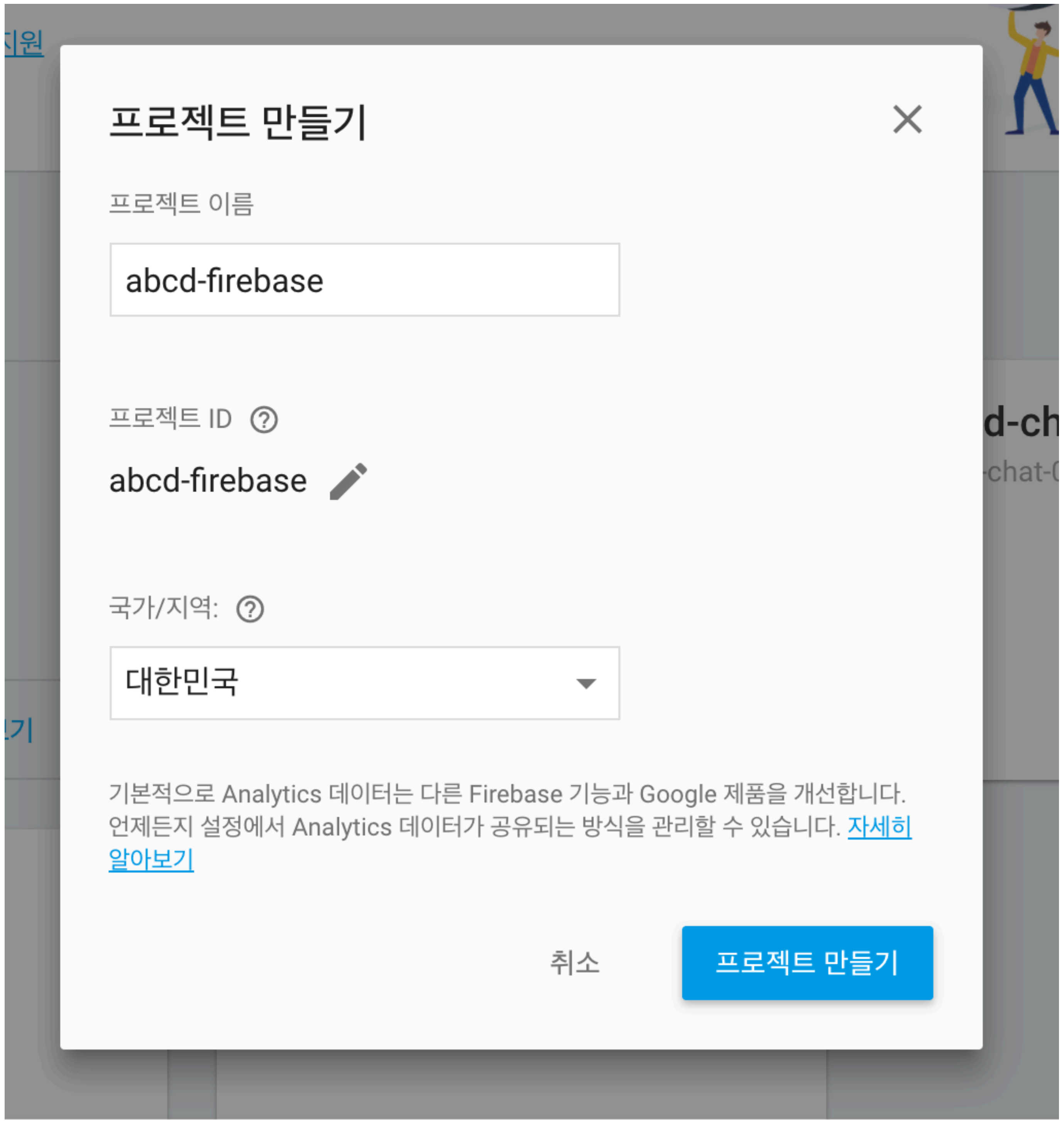
ABCD Firebase Study

https://github.com/misolab/abcd-firebase

TODO

- Project Create
- Hosting
- Authentication
- Realtime Database
- Memo App

Project Create



프로젝트 만들기

프로젝트 이름

abcd-firebase

프로젝트 ID

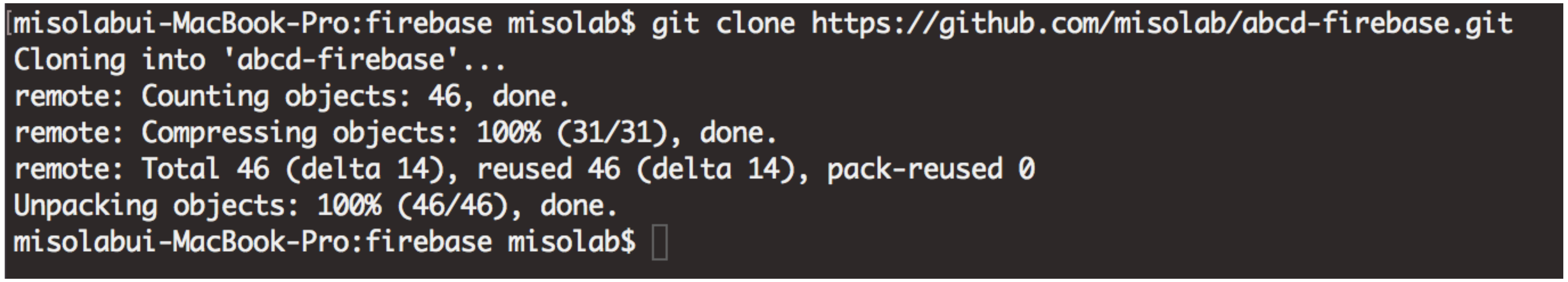
abcd-firebase

국가/지역:

