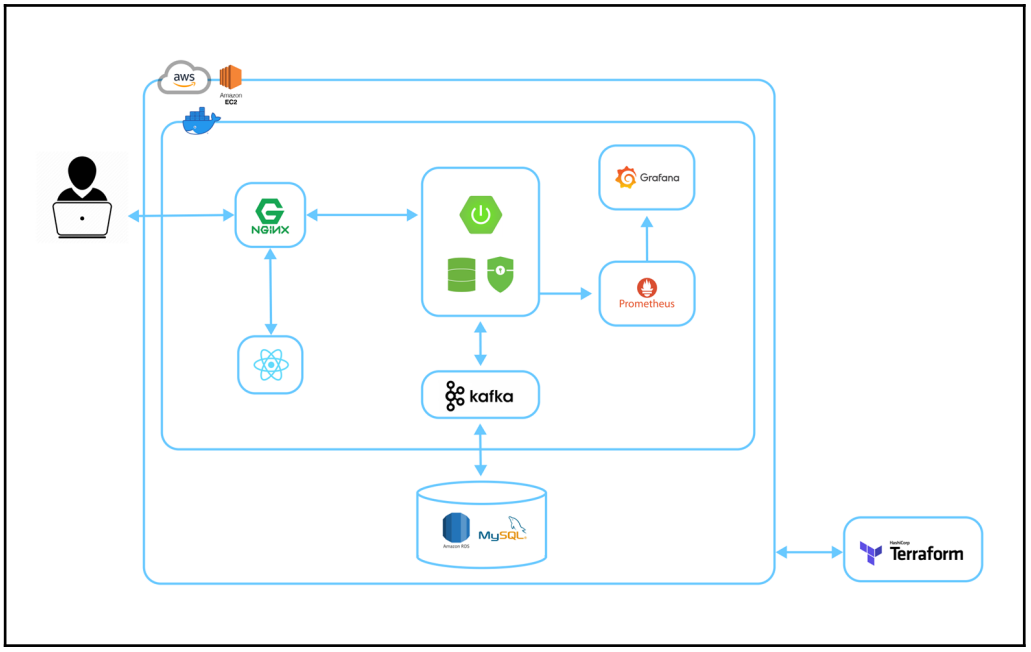
2. 문제 및 해결방안

- 정형/비정형 데이터의 저장
  - MySQL과 MongoDB를 복합적으로 사용

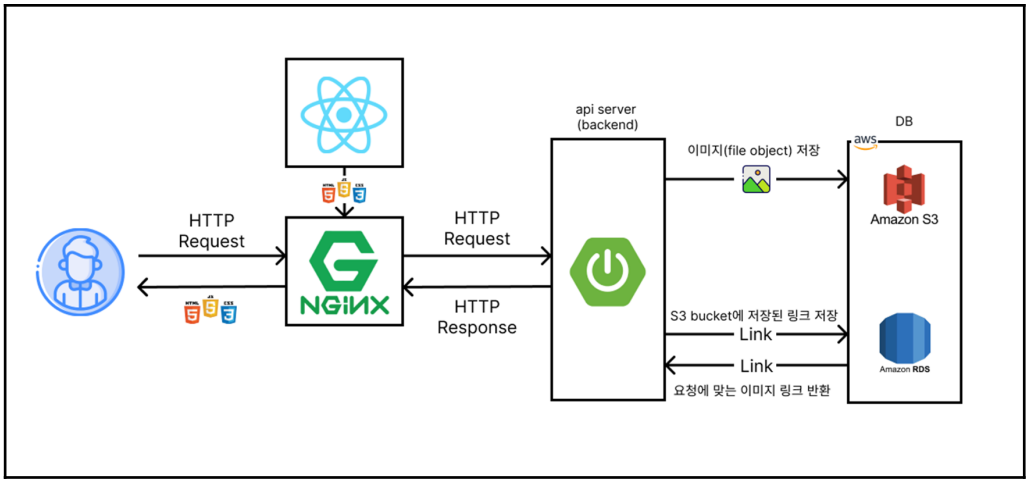
- 수정 내용 실시간 반영
  - 웹 소켓 사용

3. 시험 시나리오

- 전체 시스템 아키텍쳐



- 시스템 흐름도



가. 사용자 정보 입력 (로그인)

