최종 프로젝트 : Vue + PWA 앱 제작

개요

- 이제까지 배운 내용들을 (PWA, Vue, Firebase, Webpack) 활용하여 앱을 제작
- 제작할 앱의 UI 화면과 기술, 개략적인 구조를 소개
- 정부 데이터 포털과 Naver Open API 를 활용하는 방법을 소개

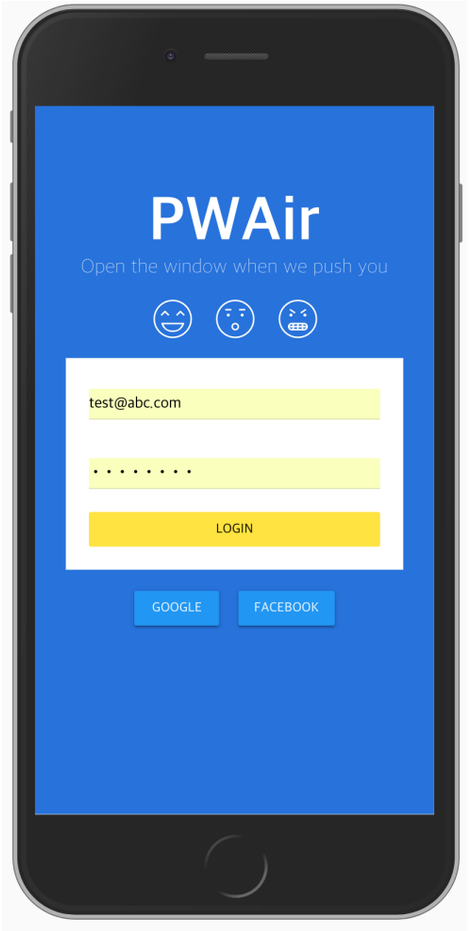
앱 소개

- 애플리케이션 명 : PWAir
- 구현 목적 : 사용자의 위치 정보를 이용하여 실시간 대기 지수 (미세먼지 등) 확인
- 상세 기능 :
    - HTML5 Geolocation API 를 활용하여 위치 정보 획득
    - 위치 정보로 해당 주소지 파악 (Naver Maps API, v3)
    - 주소지에 해당하는 대기 측정소의 실시간 대기정보 수신 (서울시 공공 데이터)
    - 그래프를 이용한 지난 대기 지수 확인 (Vue Chart)
    - Firebase DB 로 계정 별 푸시 정보 관리
    - 앱이 등록된 사용자에게 실시간 대기 정보 Push 전송 (PWA)

UI 화면 구성 - Vue.js

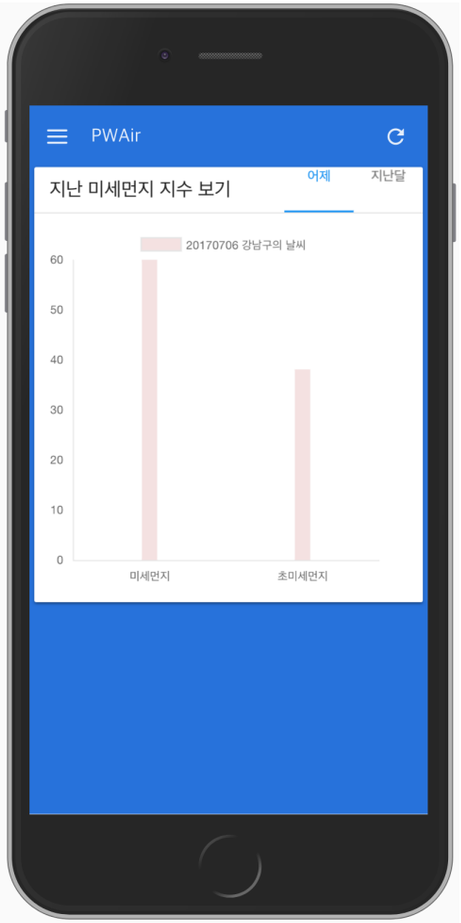
- Vue Development Flow (Single File Components)
- Vue Components Communication (Props, Event Bus)
- Vue Router (화면 전환)
- Vue Resource (Open API 데이터 수신 및 Push 알람 전송)
- Vue Chart (지난 대기지수 그래프로 표시)

UI 화면 #1 - 로그인


PWAir
Open the window when we push you
test@abc.com
LOGIN
GOOGLE
FACEBOOK

UI 화면 #2 - 실시간 대기 지수 확인

UI 화면 #3 - 이전 대기 지수 확인



HTML Geolocation 으로 위치정보 획득

- 사용자 현재 위치에 해당하는 위도 경도 획득

```
navigator.geolocation.getCurrentPosition();
```

Open API 활용 - 서울 데이터 광장

- 사이트 : 서울 열린 데이터 광장
- 사용 API :
  - 서울시 권역별 실시간 대기 현황
  - 서울시 일별 평균 대기오염도 정보
  - 서울시 월별 평균 대기오염도

Open API 활용 - 네이버

- 사이트 : Naver Maps Javascript API v3
- 사용 API :
  - 좌표로 주소 검색

Web App Manifest 등록

- install banner 및 splash screen 사용을 위한 기본 설정
- `gcm_sender_id` 지정하여 FCM Push 알람 설정

Service Worker 등록

- 웹 앱에 사용될 웹 자원들 모두 캐싱
- Push 메시지 수신 처리를 위한 Notification 구현

Firebase Auth, DB **활용**

- Firebase Auth 를 이용하여 사용자 계정 인증 (ID-PW, Google, Facebook)
- Firebase DB 를 이용하여 계정 정보 및 브라우저 키 저장
- Push 일괄 전송을 위한 node js script 활용

추가 구현사항

- http 통신으로 제공되는 open API -> https 용 open API 전환 필요
- Push 알람 전송을 위한 node js 코드에서 firebase db 데이터 조회 필요
- Push 알람 수신시 표시할 Notification UI & UX 커스터마이징 필요
- Offline Experience 를 위한 앱 구조 변경 필요
    - ex) Firebase 로그인 없이 바로 메인 화면 접속, API 캐싱 등

끝