

Front-End Developer

항상 새로운 기술을 시도하는 개발자

박준영

Park Jun Yeong

1995. 02. 11

010 9947 5081

psw9502@naver.com

01 학력사항

2014. 02

평택 고등학교 졸업
2021. 02

단국대학교 화학과 졸업

02 경력사항

2020. 09 – 2022.02

코리아씨키트 기술부 근무
2022. 03 – 2023.04

대덕전자 기술부 근무
2024. 01 ~ 2024.12

삼성 청년 소프트웨어 아카데미 교육 이수

03 자격사항

- SQLD 자격증 취득 (2023.12.15)
- OPIC IM3 등급 취득 (2024.09.28)

03 수상 경력

2024. 07

자율 주행 알고리즘 경진대회(SSAFY RACE) 우승
2024. 11

자율 프로젝트 우수상 (SAPI 프로젝트)

04 SKILL

 PYTHON

 C++

 HTML

 JAVASCRIPT

 VUE.JS

 REACT

 FLUTTER

 DJANGO

 DART

 CSS

 SASS

 TAILWIND CSS

 MYSQL

SAPI

제작기간 24.10.14 ~ 24.11.19

구현 사항

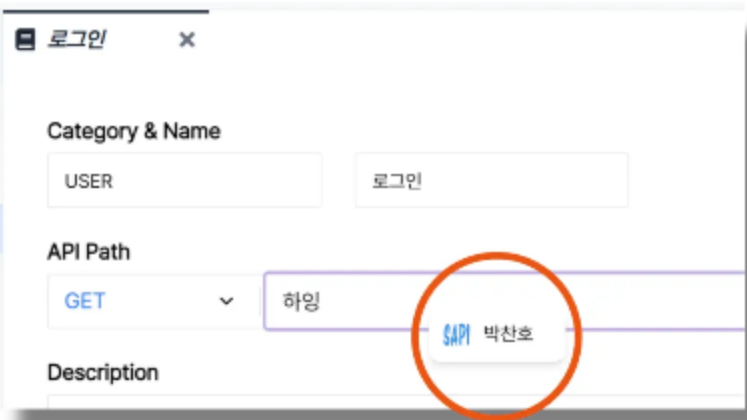
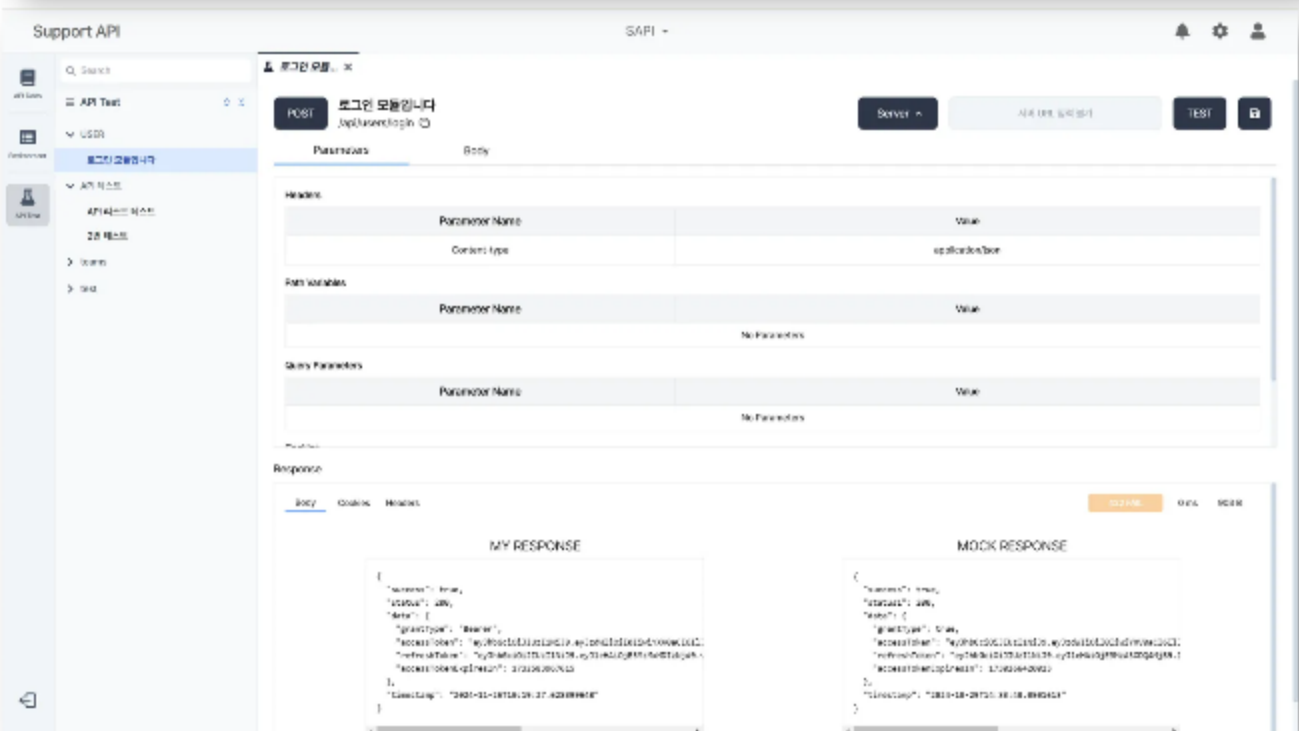
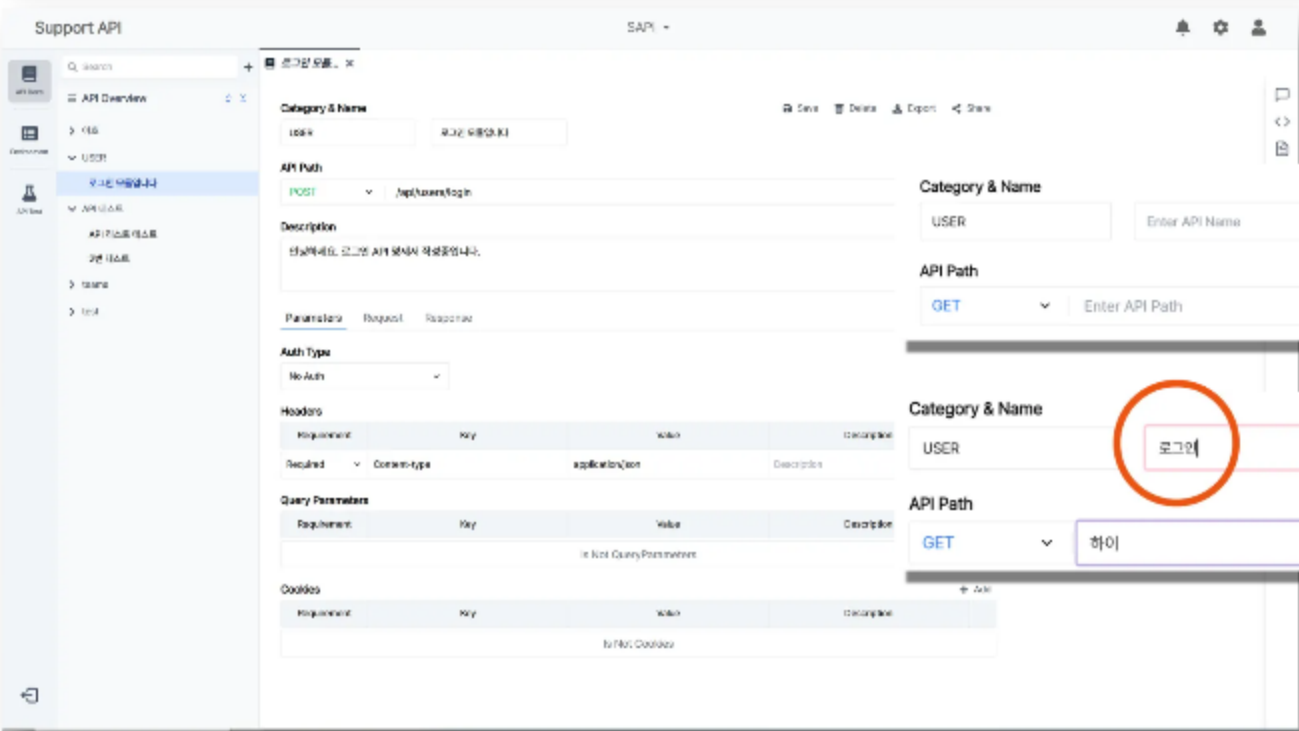
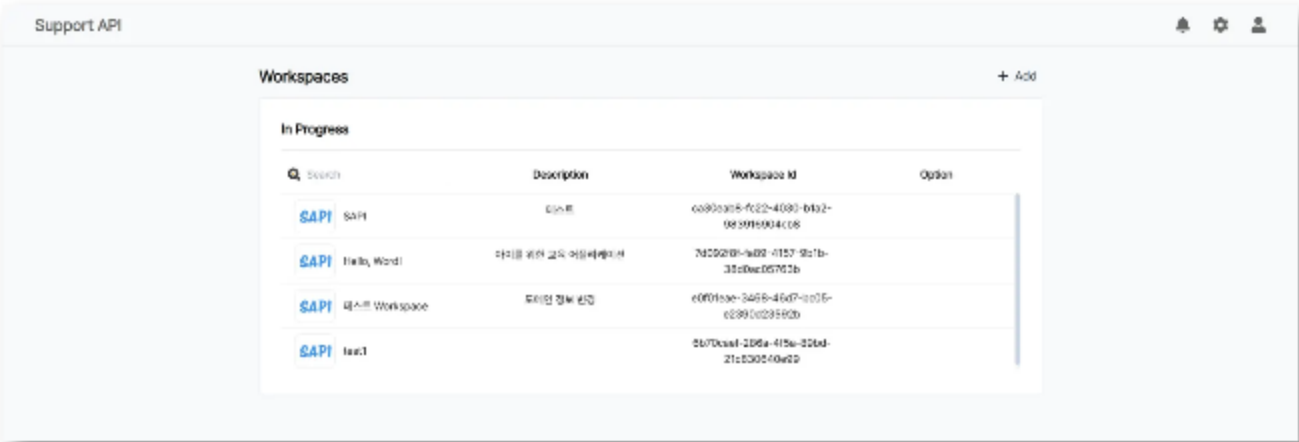
- React 를 활용한 Front 개발
- 웹 기반 에디터 실시간 편집 기능 개발
- API 명세서 작성, API 테스트(Front, Back) 기능 개발
- 코드 스니펫, 문서 Export, 문서 버전 관리 기능 개발