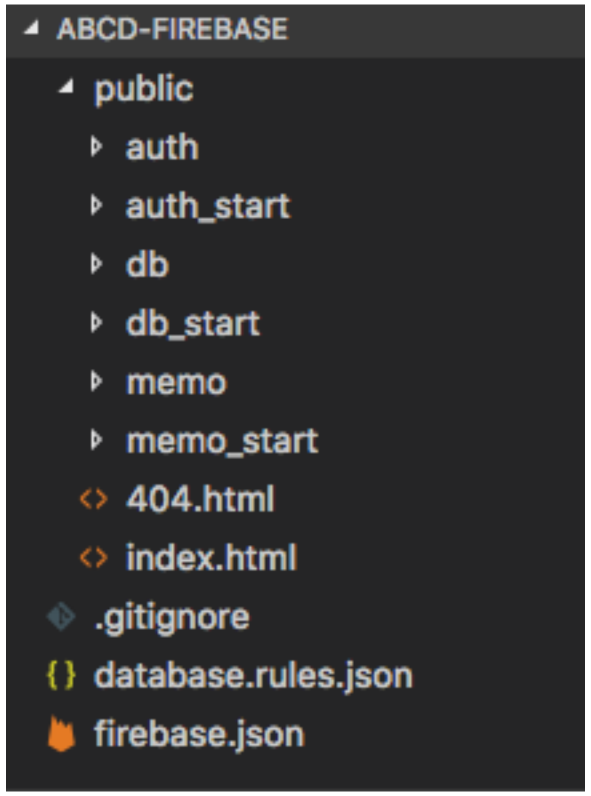
대한민국

기본적으로 Analytics 데이터는 다른 Firebase 기능과 Google 제품을 개선합니다. 언제든지 설정에서 Analytics 데이터가 공유되는 방식을 관리할 수 있습니다. 자세히 알아보기

git clone https://github.com/misolab/abcd-firebase.git



```
misolabui-MacBook-Pro:firebase misolab$ git clone https://github.com/misolab/abcd-firebase.git
Cloning into 'abcd-firebase'...
remote: Counting objects: 46, done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 46 (delta 14), reused 46 (delta 14), pack-reused 0
Unpacking objects: 100% (46/46), done.
misolabui-MacBook-Pro:firebase misolab$
```



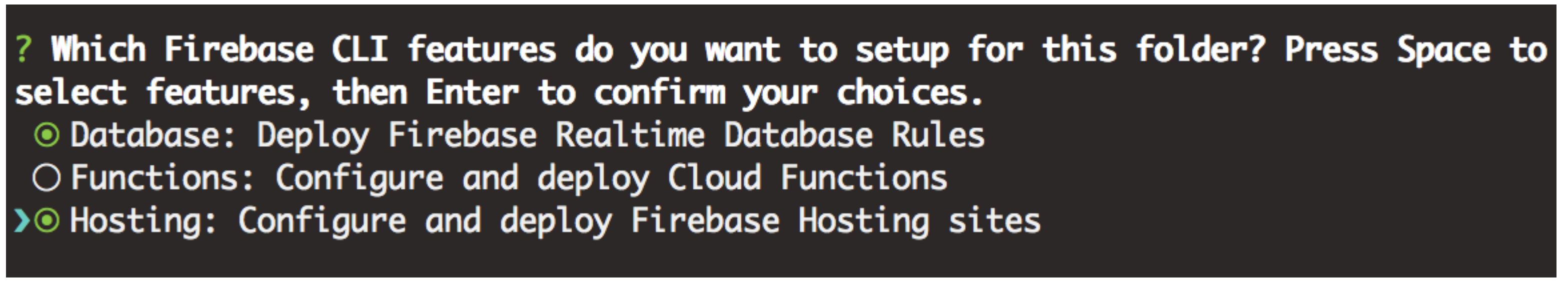
```
ABCD-FIREBASE
  public
    auth
    auth_start
    db
    db_start
    memo
    memo_start
    404.html
    index.html
  .gitignore
  database.rules.json
  firebase.json
```

<u>실습 파일</u>
auth : 인증
db : 실시간DB
memo : 메모앱
(*_start는 실습용)

