

# 자기주도 PJT

Firestore 로 로그인기능 5Line 구현하기

## 목차

1. 과제 개요.....	3
2. 과제 목표.....	4
3. 필수 지식 학습.....	5
4. 기능 명세.....	6
5. 산출물 제출.....	11

## 1. 과제 개요

---

본 과제는 구글의 클라우드 자동화 환경 Firebase 를 이용하여 클라우드 인프라 환경의 경험을 해보고 더 나아가 권한관리, 로그인연동을 해봄으로써 2 학기 과제를 수행하는데 어려움을 개선해주는 목적을 가지고 있다. 1 학기 교육 과정에서 이미 학습한 Vue.js 를 토대로 이메일 로그인, 소셜로그인(카카오,구글 등)을 구현하고, 토큰을 주고받는 과정을 학습함으로써 모든 프로젝트의 기본을 다지는데 도움이 될 것 입니다.

2 학기 프로젝트에는 회원기반의 프로젝트로 지금까지의 대부분의 프로젝트가 구성되어 있습니다. 이에 본 과제에서는 인프라 작업과 백엔드 작업을 줄여줄 수 있는 과정을 수행해보고, 구글의 최신 클라우드인 Firebase 를 사용하는 것이 실제 2 학기의 과정에도 포함되어 있습니다. 로그인 기능을 사용해보고 Firebase 와의 프로세스과정을 이해해야 하며, 로그인의 기본적인 토큰요청 방식과 보안성에 대해 학습할 수 있습니다.

## 2. 과제 목표

---

### 1) Firebase Vue(Javascript) 프로젝트 생성

Android , iOS , 웹, Unity, C++ 중 웹프로젝트로 진행 및 기본적인 세팅을 학습

### 2) Firebase Authentication 소셜 로그인 카카오, GOOGLE 연동

지원하는 로그인제공업체중, 이메일/비밀번호와 Google 을 사용설정하고  
환경설정을 학습

### 3. 필수 지식 학습

구글의 Firebase 를 학습하기 위해서는 클라우드 개념과 인프라, 백엔드의 개념을 알고 있어야 기본적인 기능을 사용할 수 있습니다. 로그인인 경우 Firebase 의 권한 관리 기능을 이해하여야 합니다.

#### API 서버의 이해와 구현을 위한 참고 자료

구분	제목	링크
이해	Firebase	<a href="https://firebase.google.com/?authuser=0">https://firebase.google.com/?authuser=0</a>
이해	Firestore	<a href="https://firebase.google.com/products/firestore?authuser=0">https://firebase.google.com/products/firestore?authuser=0</a>
이해	Functions	<a href="https://firebase.google.com/products/functions?authuser=0">https://firebase.google.com/products/functions?authuser=0</a>
이해	Login	<a href="https://firebase.google.com/products/auth?authuser=0">https://firebase.google.com/products/auth?authuser=0</a>
이해	Storage	<a href="https://firebase.google.com/products/storage?authuser=0">https://firebase.google.com/products/storage?authuser=0</a>
이해	hosting	<a href="https://firebase.google.com/products/hosting?authuser=0">https://firebase.google.com/products/hosting?authuser=0</a>

## 4. 기능 명세

### 1. 기능/과제 목록

Req.	Category
1	Firebase 프로젝트 생성 및 환경설정 세팅
2	Firebase Authentication 설정 및 로그인 연동

### 2. 기능/과제 상세

#### Req. 1. Firebase 프로젝트 생성 및 환경설정 세팅

Req. 1-1	프로젝트를 생성하고 config 코드를 프로젝트내에 입력하여 연동시킨다
기능 상세	<p>Firebase cli , npm 등 필요한 라이브러리를 설치한다          Firebase 앱에서 웹을 생성한다          생성된 앱에 대한 정보를 보고 config 및 설정코드들을 학습한다          프로젝트에 코드를 넣어보고 연동이 되는지 확인한다</p> <p>Main.js 에 다음과 같은 본인의 설정 코드를 추가한다          (콘솔창에서 직접 확인할 수 있다)</p> <pre>import firebase from 'firebase' var firebaseConfig = {   apiKey: "AIzaSyBjoGqgAS08Exxxxxxxxxxxxxxxx",   authDomain: "ssafy-be63a.firebaseio.com",   projectId: "ssafy-be63a",   storageBucket: "ssafy-be63a.appspot.com",   messagingSenderId: "197568576359",   appId: "1:197568576359:web:599ef9xxxxxc2",   measurementId: "G-3RC3JL9R9S" }; firebase.initializeApp(firebaseConfig)</pre>

#### Req. 2. Firebase Authentication 설정 및 소셜 로그인 연동

Req. 2-1	로그인 기능을 연동하여 Firebase 와 소셜 로그인을 학습해본다
기능 상세	로그인 제공업체의 Google, 이메일/비밀번호를 활성화 한다

#### 로그인 제공업체

제공업체	상태
이메일/비밀번호	✓ 사용 설정됨
Google	✓ 사용 설정됨
Facebook	✓ 사용 설정됨
Twitter	✓ 사용 설정됨

카구글 로그인 npm 을 설치해본다  
 npm i vue3-google-login

로그인 이벤트를 담당할 이미지버튼을 만들어본다

```

```

각 소셜사이트 별로 필요한 토큰, 설정작업의 코드를 작성해본다

버튼을 누르게되면 소셜사이트에 요청이 전달되게 한다  
 심화학습으로서 필요에 따라, 요청한뒤 토큰값을 받으면 다시 백엔드를 호출 할 수 있도록 코드를 작성한다

```
const googlelogin = () => {
  var provider = new firebase.auth.GoogleAuthProvider();
  firebase.auth()
    .signInWithPopup(provider)
    .then(async (result) => {
      /** @type {firebase.auth.OAuthCredential} */
      var user = result.user;
      const httpsCallable = await
firebase.functions().httpsCallable('auth');
      await httpsCallable({ uid:
user.uid,user:result.user.providerData[0] })
        .then(({ data }) => {

          console.log(data)
        })
        .catch(httpsError => {
          console.log(httpsError);
        })

      // ...
    }).catch((error) => {
      console.log("error", error)
    })
}
```

	<pre>}); };</pre>
--	-------------------

## 5. 산출물 제출

산출물 제출	<p>생성한 프로젝트를 제출한다.</p> <p>산출물 제출 가이드 : <a href="https://lab.ssafy.com/s10-study/self-project/">https://lab.ssafy.com/s10-study/self-project/</a></p> <p>주의사항 : 프로젝트를 git 에서 올려야 node_modules 제외</p>
--------	---