담당 역할

- WebSocket을 사용한 실시간 편집 기능 개발
- API 워크스페이스 기능 개발
- API 테스트 기능 개발

개발 환경

Front-End : React, WebSocket, Tailwind
Back-End : Java17, SpringBoot, MySQL, Redis, Lambda, WebSocket
Etc : EC2, Jenkins, Docker, Nginx, Gitlab, Jira, AWS S3



SAPI

트러블 슈팅

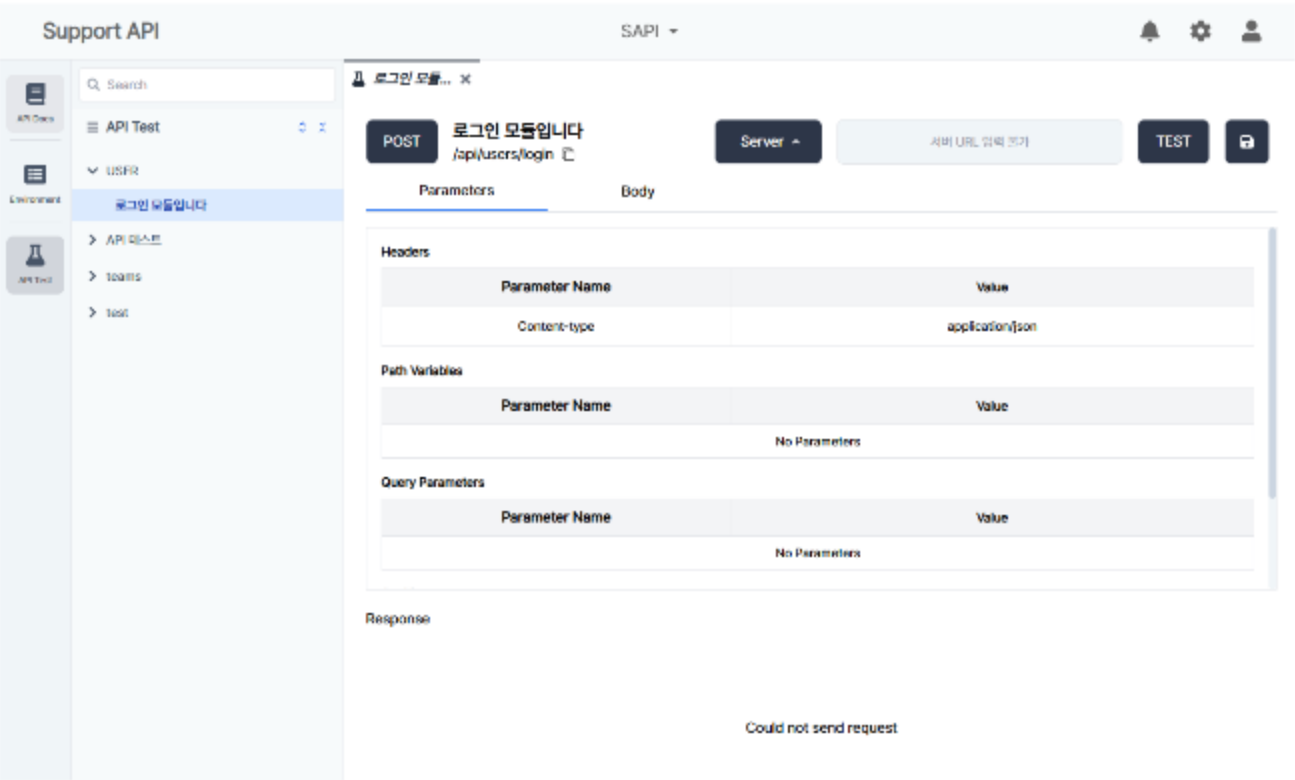
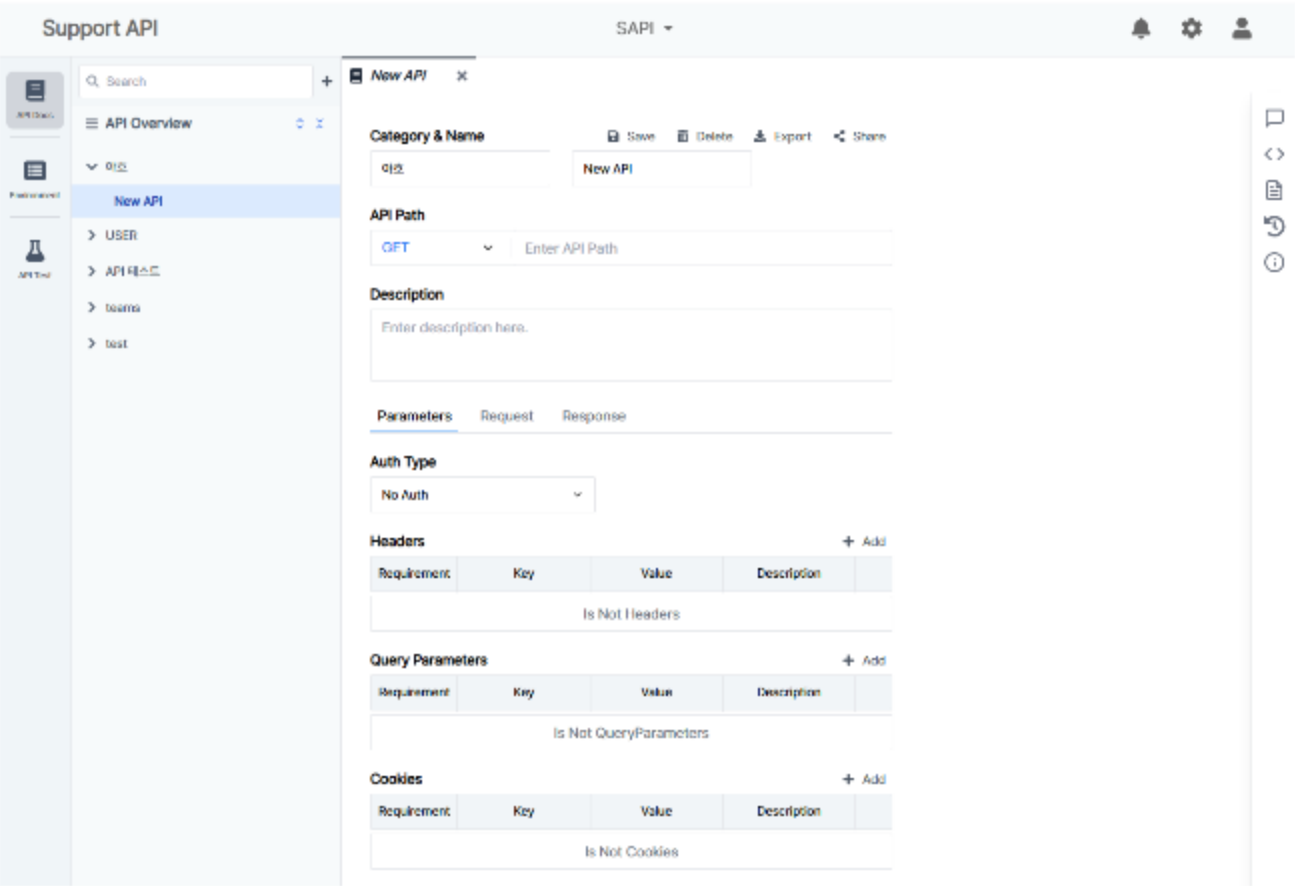
같은 문서를 여러 사용자가 동시에 편집할 때 편집 충돌 문제가 발생했습니다.이를 해결하기 위해 Redis를 활용하여 점유 상태를 관리하는 시스템을 도입하였습니다. 우선, useOccupationStatus 훅을 구현하여 다른 사용자의 점유 상태를 실시간으로 공유하도록 설정했습니다. 이를 통해, 특정 문서 또는 컴포넌트가 점유 중임을 Redis에 저장하고, 해당 상태를 클라이언트 간 동기화하도록 하였습니다. 사용자는 UI에서 점유 상태를 색상으로 즉각적으로 확인할 수 있게 하여, 다른 사용자가 작업 중인 영역임을 직관적으로 알 수 있도록 개선하였습니다.