나. 원하는 프로젝트로 이동

다. 프로젝트 내의 원하는 브랜치로 이동

라. 브랜치에 이미지 업로드

마. 브랜치의 로그 내역 확인

4. 상세 설계

가. **Rest API** 설계

- Base URL: http://localhost:8080/api/v1

- User 도메인

U001	회원가입	POST	/users/signup
U002	로그인	POST	/users/signin
U003	refresh token 발급	POST	/users/token/refresh
U004	Verify Token	POST	/users/token/verify
U005	사용자 정보 조회	GET	/users/{user_uuid}
U006	전체 프로젝트 조회	GET	/users/{user_uuid}/projects

- Project 도메인

P001	프로젝트 생성	POST	/projects
P002	특정 프로젝트 조회	GET	/projects/{uuid}
P003	프로젝트에 참여중인 유저 가져오기	GET	/projects/users

- Branch 도메인

B001	브랜치 생성	POST	/branches
B002	전체 브랜치 정보 조회	GET	/branches
B003	특정 브랜치 정보 조회	GET	/branches/{uuid}

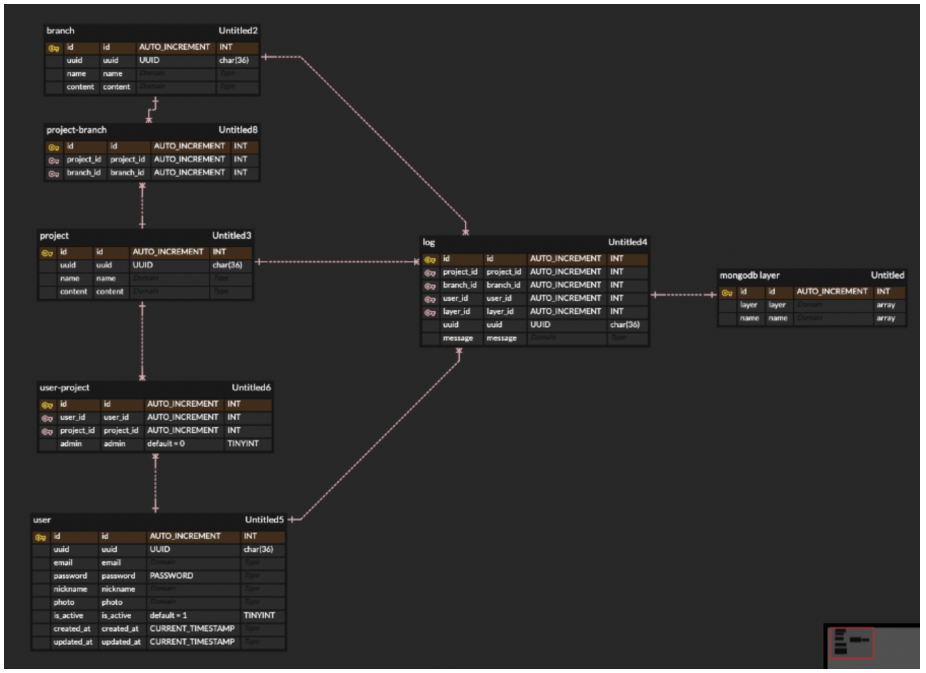
B004	브랜치에 참여중인 유저 가져오기	GET	/branches/users

- Log 도메인

L001	로그 추가	POST	/logs
L002	프로젝트/브랜치에 대한 로그 정보 조회	GET	/logs?branch_uuid={branch_uuid}
L003	로그 조회	GET	/logs/{log_uuid}

