08 - 드랍 더 8bit - 2차보고서

Team-Info

(1) 과제명	이미지 보정 AI로 아이의 그림을 캐릭터로 바꾸고, 부모와 함께 동화를 만들어가는 서비스
(2) 팀 번호 / 팀 이름	08 / 드랍더8Bit
(3) 팀 구성원	**김재원(2021070) - 리더 Backend / AI** **시스템 설계:** ERD 설계, API 명세 정의, 서비스 UI 디자인 **구현 :** - [BE] Story, Quiz, EasterEggLetter API 개발 - [AI] 음기타(가칭) 생성 AI(openAI), 이미지 생성 AI(ControlNet+Stable Diffusion, Dalle) 개발 - [배포] EC2 + Nginx + Docker + github actions 이용한 배포 / RDS / S3 **조수현(2276309) - 팀원 Frontend / AI** **시스템 설계:** 화면 플로우 설계 및 UI/UX 구조 정의, 컴포넌트 구조 설계 **구현:** - [FE] 페이지 UI 개발, 백엔드 API 연동 및 데이터 바인딩, 프론트엔드 배포 - [AI] YOLOv8 객체 탐지 모델 파인튜닝 및 AI 개발 **조서정(2276305)- 팀원 Backend / AI** **시스템 설계:** ERD 설계, API 명세 정의, 서비스 UI 디자인 **구현:** - [BE] 사용자 인증/로그인 기능 구현, Profile, EasterEggVoice, Voice, 책장, 알림 API 개발 - [AI] ElevenLabs 음성 복제 AI 개발, YOLO API 개발
(4) 팀 지도교수	신재원 교수님
(5) 과제 분류	신학과제
(6) 과제 키워드	동화 생성, 이미지 보정AI, AI음성 나레이션, 인터랙티브 스토리텔링, 디지털 책장
(7) 과제 내용 요약	저희는 이미지 생성 **AI(Stable Diffusion+ControlNet+LoRA)**를 통해 아이의 그림을 캐릭터로 발전시키고, **음성 복제 AI(ElevenLabs)**과 **객체 인식 기반 이스터에그(YOLOv8)**를 결합하여 부모와 아이가 함께 상호작용하는 인터랙티브 동화 서비스를 개발하고자 합니다

