

Vue.js

뷰 프레임워크로 PWA 만들기

[실습] PWA 테스트 및 배포를 위한 준비

<step1> node.js와 Vue-CLI가 설치

```
> npm install -g @vue/cli
```

<step2> Serve, firebase-tools 설치

- 작동 여부를 테스트하기 위해서 **Serve** 설치
- 웹 호스팅을 위해 **firebase-tools** 설치

```
> npm install -g serve  
> npm install -g firebase-tools
```

[실습] 프로젝트 폴더 생성 및 옵션 설정

<step3> 프로젝트 폴더 생성

```
> vue create [프로젝트명]
```

<step4> 프로젝트 옵션 설정

Please Pick a preset : [Manually select features]

Check the features : [Babel], [**Progressive Web App(PWA) Support**] 선택

Choose a version of Vue.js that you want to start the project with : [2.x]

Where do you prefer placing config for Babel, ESLint, etc : [In package.json]

Save this as a preset for future projects : [N]

[실습] 테스트 및 컴포넌트 설치

<step5> 뷰티파이 설치

```
project> vue add vuetify
```

Choose a preset : [vuetify 2 – Vue CLI (recommended)]

<step6> 실행 테스트

```
project> npm run serve
```

<step7> public 폴더에 manifest.json 파일 작성

[실습] 매니페스트 작성하기 (public - manifest.json)

```
{
  "name": "Hello, World~ PWA by VueJS!!",
  "short_name": "PWA by VueJS",
  "icons": [
    {
      "src": "./img/icons/android-chrome-192x192.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "./img/icons/android-chrome-512x512.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ],
}
```

[실습] 매니페스트 작성하기 (public - manifest.json)

```
"start_url": "./index.html",  
"display": "standalone",  
"orientation": "portrait",  
"background_color": "#FFFFFF",  
"theme_color": "#FFFFFF"  
}
```

[실습] 코드 수정

<step8> public – index.html 수정
이미지 사용, 화면 배치 등 다양한 내용 실습

[실습] public/index.html 파일 수정

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <!-- 상태 표시줄 테마 색상을 흰색으로 변경 -->
    <meta name="theme-color" content="#ffffff">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <link rel="manifest" href="manifest.json">
    <title>Hello, PWA by VueJS</title>
```


[실습] public/index.html 파일 수정

```
<!--머터리얼 디자인 아이콘 추가-->
```

```
<link rel="stylesheet"
```

```
href="https://fonts.googleapis.com/css?family=Roboto:100,300,400,500,700,900|Material+Icons">
```

```
<link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/@mdi/font@latest/css/materialdesignicons.min.css">
```

```
</head>
```

* 구글 머터리얼 아이콘 사용방법

> https://developers.google.com/fonts/docs/material_icons

[실습] public/index.html 파일 수정

```
<body>
  <noscript>
    <strong>We're sorry but ex08 doesn't work properly without JavaScript enabled.
Please enable it to continue.</strong>
  </noscript>
  <div id="app"></div>
  <!-- built files will be auto injected -->
</body>
</html>
```

[실습] src/App.vue 파일 수정

```
<template>
  <v-app>
    <v-main>
      <!-- fill-height는 브라우저 높이를 100%, 수직으로 가운데 정렬 -->
      <v-container fluid fill-height>
        <v-row>

          </v-row>
        </v-container>
      </v-main>
    </v-app>
  </template>
```

[실습] src/App.vue 파일 수정