나**. DB**

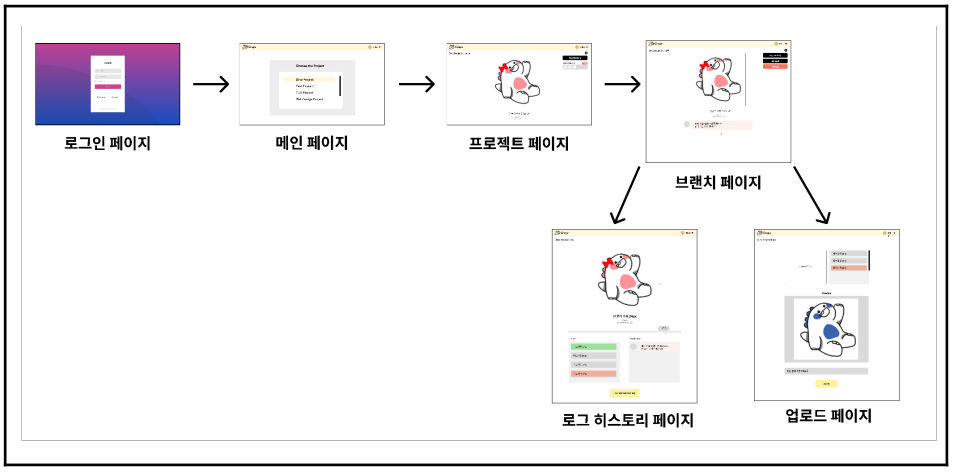
- **ERD Diagram**



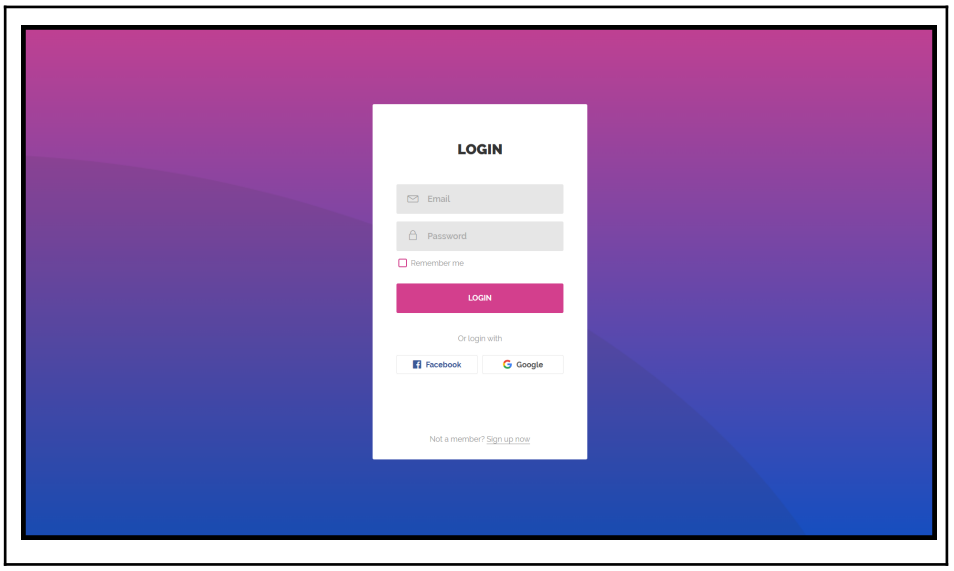
- **Entity** 설계

다. **UI/UX**

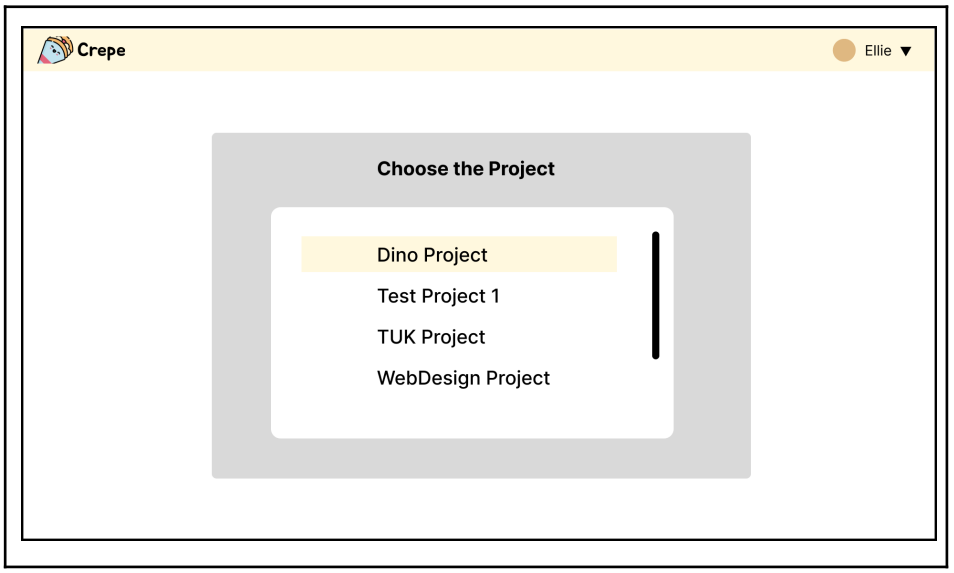