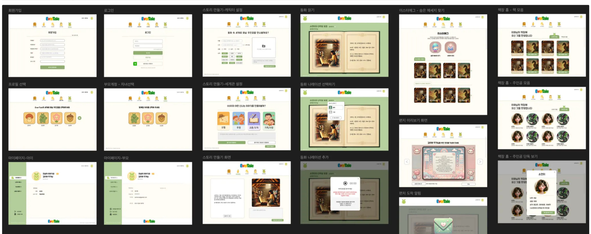
Project-Summary

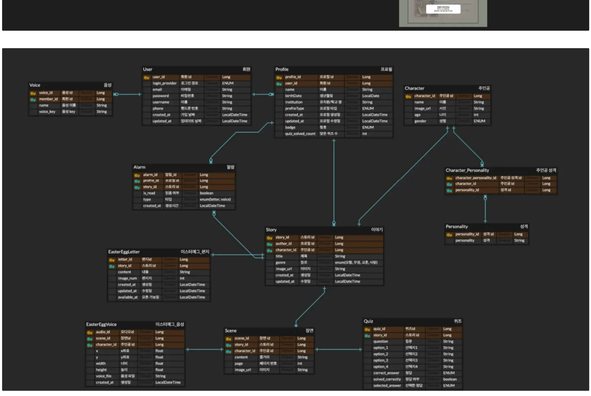
항목	내용
(1) 문제 정의	AI 기술이 아이의 창의력에 어떤 도움을 줄 수 있을지 고민한 끝에 이미지 보정 AI 등을 활용해 아이의 그림을 캐릭터로 변환하고, 부모와 함께 동화를 만들어가는 서비스를 개발하고자 한다. 연구에 따르면, 5~8세 아동은 하루 평균 약 3시간을 기기를 통한 미디어 소비에 사용하며, 그 대부분을 영상 시청에 할애하고 있다. 그러나 수동적 영상 시청을 할 경우 정서 표현 수준이 낮았던 반면, 인터랙션 활동을 경험했을 때 정서 어휘 사용이 약 45% 더 높아지는 것으로 나타났다. 이러한 결과는 단순한 미디어 소비가 아닌, 아이가 사고력과 창의력을 마음껏 표현할 수 있는 활동이 필요함을 보여준다. 따라서 우리는 이러한 활동을 AI 기반 동화책 제작으로 구현하여, 아이가 능동적으로 창의성을 발휘할 수 있도록 하고자 한다. 그러나 현재 부모와 아이 간의 정서적 교감을 온전시키면서, 동시에 가정과 유치원 모두에서 관리·활용할 수 있는 동화책 제작 서비스는 부족하다. 이에 따라 AI를 활용해 창의적 발달, 정서적 교감, 교육 현장과 가정 연계 성을 아우를 수 있는 새로운 서비스의 필요성을 느끼고 있다. **Target Customer 및 Pain points** 본 서비스의 주요 대상 고객은 다음과 같다. **1. 4~8세 아동** - 자신의 생각과 감정을 직접 표현할 기회가 부족한 아이 - 상상력을 키울 수 있는 능동적 활동이 제한적인 아이 **2. 맞벌이 부모(양육자)** - 바쁜 일정으로 자녀에게 충분한 교육적 콘텐츠를 제공하기 어려움 - 영상 콘텐츠에 의존하는 양육 방식에 한계가 느끼는 부모 **3. 국공립 유치원 교사** - 다양한 성향의 아이들을 반복시킬 수 있는 콘텐츠 부족 - 가정과 연계하여 활동할 교육 자료가 부족 **다음과 같은 문제를 해결하여 창의력을 키우려는 교육 도메인을 만드는 것을 목표로 한다.** 1. 아이의 적극적인 참여 유도: 기존 동화책/앱은 수동적 소비 위주로 창의력 발달에 한계가 있기에 직접 동화를 쓰고 그리며 제작에 적극적으로 참여할 수 있는 콘텐츠가 필요하다. 2. 맞벌이 부모와 아이의 정서적 교감 돕기: 퇴근 후 시간이 부족해도 맞벌이 부모는 아이와 정서적 교감을 효과적으로 이어갈 수 있는 방식이 필요하다. 3. 기존 연구 결과를 반영한 콘텐츠: 기정과 유치원 모두에서 활용하고 관리할 수 있는 구조가 필요하다.
(2) 기존연구와의 비교	**1. 기존 동화 생성 앱(랜덤 이야기 생성 / AI스토리고보, dreammoven)** **[단순 즐거운 생성]** - 초기 단계에서 전혀 즐거움을 한 번에 생성하고, 사용자가 결과를 직접 수정·보완하는 방식 - 장점: 빠른 즐거움 생성 가능 - 단점: 아이와의 상호작용이 전혀 없으며, 즐거움을 변경하려면 전체를 다시 생성하거나 사용자가 수동으로 수정해야 함 **[단순 프롬프트 기반 이미지 생성]** - DALL-E 등 모델을 사용해 즐거움에 기반한 단순 프롬프트 입력을 통해 완성 이미지를 생성 - 장점: 신속한 이미지 생성, 다양한 캐릭터 표현 가능 - 단점: 아이가 원하는 구도·의상·외형을 반영하기 어려우며, 원하는 이미지를 얻기 위해서는 추가 API 호출이 필요 → 불필요한 토큰(결제) 소모 증가 **2. 본 과제의 장점 및 차별성** **[질의응답 기반 즐거운 생성]** - AI가 아만 즐거움를 바탕으로 질문을 제시하고, 아이의 응답을 반영하여 다음 즐거움를 생성 - 예시 질문 "스쿠먼는는 힘이나 다리밖에서 반짝이는 상자를 발견했어요. 소원이는 무엇을 발견했을까요?" 응답 "물고 상자" 즐거움 생성: "요정은 말했어요. '이 물감을은 계절의 마법을 지녔단다. 하늘에 뿔을 때마다 봄, 여름, 가을, 겨울이 펼쳐지지.' 소원이가 초록 물감을 집어자 방 안에 봄바람이 불어왔어요." **[아이의 그림 기반 Scribble 이미지 생성]** - 아이가 직접 그린 그림을 업로드 → ControlNet이 스케치 선을 인식 → Stable Diffusion + LoRA가 배경과 캐릭터를 보강하여 최종 이미지 생성 **[음성 복제 기능]** - 가족 구성원의 목소리를 AI 나레이션으로 등록 가능 - 형구, 보로로, 아기상어, 캐쿼렌, 타니팅 등 캐릭터 음성을 흉내내는 음성 놀이 기능 제공 **[이스터에그 기능]** - 완성된 그림에 대해를 Yolov8 객체 탐지 모델을 사용하여 인식 후 화표 반환 → 클릭 시 슬거진 음성 에시지(이스터에그) 팝업 - '사랑의 편지' 기능: 열람 가능 시간 및 열람 기능 자체 → 특정 시간이 지나야 부모의 메시지를 확인 가능 **[관리자 전용 인터페이스 분리]** - 자녀별 동화 관리 기능 제공 (삭제된 동화리라입 동화 조회, 마지막 편지 음성 메시지 확인 등) - 이스터에그 페이지 접근은 관리자만 가능 - 삭제 및 열람 시간 설정 지원 - 셋플릭스 구조 도입 → 다시보기 가능 및 교육자 맞춤형 관리에 최적화