```
<v-row>
```

```
<!-- text-center는 수평 가운데 정렬 -->
```

```
<v-col cols="12" class="text-center">
```

```
<!-- 타이포 스타일은 title, 글자색은 흰색으로 설정 -->
```

```
<h1 class="title white--text">Hello!</h1>
```

```
<p class="caption">by VueJS</p>
```

```

```

```
</v-col>
```

```
</v-row>
```

[실습] src/App.vue 파일 수정

```
<script>
export default {
  name: 'App',
  created () {
    // 배경색을 다크모드로 함
    this.$vuetify.theme.dark = true;
  }
}
</script>
```

[실습] 테스트 및 실행

<step1> 웹팩 모듈 설치,

(> npm install -- 필요한 노드 패키지 자동 설치, 필요시 설치)

<step2> 브라우저에서 결과 확인

- [CTRL] 키를 누르고 [Hosting URL] 주소를 클릭

> npm run serve -- 개발자 모드에서 웹 브라우저로 확인(서비스워커 동작 안함)

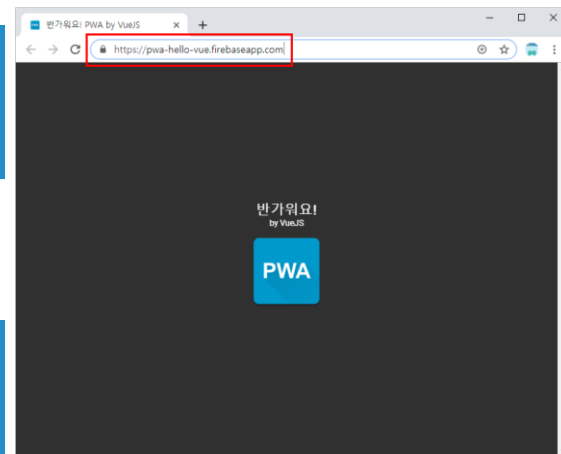
<step3> 빌드 완료

> npm run build -- 최종 빌드

<step4> 브라우저에서 결과 확인

- [CTRL] 키를 누르고 [Hosting URL] 주소를 클릭

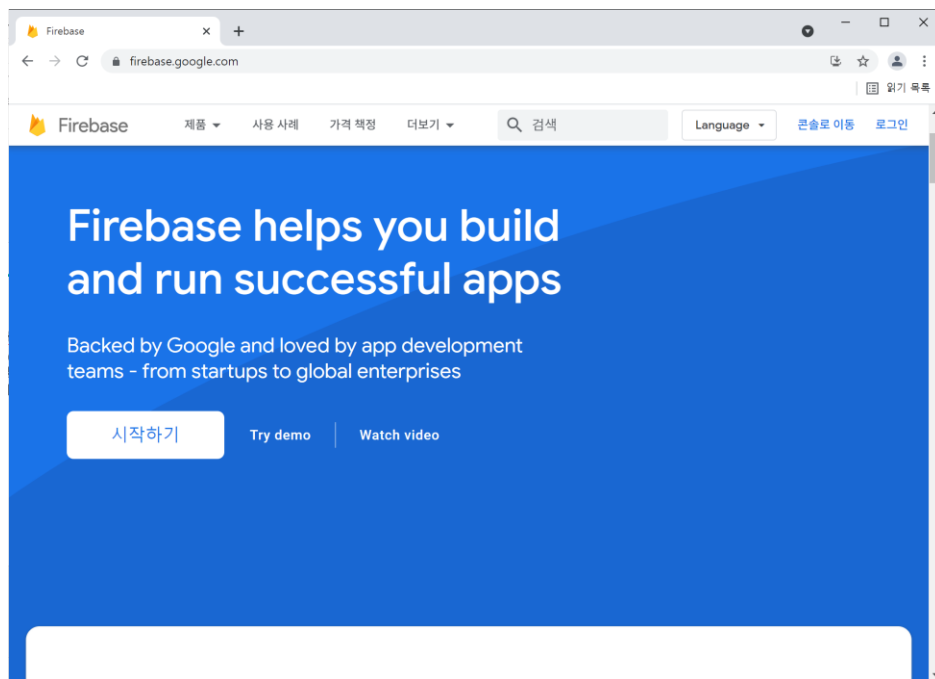
> serve dist -- 최종 결과를 웹 브라우저로 확인(서비스워커 동작 함)



3. 파이어베이스 호스팅에 HTTPS로 PWA 배포하기

파이어베이스란?

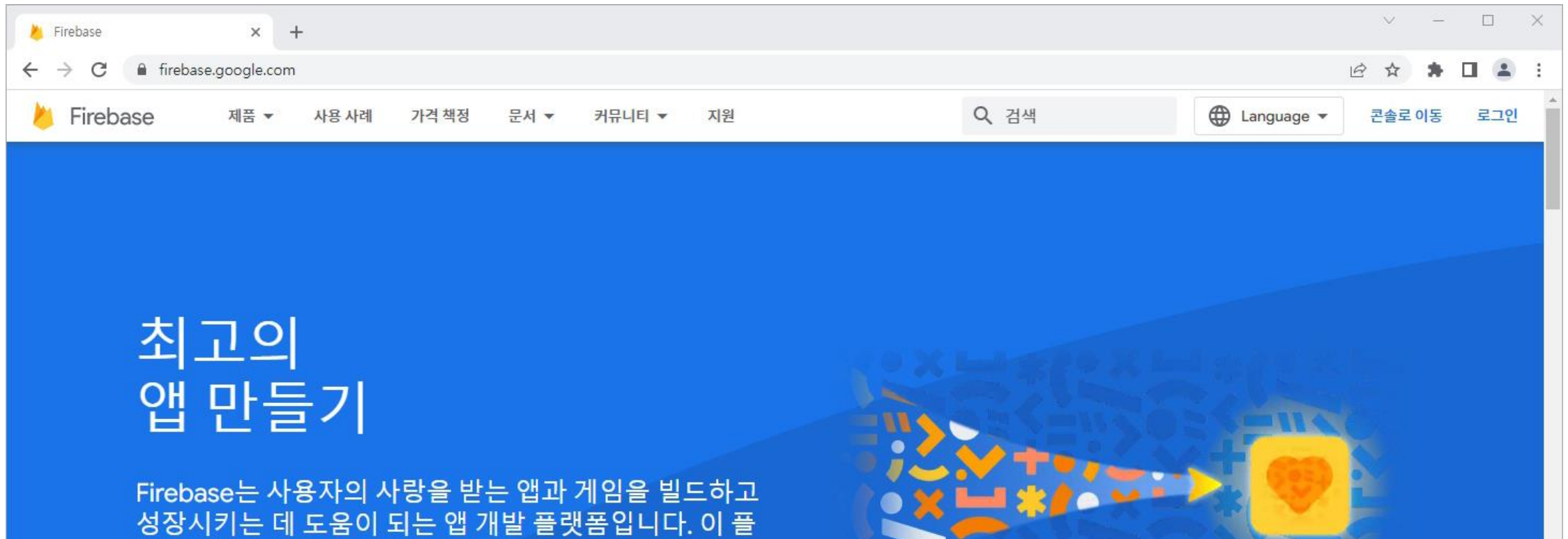
- 실전에 사용 가능한 높은 품질의 모바일 앱과 웹앱을 쉽고 빠르게 개발할 수 있도록 서버단에서 필요한 다양한 기능을 클라우드 서비스로 지원하는 구글의 통합 플랫폼
예) 실시간 데이터베이스, 오픈 인증, 스토리지, 호스팅, 푸시 알림 서비스, 광고, 분석 등