- 페이지 흐름도



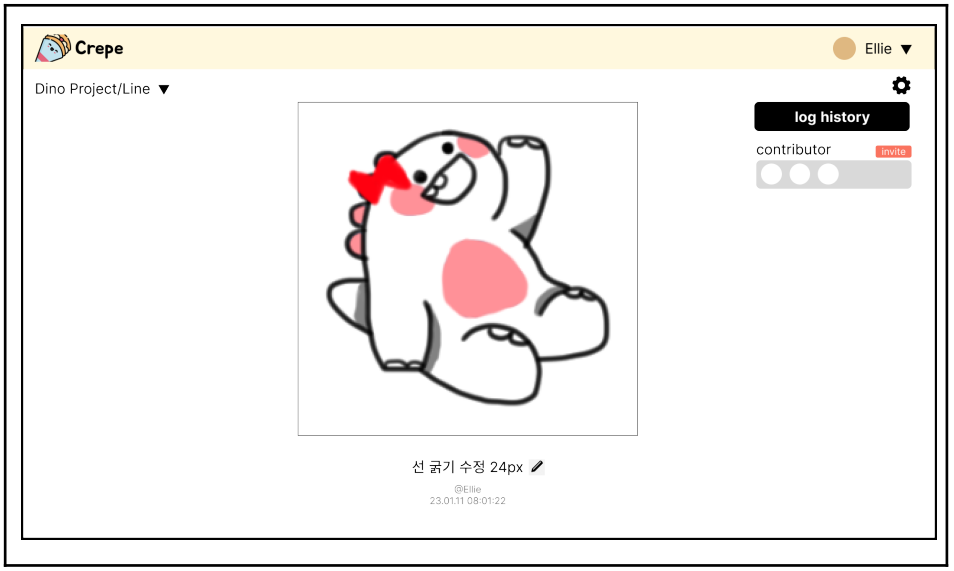
- 로그인 페이지



- 메인 페이지

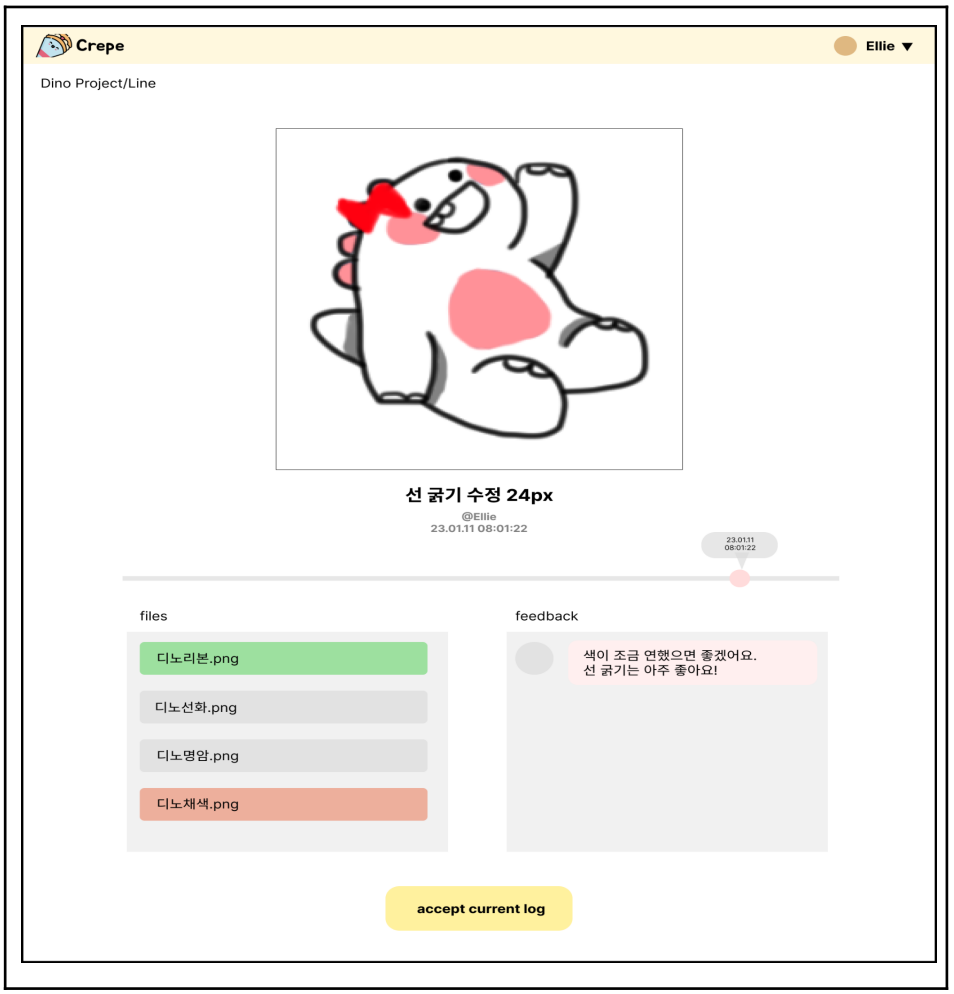


- 프로젝트 페이지