성과 및 학습 내용

- Redis를 활용해 실시간으로 유저 점유 상태를 공유하고 관리하는 방식을 학습했습니다.
- WebSocket과 UI 동기화를 통해 협업 중 발생할 수 있는 편집 충돌 문제를 효과적으로 해결했습니다.

프로젝트 리뷰

이번 프로젝트에서는 UI를 최대한 직관적이고, 유저가 사용하기 편하게 만들기 위해서 노력했습니다. 다만, 너무 많은 기능들이 만들어지다보니 점점 UI가 복잡해짐을 느꼈습니다. 좋은 기능들은 많았지만 이런 개발 툴을 한번도 사용해보지 않았던 유저라면 사용하는데에 많은 어려움을 겪을 것 같아 프로젝트에 아쉬움이 남습니다.



Hello, Word!

제작기간 24.08.19 ~ 24.10.11

구현 사항

- React, PWA 를 활용한 Front 개발
- 학습 환경을 고려한 반응형 UI 디자인 개발 (Pad, Mobile)
- 단어 카드, 말하기, 짝맞추기, 동화 등 게임 로직 개발
- 아이 학습 통계 화면 구현

담당 역할

- 로그인, 회원가입 및 Auth 기능 개발
- 단어 카드, 동화 완성하기 게임 기능 개발
- Redux를 활용한 앱 상태 관리 기능 개발

개발 환경

Front-End : React, Javascript, Sass
Back-End : Java17, SpringBoot, MySQL
Etc : EC2, Jenkins, Docker, Nginx, AWS S3, Gitlab, Jira, Spark



Hello, Word!

트러블 슈팅

게임에서 이미지 로딩 시간이 길어져 사용자 경험이 저하되는 문제를 해결하기 위해 여러 가지 방법을 도입했습니다.