<https://firebase.google.com/>

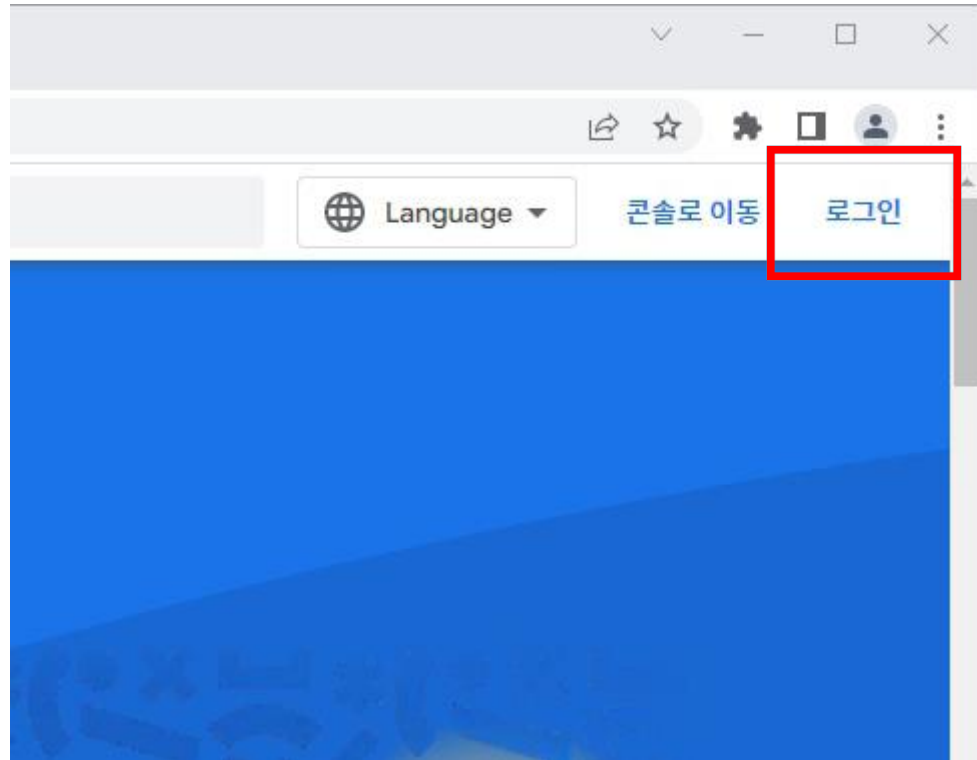
[실습] 파이어베이스 프로젝트 만들기

<step1> [http://firebase.google.com]에 접속



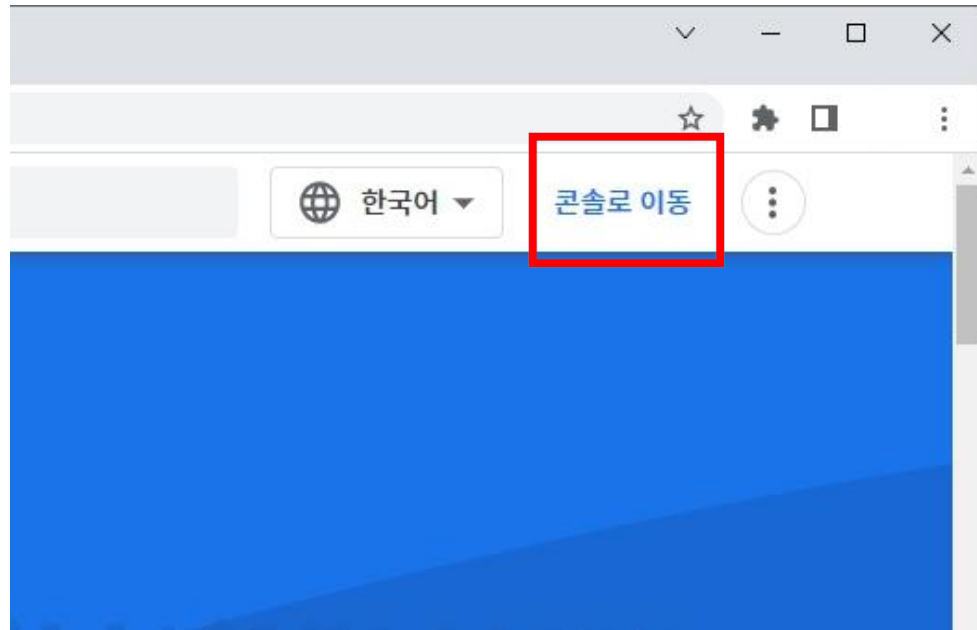
[실습] 파이어베이스 프로젝트 만들기

<step2> 오른쪽 상단 [로그인] 클릭하여 구글 계정으로 로그인



[실습] 파이어베이스 프로젝트 만들기

<step3> [콘솔로 이동] 클릭



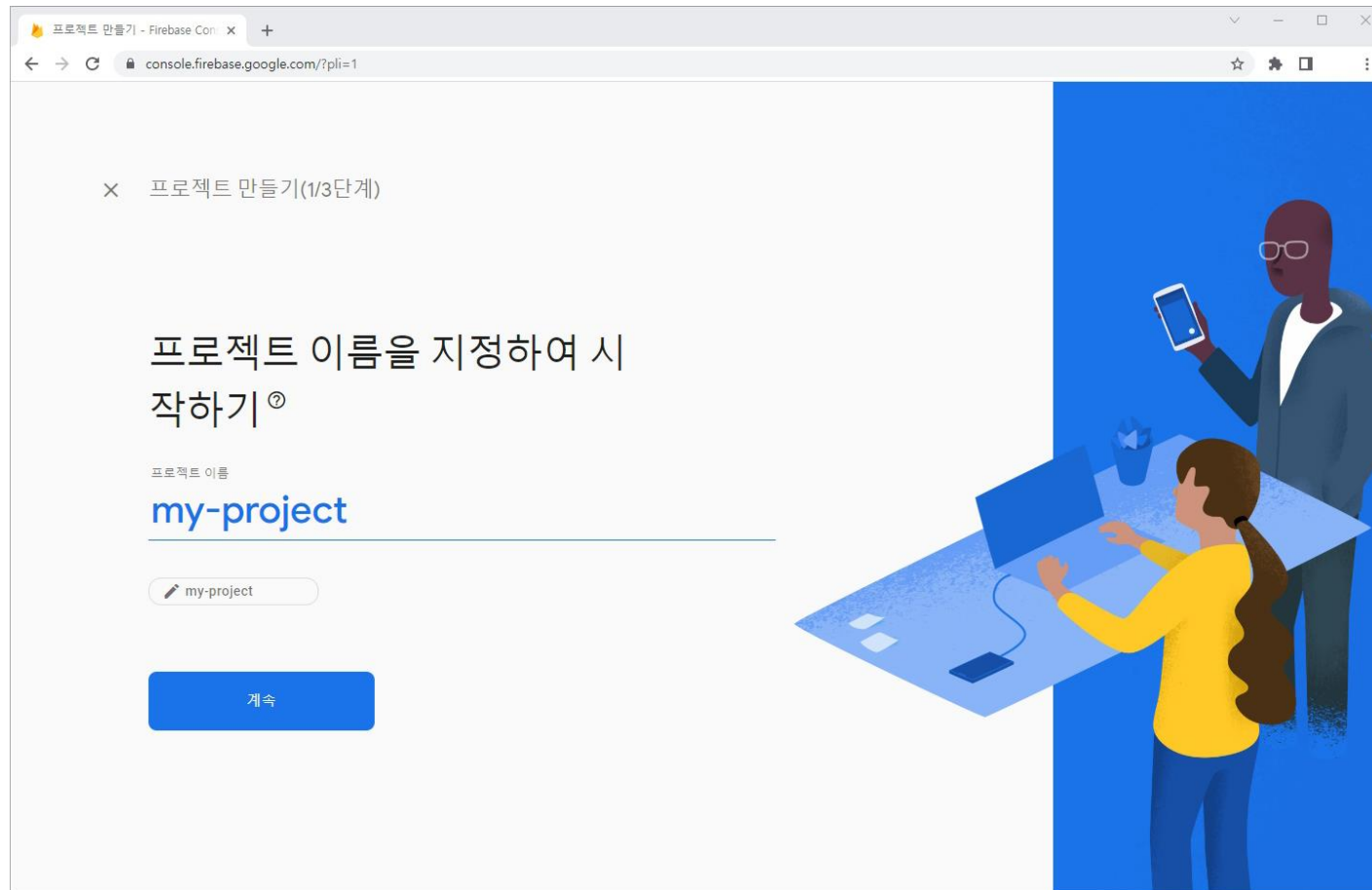
[실습] 파이어베이스 프로젝트 만들기

<step4> [프로젝트 추가] 클릭



[실습] 파이어베이스 프로젝트 만들기

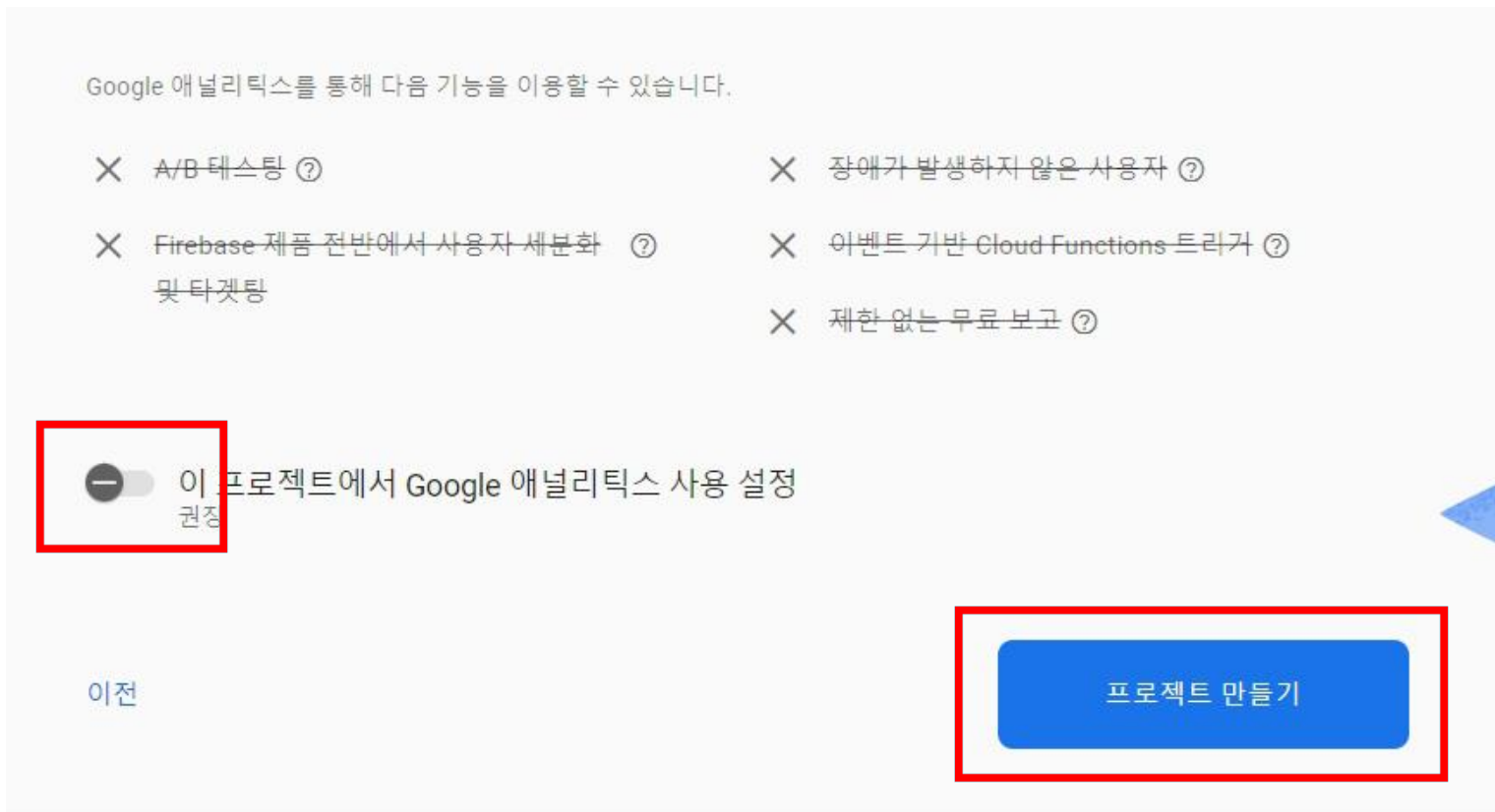
<step5> 프로젝트 이름 입력 후 [계속] 클릭



[실습] 파이어베이스 프로젝트 만들기

<step6> 연습용으로 사용할 계정이므로 구글 애널리틱스 해제