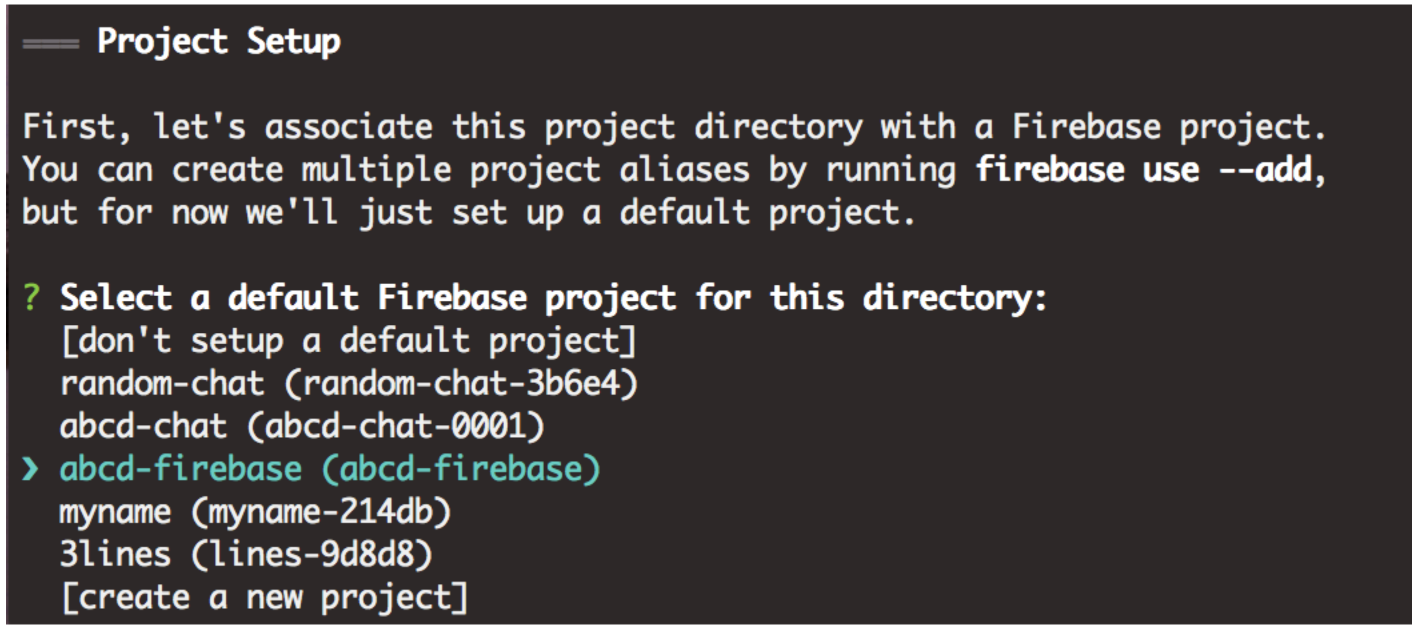
<u>firebase에서 만든 파일</u>
index.html
404.html
***.json : 설정파일**

Hosting

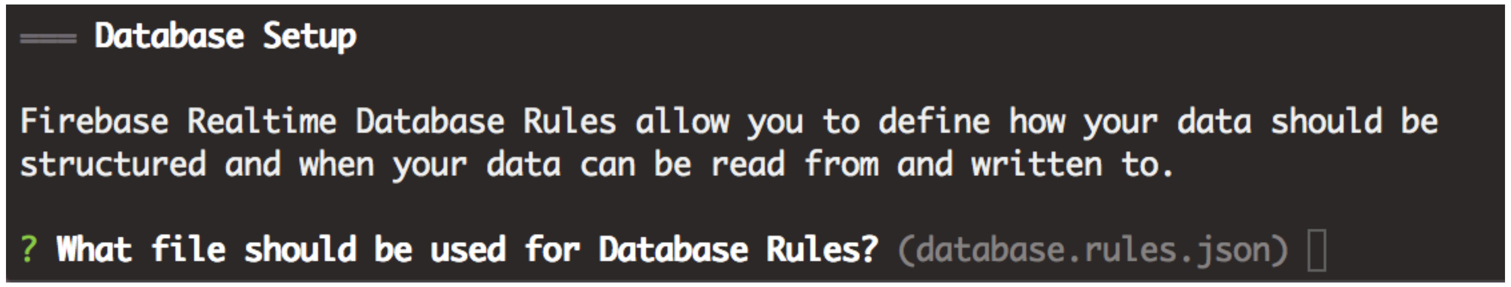
firebase init



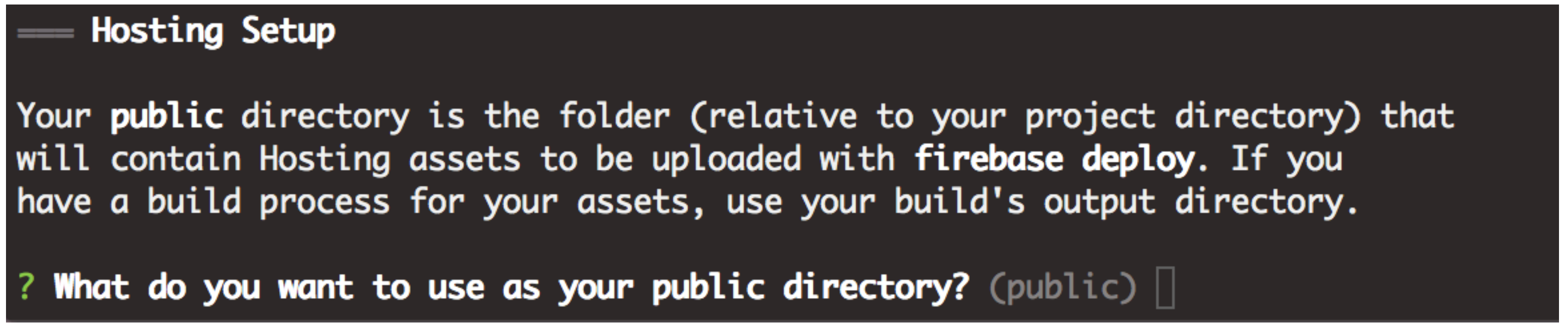
```
? Which Firebase CLI features do you want to setup for this folder? Press Space to
select features, then Enter to confirm your choices.
 ◉ Database: Deploy Firebase Realtime Database Rules
 ◯ Functions: Configure and deploy Cloud Functions
❯◉ Hosting: Configure and deploy Firebase Hosting sites
```



```
=== Project Setup

First, let's associate this project directory with a Firebase project.
You can create multiple project aliases by running firebase use --add,
but for now we'll just set up a default project.

? Select a default Firebase project for this directory:
  [don't setup a default project]
  random-chat (random-chat-3b6e4)
  abcd-chat (abcd-chat-0001)
❯ abcd-firebase (abcd-firebase)
  myname (myname-214db)
  3lines (lines-9d8d8)
  [create a new project]
```