(3) 제안 내용	최근 연구에 따르면, 일본 등지 기업을 제외한 인터랙티브 동화 콘텐츠는 유아의 창의성과 언어 표현력 향상에 긍정적인 효과를 보였다. 단순히 이야기를 수동적으로 듣는 것이 아니라, 아이가 스스로 질문에 답하거나 자신의 경험을 그림·역할극·이야기 형태로 표현할 때 학습 효과가 더 크게 나타났으며, 이러한 과정은 부모·자녀 간의 상호작용을 촉진하고 나아가 사회·정서 학습(SEL) 역량을 자기 삶과 연결하는 데도 효과적이었다. 특히, 이야기를 그림으로 재구성하는 활동은 유아의 상상력과 창의성, 언어 표현 능력을 동시에 자극하는 것으로 확인되었다. 단순히 그림만 그리는 것보다 스토리텔링과 결합했을 때 아이가 시고를 조직하고 자기 생각을 구체적으로 표현하는 능력이 더욱 발달하였다. 이는 본 서비스가 목표인 "아이의 그림을 캐릭터화해 동화로 확장"과 부합한다. 아울러, 기정과 유치원의 교육이 연계될 때 학습 효과와 지속성이 높아진다는 점도 여러 연구에서 강조되기, 부모가 곁에 계속해 적극적으로 참여하면 아이의 이해와 몰입이 깊어지고, 학습 경험이 일상 속에서 반복·강화되어 발달 효과가 배가된다. 따라서 부모가 자녀와 함께 동화를 만들고 이야기를 나누는 과정은 단순한 놀이를 넘어 정서적인지적 발달을 동시에 촉진하는 학습 경험이 된다. 이러한 연구 결과를 바탕으로, 본 프로젝트는 AI 기반 동화 생성 및 대화형 학습 시스템을 제안한다. 아이의 그림을 캐릭터로 변환하고 부모와 함께 "나만의 동화책"을 만들어가는 과정을 통해, 기존의 일방향적 교육 방식을 넘어 유아의 창의성과 언어 표현력, 그리고 사회·정서 발달을 아우르는 차별화된 학습 경험을 제공하고자 한다. [참고 논문 및 기사] *[학위논문] 질문 유도 기법을 적용한 유아 인터랙티브 동화 콘텐츠 디자인 연구(A Study on the Design of Children's Interactive Fairy Tale Content Using Question-Inducing Techniques)* *스토리텔링을 활용한 그리기활동이 유아의 창의성과 언어표현력에 미치는 효과(The Effect of Storytelling-used Drawing Activities on Young Children's Creativity and Linguistic Expressiveness)* *유치원교사의 가정연계교육 활성화 방안 A Study on the Revitalization of Domestic Education Connected with Kindergarten* [배경지] 1. AI 동화 생성 및 즐거리 확장 - 아이의 이름, 성격, 관심사 등을 반영하여 AI가 맞춤형 동화를 자동 생성 - 아이가 직접 그린 그림을 캐릭터로 변환하고, 즐거리 전개 과정에 반영 2. 실시간 인터랙티브 대화 시스템 - 주인공, 장르, 이야 Scene, 아이의 답변을 기반으로 아이와 자연스러운 질의응답이 오가는 구조 - 아이가 대답한 내용이 이야기 전개에 반영되어 능동적 학습 경험을 유도 3. 부모-자녀 상호작용 강화 - 부모가 자녀와 함께 동화를 완성하고, 슬거진 에세지나 음성 나레이션을 통해 특별한 교감 제공
(4) 기대효과 및 의의	**1. 스토리텔링 기반 활동 효과** - 한구 결과, 유아가 스토리텔링과 그림 활동을 함께 할 때 창의성과 언어 표현력이 크게 향상됨 - 단순히 이야기를 듣는 수동적 학습이 아닌, 아이 스스로 이야기를 만드는 경험'을 통해 자기 생각을 구조화하고 표현할 수 있는 능력이 강화됨 **2. 능동적 학습 경험** - 그림, 역할극, 이야기 등 다양한 방식으로 자신의 경험을 표현하면서 몰입도가 높아짐 - 이러한 과정은 아이가 단순 시식 습득을 넘어, 실제 상황에서 적용 가능한 역각적 성찰을 가능하게 함 **3. 부모-자녀 대화 심화** - 부모와 함께 동화를 만들고 이야기를 나누는 과정에서 대화가 더 깊고 풍부해짐 - 아이는 부모와의 상호작용을 통해 사회적·정서적 개념을 자연스럽게 배우고, 이를 자기 삶에 연결할 수 있음 **4. AI교육 서비스 기대효과** - 아이가 그린 그림을 AI가 캐릭터로 변환해주어 창작의 재미와 몰입도를 높임 - 부모와 함께 나만의 동화를 만들어가는 과정에서 개별 경험이 반영되고, 이를 통해 학습의 의미가 개인화됨 - 결과적으로 아이의 창의성(표현력)과의 발달을 동시에 촉진하는 교육 도메인 서비스로 발전 가능
(5) 주요 기능 리스트	**1. 동화 생성** ~~1.1 이미지 생성 : 그림은 Stable Diffusion + ControlNet(Scribble), LoRA 파인튜닝을 통해 아이가 직접 그린 스케치가 캐릭터로 변환됨~~ ~~1.2 동화 본문 / 즐거리 생성 : 주인공/장르/이야 Scene/아이의 답변을 기반으로 AI가 즐거리로 즐거리를 생성하여 아이가에 생동감을 부여~~ ~~1.3 음성형 드랍드라(아이가 생성 즐거리 음성에 더하는 기능)~~ **2. 퀴즈** 2.1 퀴즈 생성 - 생성된 동화 속 씬을 기반으로 ChatGPT API를 활용해 4지선다 퀴즈를 자동 생성 - 학습 요소와 재미 요소를 동시에 결합해 SEL(사회·정서 학습) 강화 2.2 정오 보기 : 퀴즈의 정답률을 기반으로 (ex) 전설의 이야기꾼 과 같은 힘동화 이모티콘 부여 **3. AI 나레이션** 3.1 음성 복제 - 부모의 음성을 ElevenLabs API로 자연스럽게 변환해 동화 속 나레이션으로 삽입 3.2 음성 나레이션 - 아이가 특정 페이지에서 원하는 목소리로 음성 요청 시, 해당 목소리로 동화를 읽어줌 **4. 이스터에그(편지, 목소리)** 4.1 그림 속 숨겨진 음성 메세지 - 동화 속 그림에 객체 인식 모델(YOLOv8)로 숨겨진 이미지를 삽입, 아이가 찾아 클릭할 수 있도록 설계 - 해당 객체 클릭 시 자녀에게 특별한 '음성 메시지'를 전달할 수 있도록, 놀이와 정서적 교감을 동시에 강화 4.2 마지막 페이지의 사랑의 편지 : 동화의 맨 마지막 페이지에 매번 편지지와 함께 사랑의 메세지를 남길 수 있음