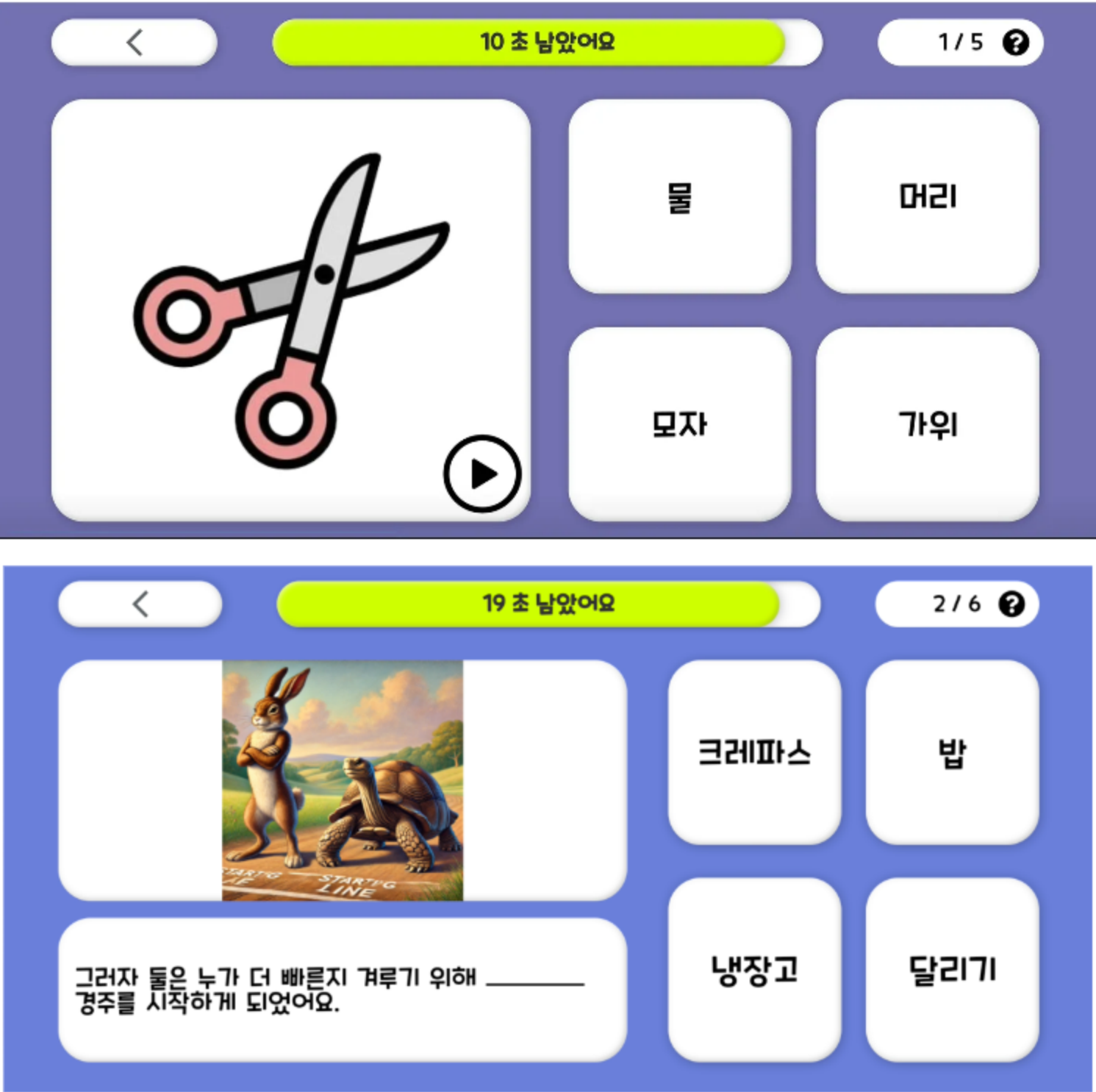
우선, 각 라운드에서 필요한 이미지를 미리 로드하는 프리로딩 기법을 사용하여 게임 진행 시 로딩 대기 시간을 최소화했습니다. 또한, 이미지가 완전히 로드된 후에만 다음 라운드로 넘어가도록 Promise 기반의 동기화 처리를 구현하여 화면 깜빡임 현상을 방지했습니다. 더불어, 이미지 크기 최적화와 WebP 형식과 같은 경량 이미지 포맷을 활용해 네트워크 전송 시간을 단축하였으며, Lazy Loading을 통해 현재 라운드의 이미지만 우선 로드하고 이후 라운드 이미지는 백그라운드에서 로드하도록 설정했습니다. 이러한 다양한 방법을 조합하여 성능 최적화와 원활한 사용자 경험을 동시에 구현할 수 있었습니다.

성과 및 학습 내용

- 이미지 프리로딩과 Lazy Loading을 통해 효율적인 로딩 방식과 사용자 경험 개선 방법을 학습했습니다.
- Promise 기반 동기화 처리로 비동기 작업을 안정적으로 관리하는 방법을 이해했습니다.

프로젝트 리뷰

처음으로 웹 개발을 통해서 게임을 구현해보았는데, 각 화면의 순서에 맞게 개발하고, 이미지 로딩과 같은 비동기적으로 처리되는 부분에 있어서도 시간을 맞추어야하기 때문에 많은 사항들을 고려해야만 했습니다. 단순한 UI를 구성하면서도 최대한 게임스럽게 진행될 수 있도록 많이 노력했던 프로젝트였던 것 같습니다.



Day-Log

제작기간 24.07.08 ~ 24.08.16

구현 사항

- Dart와 Flutter 를 이용한 Front 개발
- 캘린더, 그림일기, 설정화면, 알람 등 기능 개발
- WebRTC를 이용한 영상통화 기능 개발
- 영상통화 중 핫클립 영상 제작 기능 개발