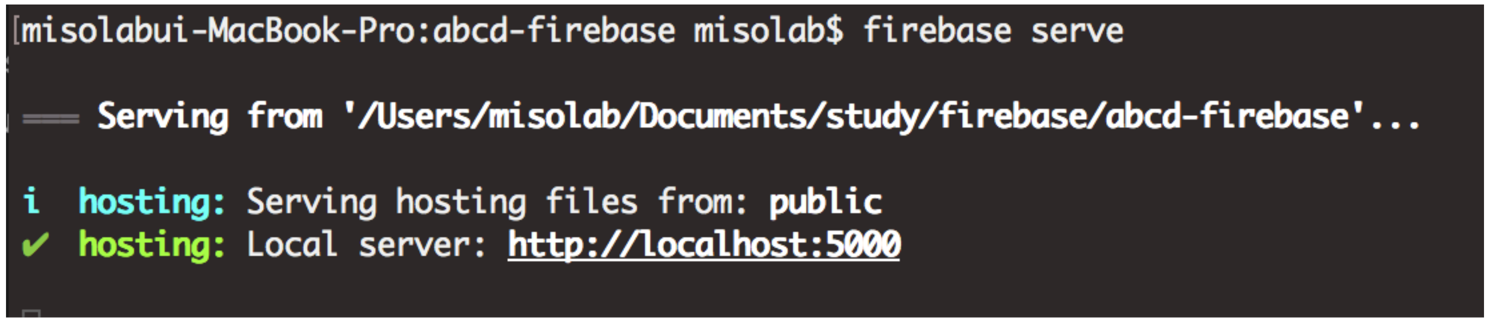
```
=== Database Setup

Firebase Realtime Database Rules allow you to define how your data should be
structured and when your data can be read from and written to.

? What file should be used for Database Rules? (database.rules.json)
```



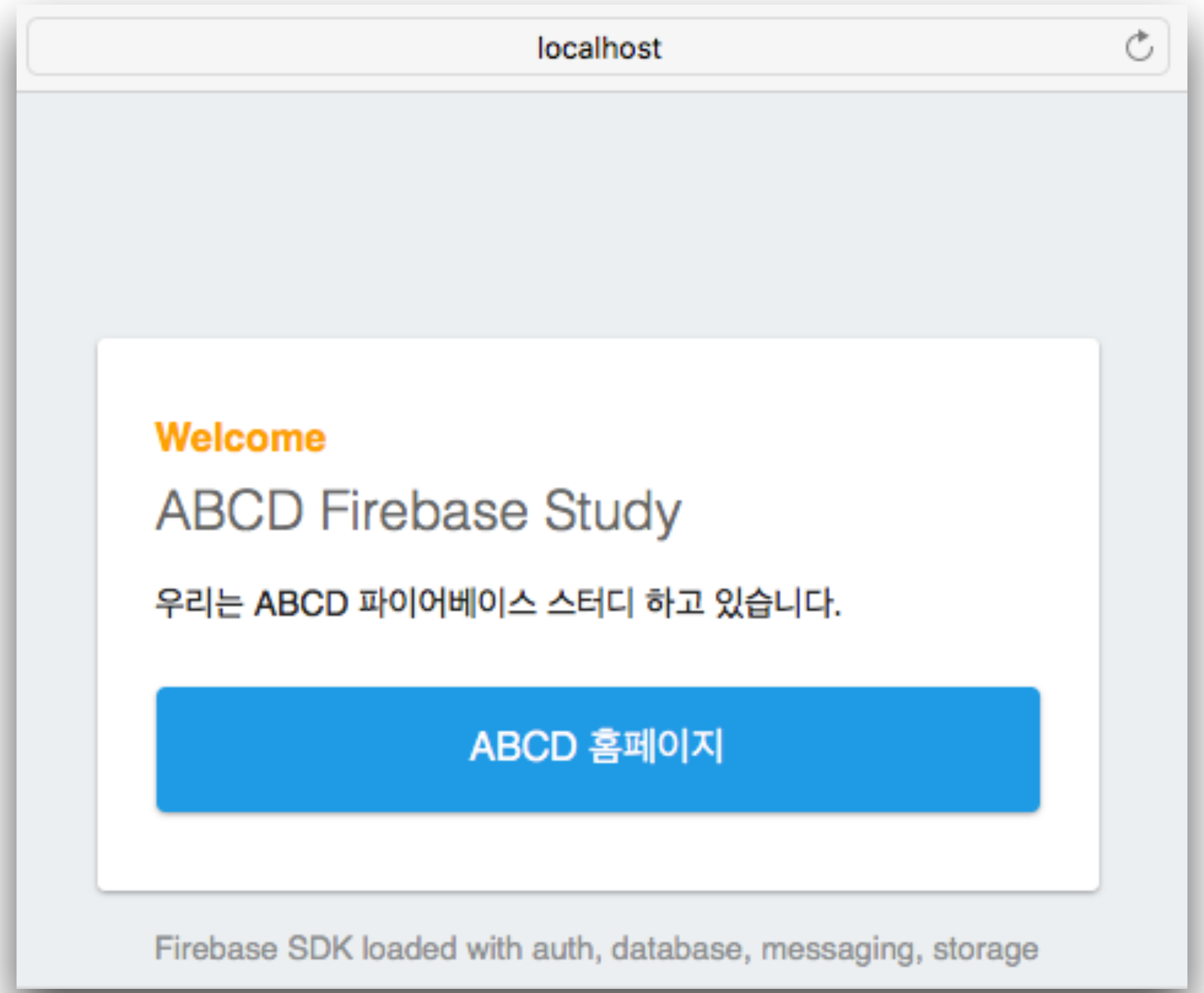
```
=== Hosting Setup

Your public directory is the folder (relative to your project directory) that
will contain Hosting assets to be uploaded with firebase deploy. If you
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? (public)
```