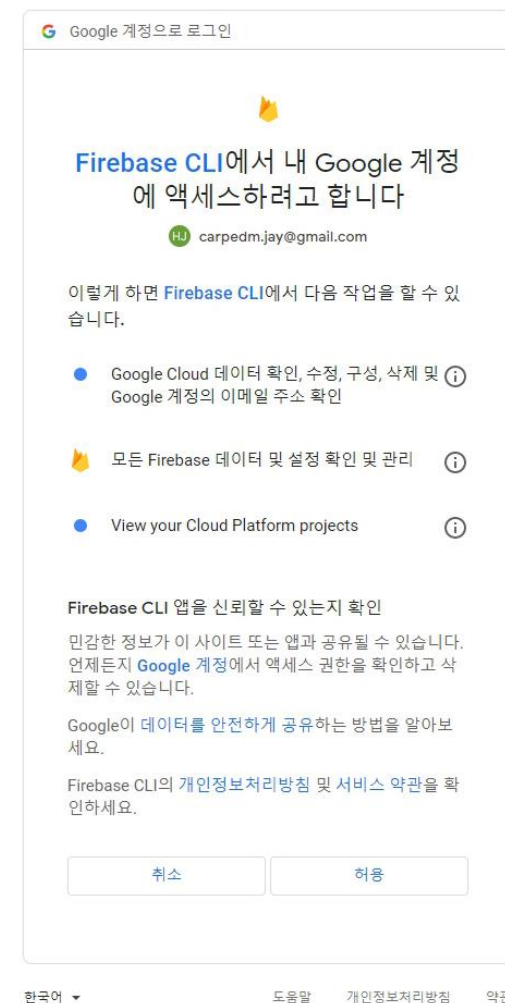
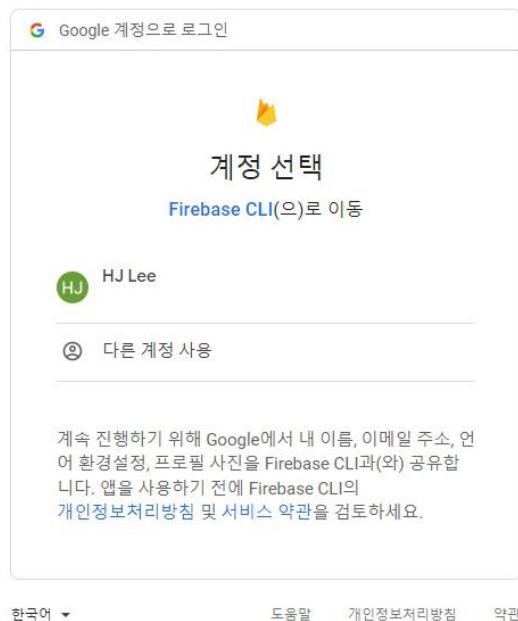
<step7> 프로젝트 만들기 클릭



[실습] 파이어베이스 프로젝트 만들기

<step8> 파이어베이스에 로그인

> firebase login



[실습] 파이어베이스 프로젝트 만들기

<step9> 파이어베이스 CLI가 내 계정에 액세스 허용

Woohoo!

Firebase CLI Login Successful

You are logged in to the Firebase Command-Line interface. You can immediately close this window and continue using the CLI.

[실습] 파이어베이스 프로젝트 만들기

<step10> 로그인 성공 시 콘솔에 성공 메시지 출력

+ Success! Logged in as " 자신의 계정 "

[실습] 파이어베이스 초기화

<step1> [firebase init]을 입력, 파이어베이스 초기화

