toPangyo팀 PMP

팀장 & 프론트엔드 우명규 프론트엔드 류나연 백엔드 선원종 백엔드 전혜지 백엔드 최성민

목차



01 팀목표

02 동세권

03 개인 목표

04 계획

*프로젝트 주제 선정 전 나눠본 각자의 목표

실시간 통신 로직 궁금

지도 api 사용 희망

프론트엔드

지도 및 위치 기능을 사용하고 가독성 있게 UI및 UX를 배치하며 기능이 오류없이 계획대로 구현되는 것

백엔드

이론적으로만 알고 직접 경험해보지 않았던 통신 소켓 프로젝트를 하는 것 지도를 통해 봉 사활동할 수 있 는 플랫폼 희망 -> 쉽게 봉사활 동에 다가갈 수 있도록 도와주 는 서비스 희망

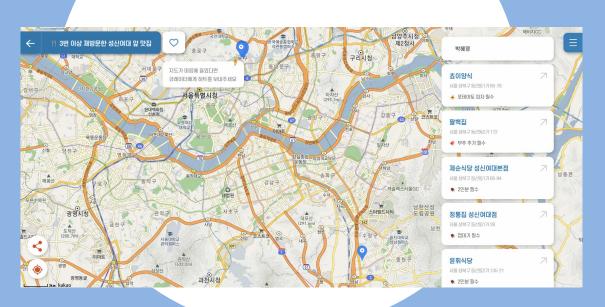
위치 데이터 궁금

=> 모두의 니즈를 충족해보자

디스코드를 많이 사용하 는 입장-> 채 팅 구현 경험 희망

진짜 서울

팀원들이 구현하고 싶었던 것들을 분석 => 지도api 기반의 플레이스 커뮤니티



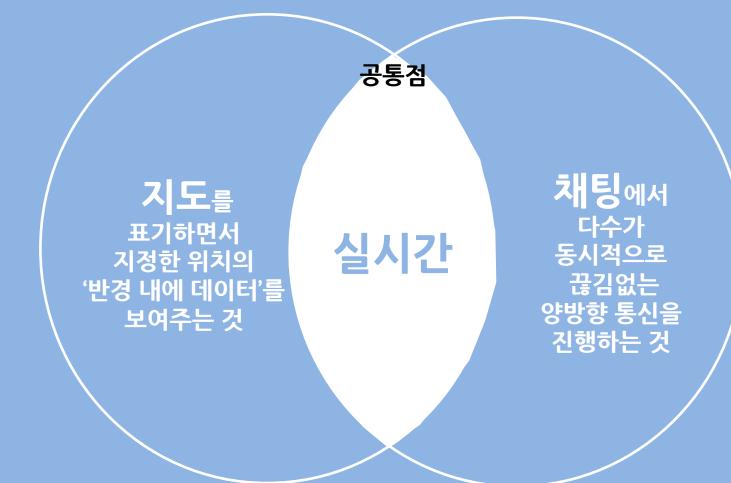
✓진짜 서울 단점 찿아보기

단순히 플레이스에 대한 코멘트와 그 코멘트에 대 한 의견만 남길 수 있을 뿐. 유저와 유저간 양방향 소 통이 안된다는 단점이 있 었음.

✓ 통신 및 채팅 기능 관심 O

팀원 중 통신 또는 채팅에 관심이 있는 사람이 있었음 또한 협업도구라는 공통 희망 주제 O

'진짜 서울' + 채팅 기능



실시간

페이지 렌더링 및 로딩속도

데이터 처리의 반응속도

통신의 안전성

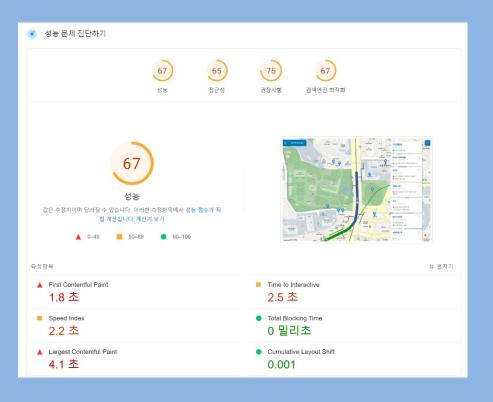
=> 페이지를 빠르게 렌 더링하거나 새롭게 그려 지는 요소들의 연산이 많으면 안된다고 생각 => 데이터를 빠르게 response할 수 있어 야함 => 소켓 통신을 할 때 끊김없이 상대방과 양 방향 통신을 할 수 있어 야함

Web Vitals에서 사용자 경험의 품질을 최적화하기 위한 지표로 LCP, FID, CLS 세 가지를 선정

그 중 실시간에 맞게 로딩 성능을 측정하기 위한 지표로 <u>3가지 요소 중 2가지 요소를 고려 가능</u>한 LCP를 목표로 선정

앱이 아닌 웹에서 사용자들이 로딩시간 및 대기시간이 길어지면 사용에 불편함을 겪고 사용자가 떠나게 됩니다. 구글 리서치에 따르면 웹 사이트의 로딩 시간이 3초 이상일 때 32%의 이탈률이 발생한다고 합니다. 팀 목 표

> ¹ LCP 2.5 초 이내



벤치마킹한 '진짜 서울'의 LCP 성능은 4.1s로 확인 우리는 조금 더 빠른 2.5초 이내의 LCP를 목표로 설정 팀 목 표

2

MSA 아키텍쳐 설계

안정성, 속도, 성능 때문이 아닌 팀원들이 다른 서비스를 개발해서 연동하는 경험을 해보기 위해





플레이스 매칭 커뮤니티 플랫폼

위치를 기반으로 하여 유저간 매칭을 통해 유저들이 플레이스를 직접 선택하여 함께 즐길 수 있는 콘텐츠 제공



우명규

1.MVC패턴 아키텍쳐 파악하기

- -리액트에서 사용되는 MVC 패턴의 아키텍쳐를 파악하여 페이지 설계하기
- -MVC의 단점으로 인해 생기는 Side Effect를 방지할 수 있는 방법 터득하기