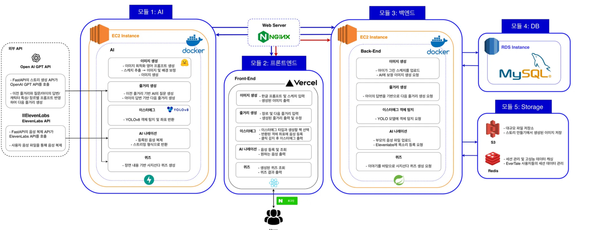
Project-Design & Implementation

(1) 요구사항 정의

1. 동화 생성 기능	1.2 이미지 생성	- 아이 그림의 윤곽선을 Stable ControNet을 통해 Scribble - AI가 → Stable Diffusion → LoRA fine tunning 통해 아이의 그림을 캐릭터화
	1.2 즐거리 생성	- 주인공/장르/이야 Scene 기반으로 AI가 집의 - 아이의 답변을 기반으로 다음 즐거리 생성
2. 퀴즈	2.1 퀴즈 생성	- 스토리의 각 Scene 내용을 기반으로 ChatGPT API가 4지선다 퀴즈를 생성
	2.2 정오 부여	- 퀴즈의 정답률을 기반으로 힘동화 이모티콘 부여
3. AI 나레이션	3.1 음성 복제	- 부모의 음성을 초기에 등록하고, 이후 ElevenLabs로 목소리 복제
	3.2 음성 재생	- 동화의 각 페이지마다 등록된 음성으로 자동 나레이션 제공
4. 이스터에그	4.1 그림 속 숨겨진 음성 메세지	- 객체 탐지 모델인 Yolov8로 이미지 속 객체를 탐지하여 클릭 가능하도록 설정 - 해당 객체를 클릭하면 부모의 음성 메세지 출력
	4.2 마지막 페이지의 사랑의 편지	- 각 스토리의 마지막 페이지에 편지지 형식의 '사랑의 편지'를 삽입할 수 있음
5. 인터페이스 분리	5.1 관리자 전용 인터페이스	- 관리자(부모)교사)만 자녀 사용자 인터페이스를 구분하여 부모 화면에서만 이용 가능한 기능(이스터에그)를 별도로 제공함
	5.2 이스터에그 페이지 접근 권한	- 이스터에그 생성 주체인 관리자만 해당 페이지 접근 권한 부여
	5.3 마이페이지	- 관리자 마이페이지에서만 화면 정보 수정 가능
6. 디지털 책장	6.1 스토리별 조회	- 완성된 동화책만 조회하기
	6.2 주인공별 조회	- 생성한 주인공 캐릭터만 조회하기
7. 사용자인증	7.1 회원가입	- 부모 이메일로 회원가입 - 네이버 로그인 API로 회원가입 가능
	7.2 로그인	- 일반회원가입,네이버 소셜 로그인 가능
8. 알람	8.1 알람 기능	- 이스터에그가 등록되면 아이에게 알람이 또그, 확인할 수 있음






모듈 1 : AI
GPT API
Open AI GPT API
- ChatGPT-3.5 모델 API 사용
- Open AI GPT 서버와 연결
- 이전 동화에 등장했던 등장인물 정보 + 사용자 입력 텍스트를 받아 새로운 동화 텍스트를 생성
ElevenLabs
ElevenLabs API
- Flask에서 받은 텍스트 API 사용하여 ElevenLabs 서버와 연결
- 사용자 음성 파일을 동화 음성 제작
EC2 Instance
docker
AI
이야기 생성
- 이전의 동화에 등장했던 등장인물 정보 + 사용자 입력 텍스트를 받아 새로운 동화 텍스트를 생성
음성 생성
- 이전 동화의 대사를 음성 형태로 변환하여 저장하고 클라이언트에 전달
이미지제작 Node.js
- Node.js 서버에서 이미지를 생성할 때 사용할 파일 및 데이터를 준비하고 파이썬에게 전달
AI 나레이션
- 동화의 음성 제작
- 스토리텔링 형태의 음성 제작
퍼즈
- 일반 서버 기본 서버라고 하드 코딩 됨
Web Server
Nginx
모듈 2: 프론트엔드
Front-End
Vercel
이야기 생성
- 현재 진행되고 있는 동화에 입력 텍스트를 전달
음성 생성
- 입력 된 텍스트를 음성의 일부 이야기 생성하고 음성 파일 제작
이미지제작
- 이야기에서 이미지를 생성할 때 사용할 파일 및 데이터를 받아 이미지를 출력
AI 나레이션
- 음성 파일 및 텍스트 파일 저장
퍼즈
- 생성된 퍼즈 로직 - 하드 코딩 됨
User
모듈 3: 백엔드
EC2 Instance
docker
Back-End
이야기 생성
- 하이드 그만 스토리텔링 입력드 - AI에 보낼 이야기 생성 요청
음성 생성
- 이야기 입력받아 기본인공지 다음 음성 생성 요청
이미지제작 Node.js
- Node.js 서버를 이용하여 이미지 생성
AI 나레이션
- 부모가 동화 생성 입력드 - Elevenlabs에 텍스트 형태 요청
퍼즈
- 이야기를 바탕으로 서사구조 퍼즈 생성 요청
모듈 4: DB
RDS Instance
MySQL
모듈 5: Storage
- 개발자용 서명 키정보
- 스토리 진행 기록이 생성될 때마다 해당 이야기 정보를 저장하기 위해 생성된 이야기 저장
S3
- 생성 된 텍스트 및 오디오 파일 저장 - Eve-Tale 이용자들이 해당 이야기 생성 전에 이야기 정보를 저장하기 위해 생성된 이야기 저장
Redis