담당 역할

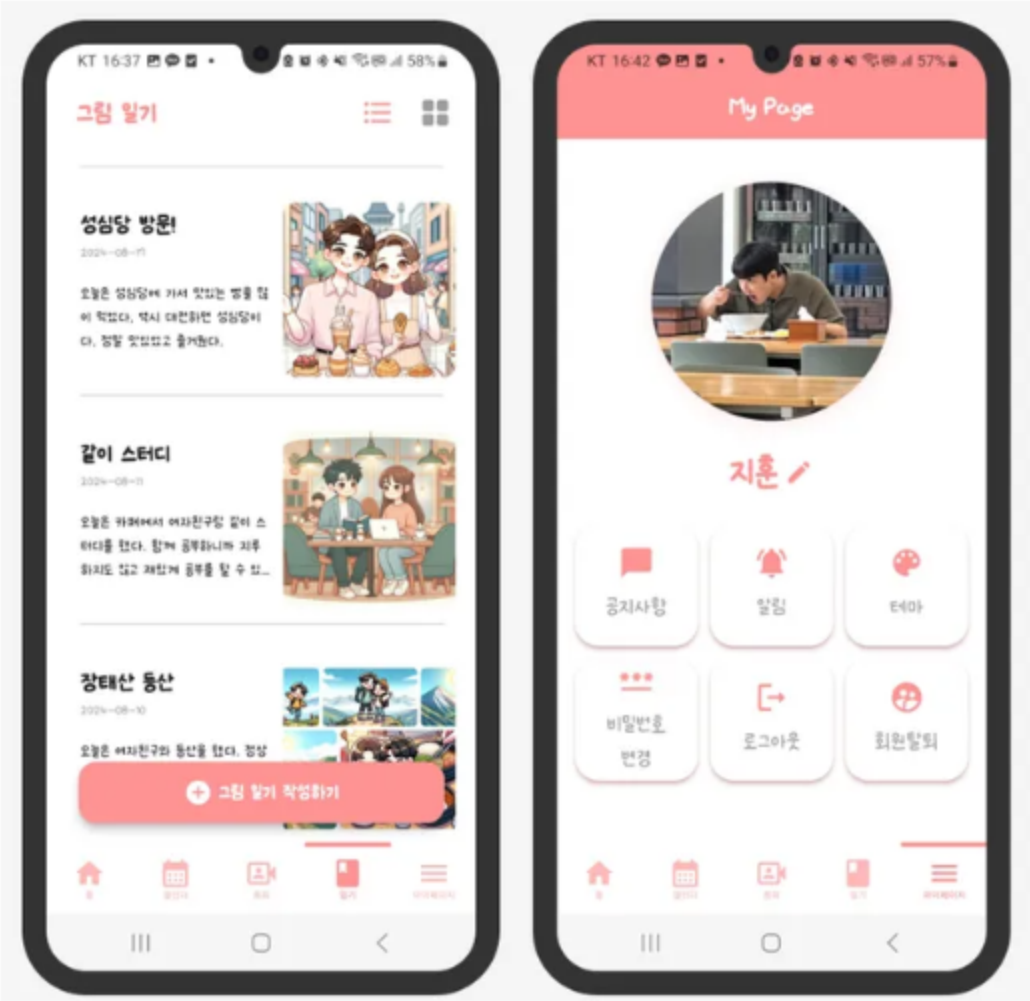
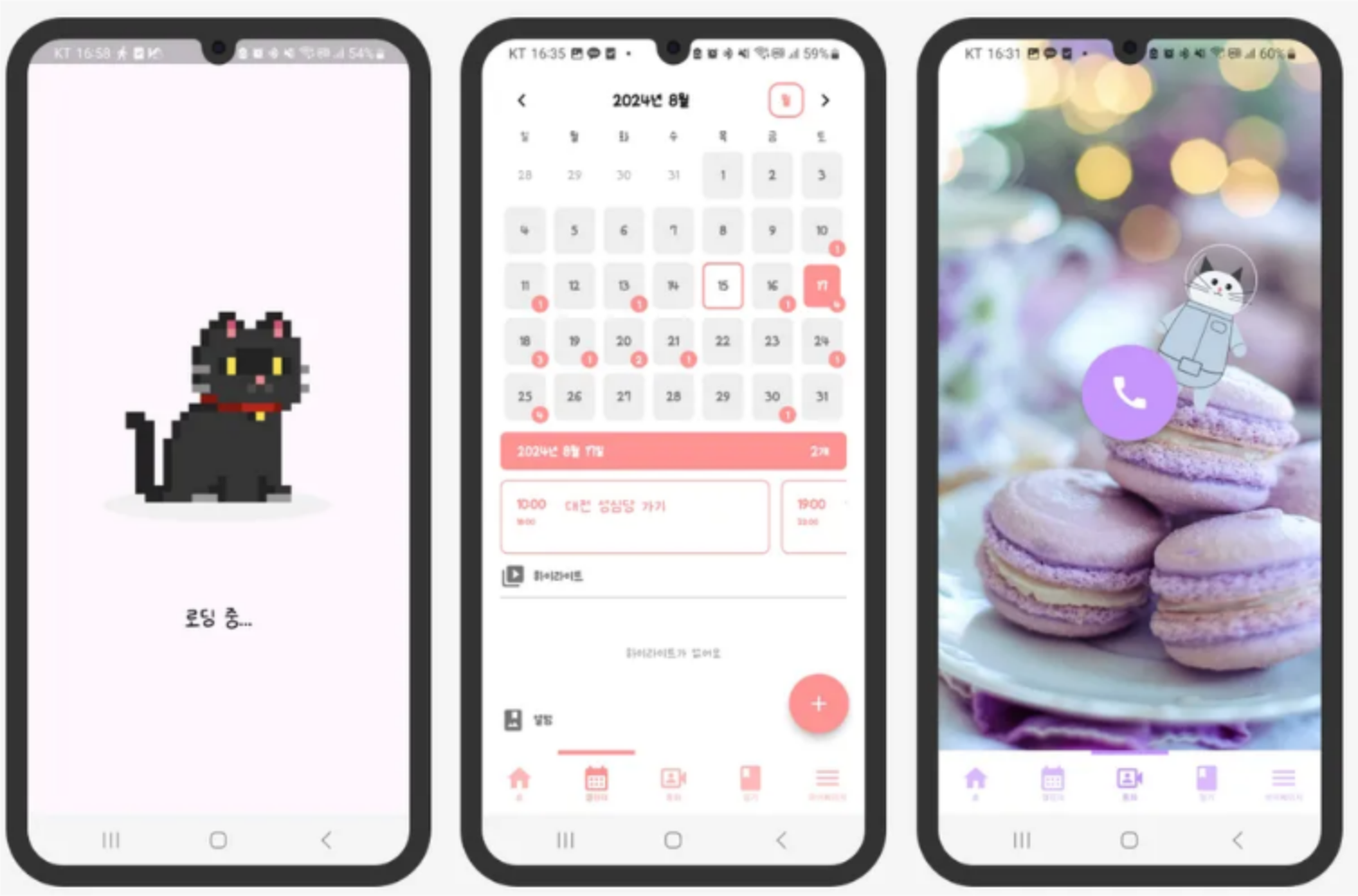
- OpenVidu를 활용한 영상통화 기능 개발
- FastAPI를 활용한 실시간 영상 편집 기능 개발
- Provider, Shared Preferences 앱 상태 관리 기능 개발
- Firebase를 활용한 유저 알람 기능 개발