https://firebase.google.com/docs/web/setup

firebase serve



```
[misolabui-MacBook-Pro:abcd-firebase misolab$ firebase serve

=== Serving from '/Users/misolab/Documents/study/firebase/abcd-firebase'...

i  hosting: Serving hosting files from: public
✔  hosting: Local server: http://localhost:5000
```



localhost

Welcome

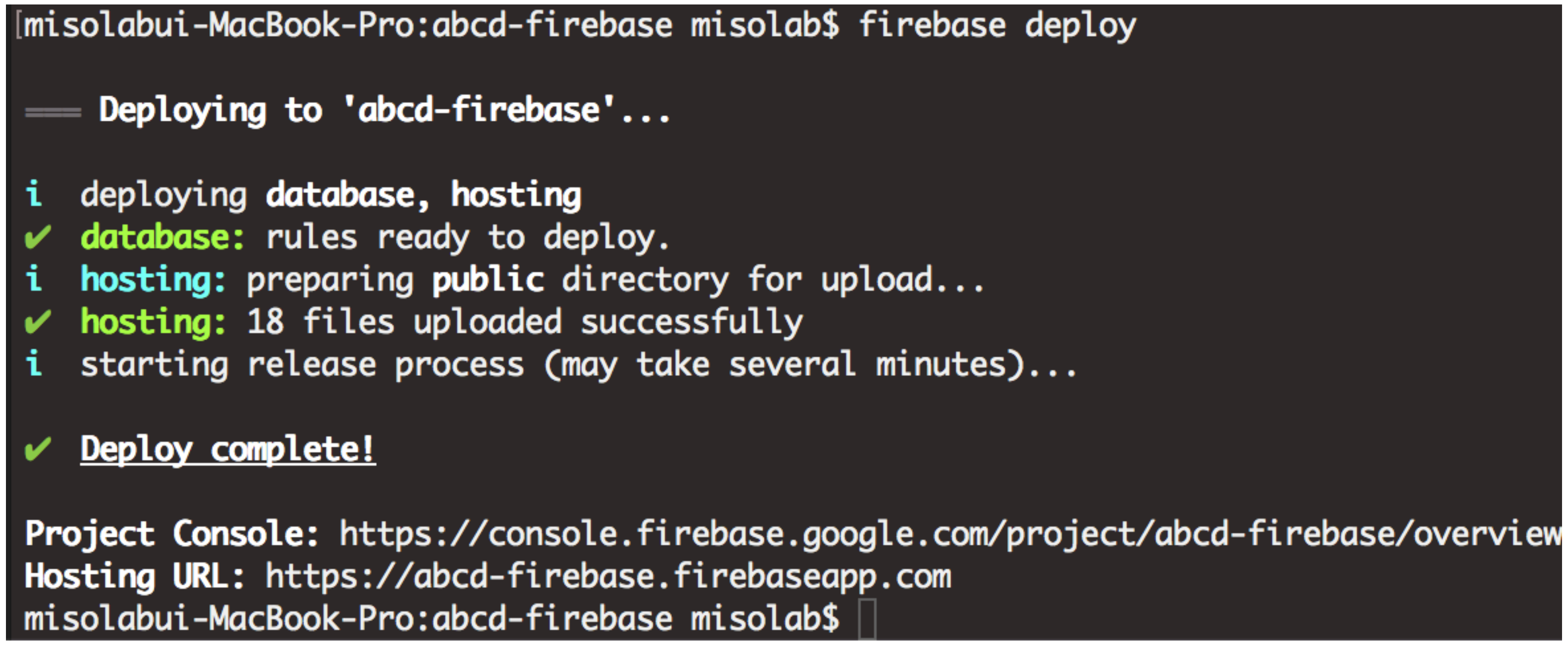
ABCD Firebase Study

우리는 ABCD 파이어베이스 스터디 하고 있습니다.