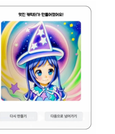
항목	내용
(2) 전체 시스템 구성	**1) AI용자 인터페이스 (Front-End)** - React를 기반으로 웹페이지를 제작하며, 개발 언어는 JavaScript이다. - fetch API를 사용하여 Spring Boot 백엔드 서버 및 FastAPI 기반 AI 서버와 HTTP 요청/응답을 처리한다. - 사용자의 상태 관리 및 동적 UI 업데이트를 효율적으로 관리하기 위해 React의 상태 관리 기능을 활용하여 페이지 간 원활한 라우팅을 지원한다. - React Router의 createCookieSessionStorage를 활용하여 쿠키 기반 세션(session)에 인증토큰 저장하고, getSession, commitSession, destroySession 함수를 통해 세션을 생성·조회·삭제한다. 이를 통해 브라우저에는 암호화된 세션 쿠키만 남아 보안성을 강화한다. - 프론트엔드는 애플리케이션은 브랜치 기반 Preview/Production 환경을 구분해 자동 CI/CD를 지원하는 Vercel을 배포 플랫폼으로 활용한다. **2) 백엔드 서버와 인증/인증 구조 (Back-End)** - Java 17 런타임 기반으로 Spring Boot 3.1.4 웹 기반 프레임워크를 활용하여 RESTful API 서버로 구현되었으며 IDE는 IntelliJ를 사용한다. [상세 기술 스택] - 데이터 관리: 데이터베이스는 AWS RDS(MySQL)을 구축하여 사용하고 영속성 관리를 위해 JPA (Spring Data JPA 3.1.4)를 사용한다. 고성능 데이터 캐싱 및 세션 관리를 위해 Redis (Spring Data Redis 3.1.4)를 NoSQL(캐싱) 솔루션으로 활용한다. - 보안인증: Spring Security 3.1.4, OAuth 2.0, JWT (JJWT 0.12.5)을 사용한 토큰 기반 인증을 진행한다. 내외부 API를 이용해 데이터 교환 방식에 사용자 인증 및 인가를 처리하며 JWT (JJWT 0.12.5)를 사용하여 토큰 기반 인증 방식을 구현한다. - 배포 및 운영: 배포 환경은 AWS EC2를 기반으로 하며 Github Actions를 통해 서버 배포 및 CI/CD 파이프라인을 자동화한다. Spring Cloud AWS 2.2.6.RELEASE를 활용해 AWS 서비스(S3, RDS)와의 연동을 처리한다. API 명세의 지침화 및 관리를 위해 Swagger (Springdoc OpenAPI 2.1.0)를 적용하여 개발 편의성을 높였으며 데이터 유효성 검증을 위해 spring-boot-starter-validation 3.1.4를 사용한다. **3) 스토리 생성 기능(음성 복제 기능(AI)** - 프론트엔드에서 업로드된 scribble 이미지는 Spring Boot의 AjaxClient를 거쳐 nginx를 통해 FastAPI 서버에 전달되며, Hugging face의 Stable Diffusion(dreamshaper-v8), ControlNet, LoRA 파인튜닝 모델(LarryXDrawjanime(screencapStyle_v230epoch))을 통해 동화 생성이 진행된다. - FastAPI 서버는 프론트엔드 요청과 백엔드 데이터 전달을 기반으로 OpenAI ChatGPT API를 호출하여 줄거리 및 퀴즈를 생성하고, ElevenLabs TTS를 활용해 음성을 합성한다. **4) 객체 탐지 기능(AI)** - AI 서스템은 Python(3.9.6) 언어와 Numpy(2.0.2) 기반으로 개발되었다. Spring Boot에서 전달받은 S3 이미지 URL을 활용해 YOLOv8 객체 탐지 모델이 실행되며, 특정 객체의 좌표(x, y)를 반환한다. 이 좌표 정보 해당 객체 이미지 생성 에서 프롬 값입력는 이스터에그 기능이 구현된다. **5) 백엔드 및 AI 서버 연동** - 전체 AI 서빙은 FastAPI 기반 AI 서버에서 운영되며 IDE는 pycharm을 사용하며, Nginx 서버를 통해 이미지 생성 객체 탐지 텍스트 생성 음성 합성이 유기적으로 연동되어 백엔드 및 데이터베이스에 실시간으로 저장된다.
(3) 주요엔진 및 기능 설계	**주요 엔진** - **동화생성** **OpenAI GPT API**: https://platform.openai.com/docs/guides/gpt **Hugging Face diffusers (Stable Diffusion + ControlNet)**: https://huggingface.co/docs/diffusers/index **ControlNet - scribble 모델**: https://huggingface.co/lllyasviel/sd-controlnet-scribble **Dreamshaper 모델 (Stable Diffusion 기반)**: https://huggingface.co/Lykon/dreamshaper-8 **LoRA 모델**: https://civitai.com/models/4982/anime-screencap-style-lora - **음성복제 : ElevenLabs**: https://docs.elevenlabs.io - **객체 탐지 : YOLOv8** : https://docs.ultralytics.com/ko/models/yolov8/ **1. 동화 생성** **기능 및 역할 :** 1.1 이야기 생성 (캐릭터 생성) : 사용자가 설정한 주인공 정보를 프롬프트에 반영하고, 업로드된 스케치 이미지와 한글 프롬프트를 기반으로 이미지를 생성한다. 이 과정에서 이야기 자체 그림 스케치를 추출 및 보정하여 최종 이미지에 반영합니다.에서 이야기 동화 삽화 제작 과정에 직접 개입하고 창의성을 표현할 수 있도록 돕는다. 1.2 동화 요약 기반 줄거리 생성 : AI가 이전 줄거리와 주인공 정보를 바탕으로 동화를 생성 하고, 이야기 이해 당면하면 그 답변을 반영해 다음 줄거리를 생성한다. → 줄거리와 이미지 생성 과정에서 '수정', '다른 스타일', '가족/사랑/모험' 등 키워드 입력으로 세분화하여, 각 장르에 맞는 공통 서술 프롬프트를 구분한다. 