```
> firebase init
```

<step2> 옵션 선택

Are you ready to proceed? [y]

Which Firebase features do you want to set up for this directory?
Press Space to select features, then Enter to conform your choices.

- () Realtime Database : Configure a security rules
- () Firestore : Configure security rules and indexes files for Firestore
- () Functions : configure a Cloud Functions directory and its files
- (*) Hosting : Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
- () Hosting : Set up GitHub Action deploys

[실습] 파이어베이스 초기화

<step2> 옵션 선택

Please select an option

- > Use an existing project
- > Create a new project
- > Add Firebase to an existing Google Cloud Platform project
- Don't set up a default project

<step3> 프로젝트 선택

Select a default Firebase project for this directory

- > "파이어베이스에 생성한 프로젝트 이름"

[실습] 파이어베이스 초기화

<step4> 기본 디렉토리 선택 : [dist] 입력 후 엔터

What do want to use as your public directory? **dist**

<step5> SPA 구성 : [N] 입력 후 엔터

Configure as a single-page app(rewrite all urls to /index.html)? **N**

<step6> 깃허브 자동 구성 : [N] 입력 후 엔터(실무에서는 필요에 따라 선택)

Set up automatic builds and deploys with GitHub? **N**

[실습] 파이어베이스 초기화

<step7> 초기화 완료

! Writing configuration info to ...

! Write project information to ...

+ Firebase initialization complete!

<Check> 초기화 코드 수정 후 다시 초기화 시 404 페이지, index 페이지 overwrite는 [N]

? File dist/404.html already exists. Overwrite? [N]

? File dist/index.html already exists. Overwrite? [N]

[실습] 최종 점검

<step1> node 패키지 미 설치시