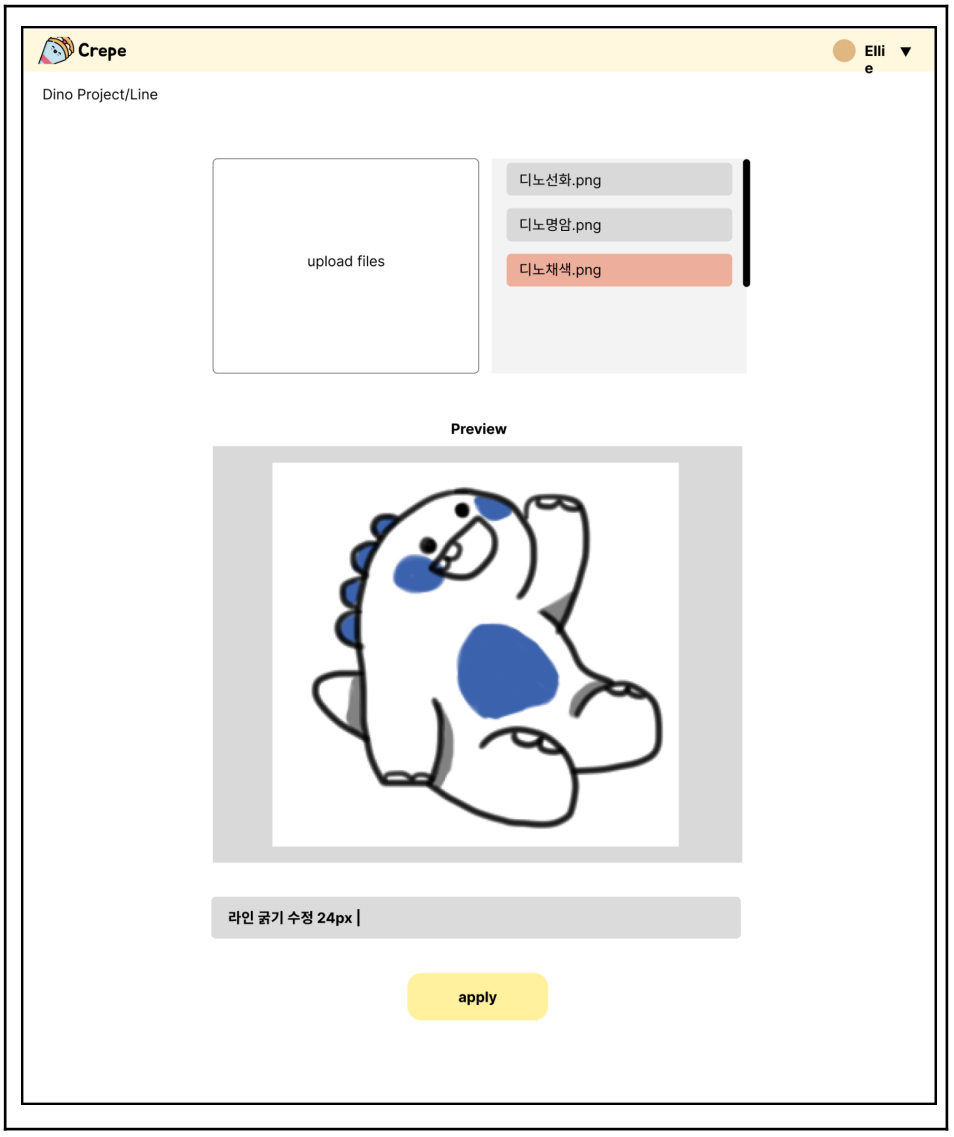


- 브랜치 페이지

- 로그 히스토리 페이지



- 업로드 페이지


Crepe
Ellie
Dino Project/Line
upload files
디노선화.png
디노명암.png
디노채색.png
Preview
라인 굵기 수정 24px |
apply