이를 통해 AI가 동화의 서사 구조(기—승—전—결)를 자연스럽게 구성하고, 각 장르의 감정선과 메시지를 이야기 전개에 반영할 수 있도록 한다. 구현 방식 : 1.1 이미지 생성 (캐릭터 생성) - 사용자가 입력한 이미지 설명 한글 프롬프트 텍스트, OpenAI는 해당 프롬프트를 이미지 생성에 최적화된 영어 프롬프트로 변환한다. - '사람, 동물' 여부에 따라 STYLE_SUFFIX_ANIMAL, STYLE_SUFFIX_HUMAN 를 고정하고, 장르별로 NEGATIVE_PROMPT(부정), GENRE_PROMPTS(강화) 를 부여하여, 이미지 생성을 위한 영어 프롬프트에 장르별 분위기를 담을 수 있도록 조정한다. - ControlNet("lllyasviel/sd-controlnet-scribble") 을 통해서 사용자의 스케치를 추출하고, Stable Diffusion("Lykon/dreamshaper-8") 을 통해서 이미지에 배경을 보정한다. 그리고 애니메이션 이미지를 학습한 LoRA ("LarryXDrawjanimeScreencapStyle_v230epoch5") 모델을 사용하여 캐릭터의 일관성을 확보할 수 있도록 한다. - 이미지가 생성되면, AI 서버에서 이를 S3에 저장하고 S3 url을 백엔드에 반환한다. 1.2 동화 요약 기반 줄거리 생성 - 이전 줄거리/주인공 정보를 취리하여 FastAPI(AI 서버)에 줄거리 생성을 요청한다. - OpenAI가 정해진 대사, 현재 장면의 서사구조, 이전 줄거리, 주인공 정보, 장르 및 교육, 학생 지침 을 프롬프트로 받아 이를 기반으로 아이에게 질문을 던진다. - OpenAI가 정해진 지침, 현재 장면의 서사구조, 이전 줄거리, 주인공 정보, 장르 및 교육, 학생 지침 을 토대로 다음 줄거리를 생성한다. **2. 퀴즈** 기능 및 역할 : 스토리 줄거리를 기반으로, 4지선다 퀴즈를 생성한다. 모든 사용자는 지신과 타인이 생성한 스토리에서 퀴즈를 풀 수 있으며, 정답 개수에 따라서 칭호를 부여받을 수 있다. 구현 방식: 2.1 퀴즈 생성 - 사용자가 조회하고 있는 스토리의 현재 페이지 줄거리를 담아 FastAPI(AI 서버)에 퀴즈 생성을 요청한다. - OpenAI가 4지선다 JSON 형식으로 "question": "주인공이 들고 온 물건은 무엇인가요?","option1": "시계","option2": "책","option3": "우산","option4": "도토리","answer": "SOFTIGRAP" 2.2 칭호 부여 - 사용자는 퀴즈를 풀고, 맞은 정답 개수를 기반으로 예를 들어, 40개를 맞춘 경우 다음과 같은 +LEGENDARY_STARLIGHT(40, "전설의 반짝이게 별빛") 칭호를 부여 받을 수 있다. **3. AI 나레이션** 기능 및 역할: ElevenLabs를 해당 엔진으로 활용하여 부모의 음성을 복제하고, 반환값으로 voice_key 를 MySQL 데이터베이스에 저장하는 구조로 설계한다. 이를 기반으로 동화 텍스트를 실시간으로 내레이션한다. 구현 방식: 3.1 음성 복제 - 사용자가 제시된 프롬프트에 따라 감정을 담아 천천히 녹음한 음성 파일을 업로드하면, 서비스는 해당 파일을 POST <https://api.elevenlabs.io/v1/voices/add> 엔드포인트로 전송하여 ElevenLabs에 음성 복제(Voice Cloning)를 요청한다. - ElevenLabs에서 음성 복제가 성공적으로 완료되면 결과로 고유한 voice_key 가 반환되고, 이 값은 백엔드로 전달되어 MySQL DB에 저장된다. - 이야기 제들 읽다 특정 텍스트로 내레이션 재생을 요청하면, 서비스는 DB에서 해당 텍스트에 대응하는 voice_key 를 조회하여 ElevenLabs API에 전달하고, 해당 텍스트로 TTS 변환을 요청한다. 3.2 표정 나레이션 - 변환된 음성 데이터는 바이트 스트림 형태로 응답을 받고, 이를 ByteArrayInputStream 을 통해 클라이언트로 실시간 스트리밍한다. 내부적으로는 ByteArrayInputStream 을 사용해 오디오 바이트 배열을 일정 크기의 버퍼 단위로 읽고, 이를 OutputStream 으로 순차적으로 전송하여 클라이언트 측에서는 음성이 도착하는 즉시 재생한다. **4. 이스터에그** 기능 및 역할: 이미지 속 객체 탐지에 최적화된 YOLOv8 모델을 활용하여, 일러스트 내 특정 객체 좌표에 부모의 사랑이 담긴 음성 메시지를 숨겨둔다. 이러가 동화를 읽으며 그림을 클릭할 때, 해당 좌표가 감지되면 숨겨진 이스터에그(음성 메시지)가 재생되어 동이 요소처럼 즐길 수 있다. 구현 방식: 4.1 기술 속 슈퍼지 묘상 제시지 - 실시 이미지에 특화된 기존 YOLOv8 모델에 AI 일러스트 데이터셋 직접 수집/어셈블하여, 일러스트 기반 객체 탐지에 최적화된 파인튜닝 모델을 구축한다. # YOLO 모델 학습 코드 yolo task=detect mode=train model=yolo8n.pt data=data_path/data.yaml epochs=30 imgsz=416 - 사용자가 동화를 선택하면 해당 동화의 음성 분할 이미지가 Nginx를 통해 AI 서버로 전달되어 객체 탐지 요청이 이루어진다. - AI 서버는 화장의 전체 이미지를 대상으로 객체 탐지를 수행한 뒤, 탐지가 완료되면 무작위로 선택된 하나의 특정 객체의 좌표값(center_x, center_y)과, half_width, half_height 정보를 반환하며, 이 정보는 백엔드로 전달되어 사용자가 업로드한 음성 메세지와 함께 MySQL DB에 저장된다. - 이후 프론트엔드에서 클릭 이벤트를 통해 좌표 값을 전달받아, 사용자가 해당 영역을 클릭했을 때 브라우저 이스터에그(부모 음성 메시지)의 url을 반환하여 재생되도록 구현한다. 4.2 이스터에 배치되어 사용의 변화 - 사용자가 동화를 선택하고 작성한 편지와 공개 날짜를 등록하면, 시스템이 해당 시간을 기준으로 아이가 책을 펼쳐지 않을 때 사용자 편지가 자동으로 노출된다.