```
> npm install
```

<step2> 소스 변경 시

```
> npm run serve
```

<step3> 배포용으로 빌드

```
> npm run build
```

<step4> 결과 확인

```
> serve dist
```

<step5> 이후 소스 변경 시 빌드만 진행

[실습] 파이어베이스에 업로드 및 HTTPS로 확인

<step1> 배포파일 생성

```
> npm run build
```

<step2> 파이어베이스에 업로드

```
> firebase deploy
```

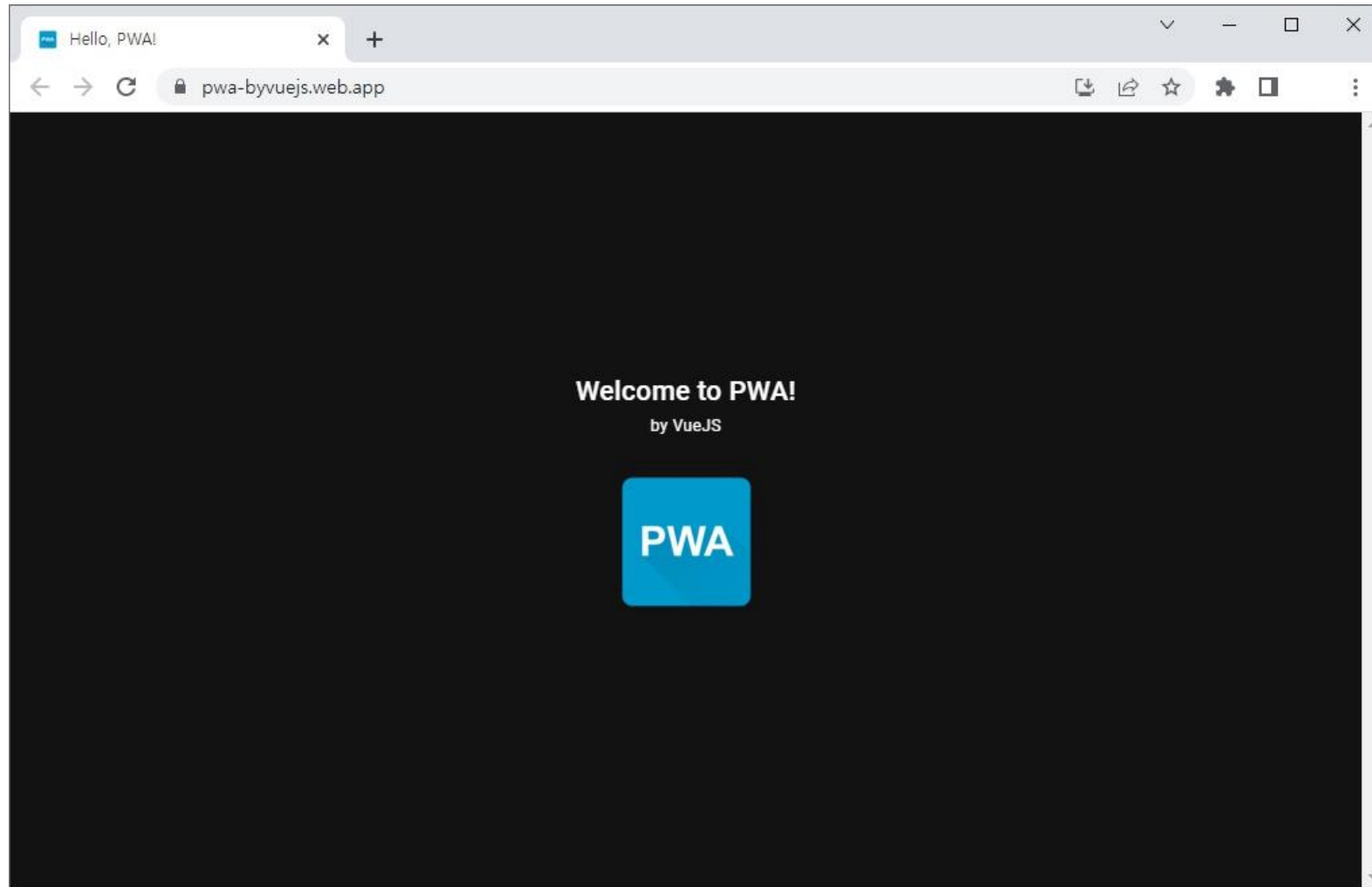
<step3> 브라우저에서 결과 확인

- [CTRL] 키를 누르고 [Hosting URL] 주소를 클릭

<step4> 코드 수정 시 build > deploy만 진행