Firebase SDK loaded with auth, database, messaging, storage

https://firebase.google.com/docs/web/setup

firebase deploy

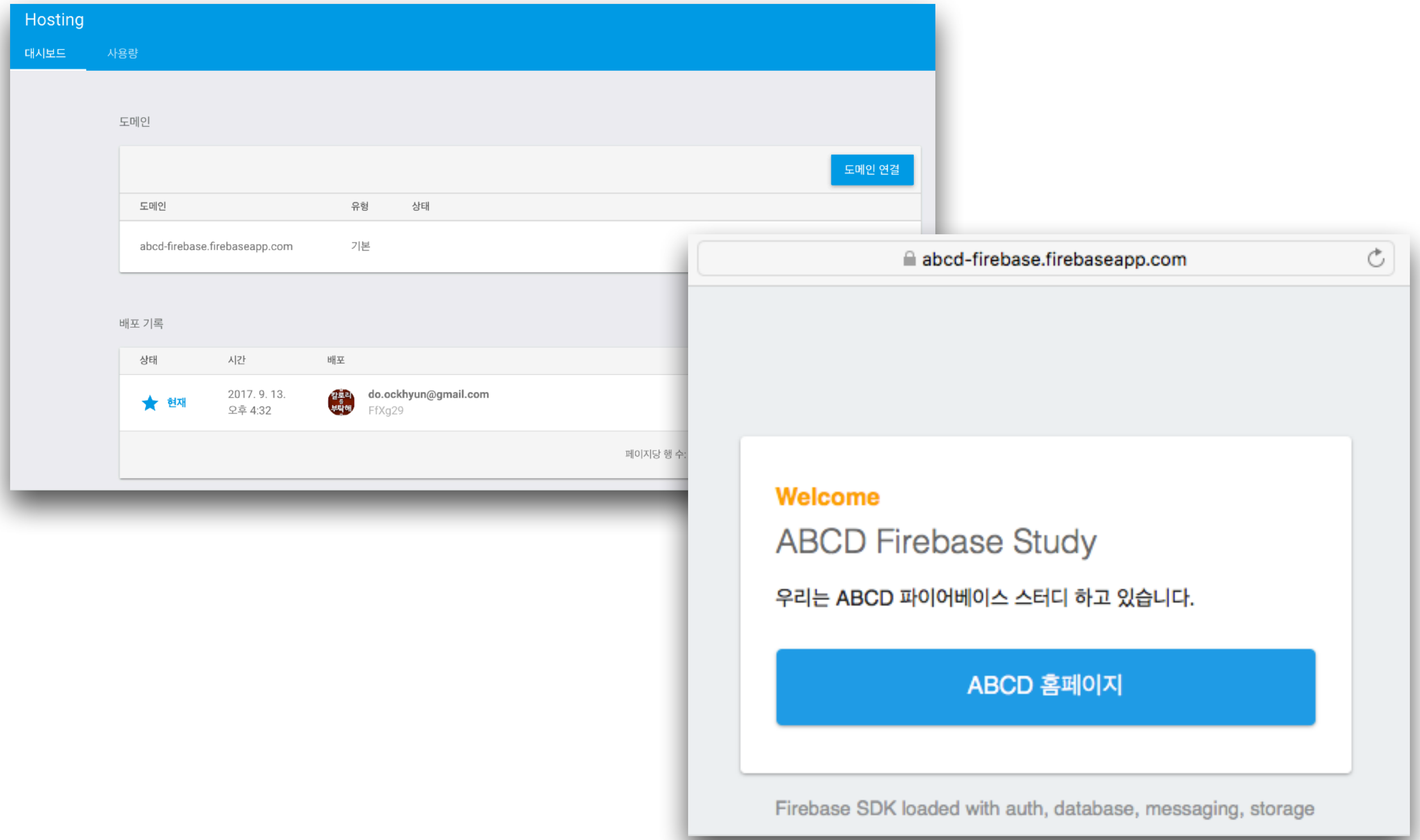


```
[misolabui-MacBook-Pro:abcd-firebase misolab$ firebase deploy

=== Deploying to 'abcd-firebase'...

i  deploying database, hosting
✔  database: rules ready to deploy.
i  hosting: preparing public directory for upload...
✔  hosting: 18 files uploaded successfully
i  starting release process (may take several minutes)...

✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/abcd-firebase/overview
Hosting URL: https://abcd-firebase.firebaseapp.com
misolabui-MacBook-Pro:abcd-firebase misolab$
```

```
[misolabui-MacBook-Pro:abcd-firebase misolab$ firebase deploy

=== Deploying to 'abcd-firebase'...

i  deploying database, hosting
✔  database: rules ready to deploy.
i  hosting: preparing public directory for upload...
✔  hosting: 18 files uploaded successfully
i  starting release process (may take several minutes)...

✔  Deploy complete!

Project Console: https://console.firebase.google.com/project/abcd-firebase/overview
Hosting URL: https://abcd-firebase.firebaseapp.com
misolabui-MacBook-Pro:abcd-firebase misolab$
```

https://firebase.google.com/docs/hosting/deploying

firebase deploy



Hosting

대시보드 사용량

도메인

도메인 연결

도메인	유형	상태
abcd-firebase.firebaseapp.com	기본	

배포 기록

상태	시간	배포
★ 현재	2017. 9. 13. 오후 4:32	do.ockhyun@gmail.com FfXg29

페이지당 행 수:

abcd-firebase.firebaseapp.com

Welcome

ABCD Firebase Study

우리는 ABCD 파이어베이스 스터디 하고 있습니다.