3.4 주요 기능의 구현

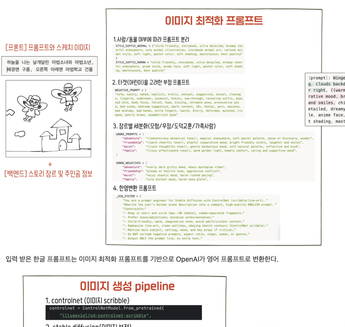
1. 동화 생성

1.1 이미지 생성 (캐릭터 생성)

동화 속 세계로 떠날 주인공을 만나볼까요?
이름
나이
성별
캐릭터
동화 장르
주인공 이미지
이야기 시작하기
이야기 시작하기


부인 '반짝이'가 만들어졌어요!
다시 만들기
이야기로 넘어가기

- 동화 생성 초기에 캐릭터 정보를 설정하고, 캐릭터 스케치와 이미지 한글 프롬프트를 입력하여 이미지를 생성한다.


[프롬트] 프롬프트와 스케치 이미지
하늘을 나는 날개달린 마법소녀와 마법모자 구름, 오로라 아래한 아방궁 건물
[백엔드] 스토리 장르 및 주인공 정보
이미지 최적화 프롬프트
1.사람/동물 여부에 따라 프롬프트 분리
STYLE_SUFFIX_ANIMAL = f"{style} friendly, storybook, ultra detailed, dreamy cinematic atmosphere, anime style, soft light, pastel color, soft shading, masterpiece, best quality"
STYLE_SUFFIX_HUMAN = f"{style} friendly, storybook, ultra detailed, dreamy cinematic atmosphere, anime style, anime face, soft light, pastel color, soft shadow, masterpiece, best quality"
2. 타깃연령대별 고려한 부정 프롬프트
NEGATIVE_PROMPT = {
"3-5y": "nudity, violence, explicit, scary, sexual, suggestive, sexual, crowd, cluttered background, complex, realistic, horror, ugly, deformed, distorted, extra limbs, bad anatomy, blurry, text, watermark, signature, cluttered background, complex, realistic, horror, ugly, deformed, distorted, extra limbs, bad anatomy, text watermark, signature, blurry, nudity, violence, explicit, scary, sexual, suggestive, crowd

3. AI 나레이션

3.1 음성 복제

- 부모가 자신의 목소리를 녹음하여 서비스에 업로드한다.
- Elevenlabs에 해당 부모의 목소리가 등록되고, 결과값으로 `voice_key` 가 반환되어 DB에 저장된다.