개발 환경

Front-End : Dart, Flutter, WebRTC

Back-End : Java17, SpringBoot, MySQL, Python, FastAPI

Etc : Jenkins, Docker, Nginx, Ubuntu, Gitlab, Jira



Day-Log

트러블 슈팅

핫 클립 영상 제작 과정에서 이미지 파일과 사운드 파일의 싱크가 맞지 않는 문제가 발생하였습니다.

먼저, 각 파일에 타임스탬프를 기록하여 서버로 전달함으로써 정확한 동기화를 가능하게 했습니다. 서버에서는 FFmpeg의 `-itsoffset` 옵션을 활용하여 미디어 파일의 시작 시간을 미세하게 조정했고, 플래그를 추가해 두 미디어 파일 중 더 짧은 쪽의 길이에 맞춰 싱크를 조정했습니다. 추가로, 프레임 기반 영상 생성 시 `-r` 옵션으로 초당 프레임 수를 고정하고, 오디오와 비디오를 결합하는 과정에서 `-map` 옵션을 사용해 멀티트랙 동기화를 최적화했습니다. 최종적으로 FFprobe를 통해 오디오와 비디오의 타임스탬프를 분석하여 싱크 상태를 검증하며 문제를 해결했습니다.

성과 및 학습 내용

- FFmpeg를 활용하여 영상 프레임과 오디오 데이터를 동기화하는 기술을 학습했습니다.
- 비동기 기능을 사용해 실시간으로 데이터를 효율적으로 처리하는 방법을 익혔습니다.
- Provider와 같은 State를 관리하는 라이브러리를 학습했습니다.

프로젝트 리뷰

영상 데이터를 처리하는 과정 중에서 실시간으로 화면을 캡처하고 API로 전달하는 과정이 있었는데, 구현하는 과정 중에서 많은 문제가 생겼지만, 하나씩 해결하고 구현하는 과정이 좋았습니다.

다만, 데이터 통신 과정을 API 통신이 아닌 RTP 프로토콜로 적용했었다면 더 매끄러운 영상이 될 수 있었는데, 검토하지 못한 것이 아쉽습니다.



읽어주셔서
감사합니다 :))

PARK JUN YEONG

010 – 9947 – 5081

psw9502@naver.com