```
> npm run build  
> firebase deploy
```

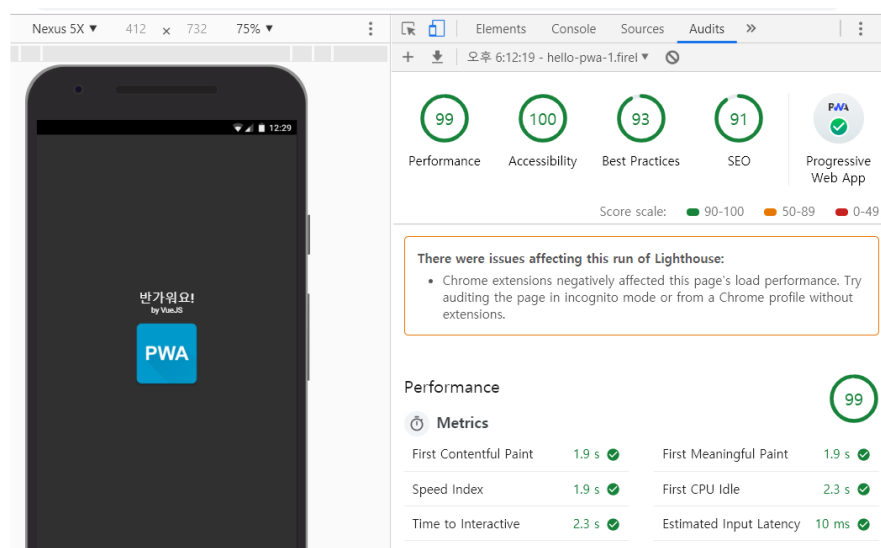
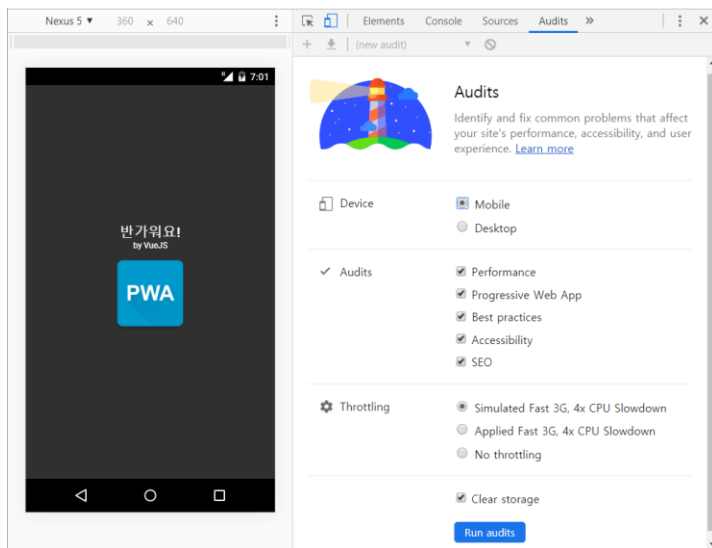
[실습] 파이어베이스에 업로드 및 HTTPS로 확인



4. PWA 성능 테스트하기

PWA의 성능을 테스트하는 방법

- 웹 성능, 웹 접근성, SEO 등
- 웹앱 매니페스트, HTTPS 같은 PWA 서비스 요건을 평가에 집중



| 크롬 개발자 도구의 라이트하우스

라이트하우스 테스트 기능

- 크롬 브라우저의 개발자 도구 제공
- 라이트하우스 기능을 그대로 탑재
- 웹 사이트의 성능, 접근성, 사용자 경험, PWA 기능이 어느 정도 되는지 정밀하게 분석하고, 보완해야 할 부분을 보고서 형태로 제공

| 크롬 개발자 도구의 라이트하우스

라이트하우스 테스트 기능