2.백엔드와의 협업방법 flow 파악하기

-협업하는 방법을 터득하고 API, git 커밋메세지, To do list 등 PM에 필수적인 문서들 정의하는 법 알기

3.필요한 라이브러리 사용법 익히기

- -지도 API를 가져와 DB에 저장되어있는 데이터를 가공하여 처리하기
- -상태관리 라이브러리 Redux로 state를 관리하고 해당 컴포넌트에 필요한 데이터 추출하기



류나연

1.프로그램 설계 조직화 및 클린 코드 작성

- 재사용할 컴포넌트와 아닌 컴포넌트를 구별하며 프로그램을 설계 조직하기
- 구상도 작성 및 피드백하는 단계를 통해 이를 조직화하여 구분하기
- 코드 작성 후 최소 2회 확인하며 코드 내 오류, 실수 확인 및 어려웠던 점 회고

2.Redux, Scss, jwt 개념 배우고 정리 및 적용해보기

- 각 기능 또는 개념 별 최소 3개의 영상 또는 문서를 정독하기
- 그 후 이해 대한 정리 문서 또는 실습 파일을 만들기
- 해당 과정을 통해 선이해한뒤 이를 프로젝트에 적용하기

3.기록 습관 만들기

- 매일 to do list를 작성하며 기록을 몸에 익숙하게 만들기
- 해당 리스트를 바탕으로 일정을 진행하고 피드백하며 체계적인 습관 만들기
- 또한 커밋 양식에 맞춰 작성하며 협업을 위한 기록을 습관화하기



선원종

일회용 프로젝트가 아닌 지속적 기록과 문서화로 회고할 수 있는 나만의 지속가능한 자산을 만들자

_실제 서비스 되는 것까지 고려한 정교화된 설계 필요, 지속적관리

세부사항)

- 필드 명세설정, 관계설정을 생각하며 모델링을 작성하고 정규화 와 그 과정을 기록
- 매일 모닝 스크럽 회의 시 팀원들이 이해할 수 있을 정도의 설명 준비
- MSA를 모든 파트가 아닌 중요도가 높은 독립된 서비스가 필요한 곳에 적용해보기 (채팅 서버)
 - 팀원들과 작은것 부터 자주 소통하고 항상 긍정적인 모습 보여주기
 - 목표를 최대한 유지하지만 제 시간에 완성될 수 있게 구현하는것 (유연한 사고, 계획)

추가사항)

- 목표달성 위한 발전 지속적으로 기록하기 및 각 단계별로 회고 진행
- 최선을 다해 구현하고 리팩토링 진행, 부족한 부분은 지도를 요청해보기
- 적절한 검색법에 대한 이해와 좋은 자료가 있으며 정리해서 내 것으로 만들기 및 팀원 공유



전혜지

- 1.DB 쿼리 효율적이고 직관적으로 짤 수 있는 능력 갖추기.
 - 가독성과 성능을 위해 WHERE 절 FROM 절 순서 확실히 파악하여 작성하기.
 - 불필요한 조건 빼기, 검색 조건 간단히 작성하기.
- 2.git을 통해 으로 프로그램 관리하는 능력을 갖추기.
 - 명령어 리스트 정리해 놓고 쓰면서 자연스럽게 외우기.
 - commit 메세지만으로 기능 확실하게 알 수 있도록 쓰기.
- 3.문서 작성하기
 - 동일한 오류가 다시 났을 때, 무의미한 검색 시간 줄이기 위해 오류에 관련한 문서 작성하기.
 - 전체적인 개발 진도 파악 위해 진행 정도, 상황 기록하기.



최성민

1.RESTful api 중심의 비즈니스 서버를 구성한다.

- api설계시 고려해야하는 컨텍스트, 네트워크에 효율적인 웹 api 설계법에 대해 서치한다.
- 이를 바탕으로 "동세권"의 서버를 설계한다.
- 이러한 설계가 적절한지에 대해 검토해본 후 멘토분들의 피드백을 적극적으로 활용한다.

2.배포 과정 파악과 아키텍쳐 설계, 구현하기

- 프로젝트에 대한 웹 서버 배포 과정을 파악한다. (Nginx, 도커, aws등에 대한 역할과 이론을 설명할 수 있는 상태까지 도달한다.)
- 내용에 대해 팀원들과 공유하며 이를 바탕으로 아키텍쳐를 설계한다.
- 설계된 아키텍쳐가 합당한지에 대한 피드백을 진행한다.
- 검색과 멘토분들을 적극적으로 활용하여 해당 설계에 대해 피드백받는다.
- 아키텍쳐의 설계는 MSA아키텍쳐를 지향한다. (이때 서버를 나누는 단위, 서버간 통신 등에 대해 자세히 파악한다.)

3.철저한 계획에 기반한 스키마 모델 구축

-현재 익숙하지 않은 RDBMS와 비교적 익숙한 NoSQL DB를 고루 생각하여 목적에 맞는데이터베이스를 구축한다.

rule:

- 각 서치의 단계는 하루를 넘기지 않는다.
- 서치한 것에 대해서는 꼭 개인 개발 과정에 기록하며, 기록은 flow차트 형식을 지향한다.
- 설계 및 기능 구현등에 대해 정리가 완료된 항목에 대해서는 즉각적으로 팀원들과 공유한다.





개발계획



감사합니다.