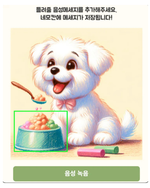
3.2 음성 재생

- 아이가 책을 읽다 원하는 목소리로 나레이션 재생을 요청한다.
- DB에서 해당하는 목소리의 `voice_key` 를 찾아 Elevenlabs에 요청하면 음성이 실시간으로 스트리밍된다.

---

4. 이스터에그

4.1 그림 속 숨겨진 음성 메세지



- 선택한 동화에서 탐지된 객체의 좌표를 시각화하고 음성메세지를 업로드한다.

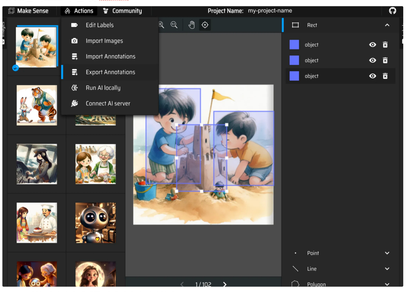
- 아이가 책을 읽다가 이미지를 클릭하고 해당 영역이 객체 좌표와 일치할 경우 저장된 음성 메시지의 url이 반환되어 자동으로 재생된다.

4.2 마지막 페이지의 사랑의 편지

- 사용자가 동화를 선택하고 작성한 편지와 공개 날짜를 등록하면, 시스템이 해당 시간을 기준으로 책을 끝까지 읽을 때 사랑의 편지가 자동으로 노출된다.

3.5 AI 상세설명

1. AI일러스트 수집 및 라벨링



- 수집한 이미지의 객체를 라벨링하여 저장한다.
- 배경과 명확히 구분되는 영역, 다른 객체와 크게 겹치지 않는 부분, 그리고 객체의 크기가 전체 이미지의 10% 이상 90% 이하일 때를 기준으로 라벨링을 수행한다.

2. YOLO 모델 파인튜닝

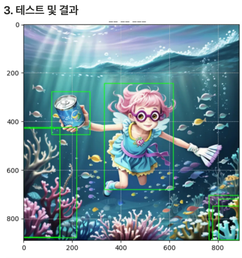
```
# YOLO 모델 학습 코드
!yolo task=detect mode=train model=yolov8n.pt data={data_path}/data.yaml epochs=30 imgsz=416
```

```
30 epochs completed in 2.501 hours.
Optimizer stripped from /content/runs/detect/train2/weights/last.pt, 6.2MB
Optimizer stripped from /content/runs/detect/train2/weights/best.pt, 6.2MB

Validating /content/runs/detect/train2/weights/best.pt...
Ultralytics 8.3.226 🚀 Python-3.12.12 torch-2.8.0+cu126 CPU (Intel Xeon CPU @ 2.20GHz)
Model summary (fused): 72 layers, 3,005,843 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% ━━━━━━━━━━━━ 1/1 4.3it/s 0.2s
                   all          1          3      0.987          1      0.995      0.908
Speed: 1.1ms preprocess, 149.3ms inference, 0.0ms loss, 1.1ms postprocess per image
Results saved to /content/runs/detect/train2
```

- 학습 시간이 오래 걸리는 단점을 해소하기 위해 입력 이미지 크기를 640에서 416으로 축소하여 최적화하였다. YOLOv8 모델은 입력 이미지 크기가 32의 배수일 때 가장 효율적으로 작동하기 때문에 이러한 설정을 적용하였다.
- mAP50(Mean Average Precision(IOU 0.5)IOU = 0.5 기준으로 모델의 전체 정확도 계산): 0.995 → 전체 정확성이 99.5%까지 증가한다는 점을 확인하였다.
- 30번의 학습(epochs)을 진행하여 그 중 가장 최적의 모델을 `best.pt` 로 저장하였다.

3. 테스트 및 결과




```
객체 1: {'center_x': 475.15606689453125, 'center_y': 462.9771728515625, 'half_width': 142.0509490966797, 'half_height': 218.82508850097656}
객체 2: {'center_x': 833.9075927734375, 'center_y': 819.7128295898438, 'half_width': 54.092407226562 5, 'half_height': 70.28717041015625}
객체 3: {'center_x': 826.0597534179688, 'center_y': 797.6727294921875, 'half_width': 61.94024658203125, 'half_height': 92.32723999023438}
객체 4: {'center_x': 195.31085205078125, 'center_y': 367.2190551757812 5, 'half_width': 79.10723876953125, 'half_height': 88.62989087128906}
객체 5: {'center_x': 109.59574890136719, 'center_y': 652.51171875, 'half_width': 109.59574890136719, 'half_height': 227.03884887695312}
객체 6: {'center_x': 75.29943004716797, 'center_y': 653.2989501953125, 'half_width': 75.29943004716797, 'half_height': 224.15999426269531 2}
객체 7: {'center_x': 846.058349609375, 'center_y': 804.5604248046875, 'half_width': 41.94168090820312 5, 'half_height': 83.60958862304688}
```

- 바인딩 박스로 객체가 탐지되면 시각화되도록 구현했다.
- 탐지한 객체의 모든 좌표값(`center_x, center_y`)과, `half_width, half_height` 정보를 반환하였다.