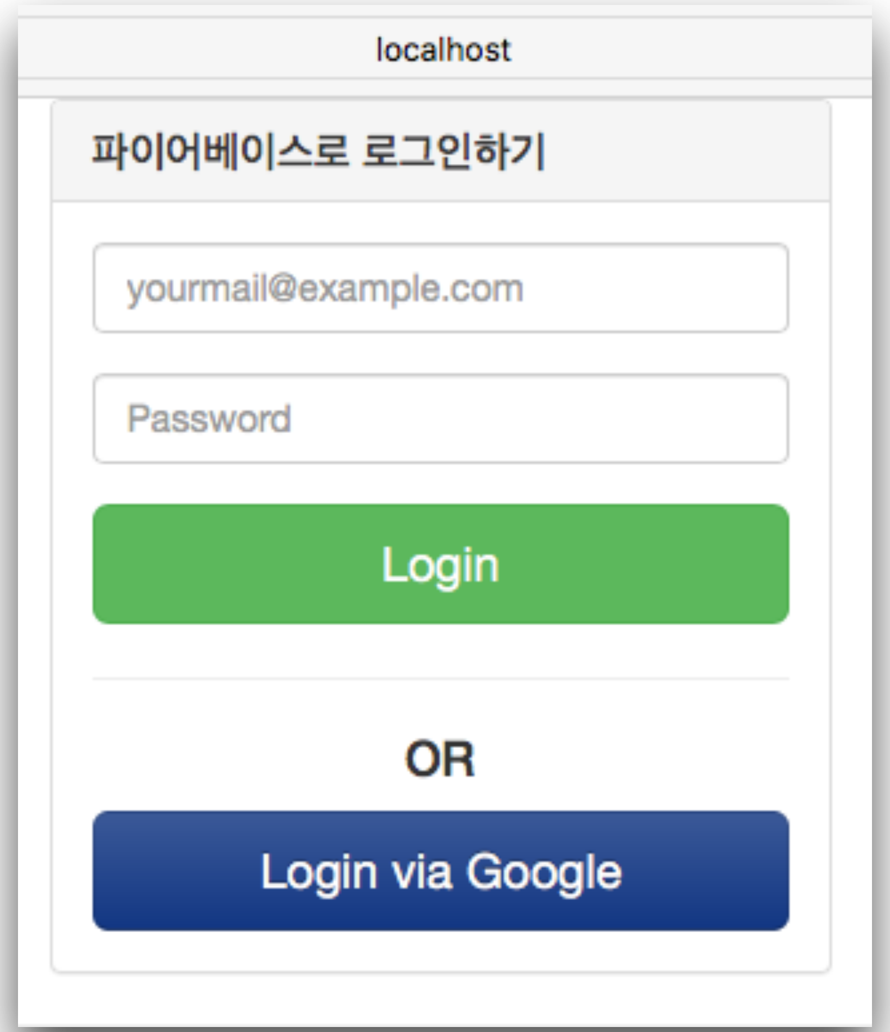
ABCD 홈페이지

Firebase SDK loaded with auth, database, messaging, storage

https://firebase.google.com/docs/hosting/deploying

Authentication

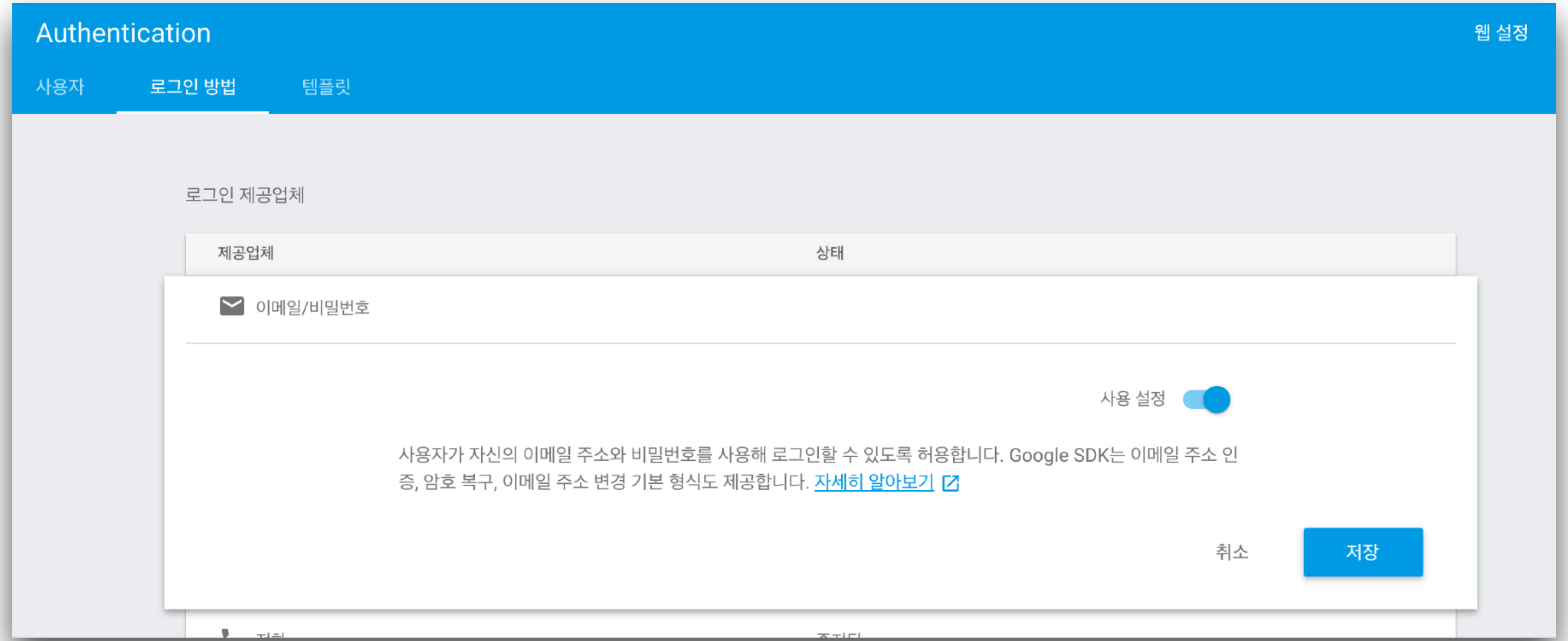
http://localhost:5000/auth/



https://firebase.google.com/docs/auth/web/start

https://firebase.google.com/docs/auth/web/manage-users

project_home > Authentication > 로그인방법



로그인 제공업체

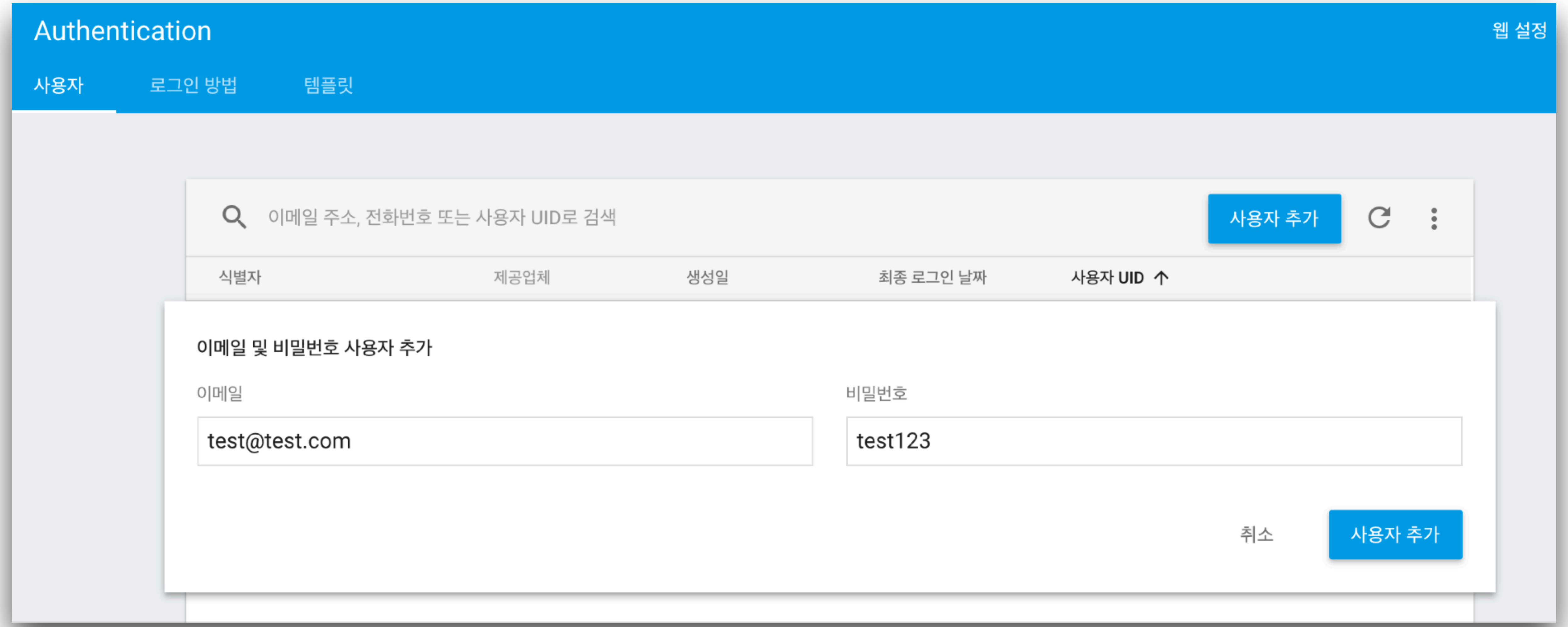
제공업체	상태
이메일/비밀번호	

사용 설정

사용자가 자신의 이메일 주소와 비밀번호를 사용해 로그인할 수 있도록 허용합니다. Google SDK는 이메일 주소 인증, 암호 복구, 이메일 주소 변경 기본 형식도 제공합니다. 자세히 알아보기

https://firebase.google.com/docs/auth/web/password-auth

project_home > Authentication > 사용자



이메일 및 비밀번호 사용자 추가

이메일	비밀번호
test@test.com	test123

https://firebase.google.com/docs/auth/web/password-auth

AUTH - authStateChanged

```
document.addEventListener('DOMContentLoaded', function () {

        firebase.auth().onAuthStateChanged(user => {
            console.log('onAuthStateChanged');
            console.log(user);

            if (user) {
                var displayName = user.displayName;
                var email = user.email;
                var photoURL = user.photoURL;
                var uid = user.uid;
                alert('Hi!! ' + email);

                btnLogout.classList.remove('hide');
            } else {
                // User is signed out.
                btnLogout.classList.add('hide');
            }
        });
    });
```

https://firebase.google.com/docs/auth/web/password-auth

AUTH - email, pwd

```
btnLogin.addEventListener('click', e => {
        console.log('login by email');

        const email = txtEmail.value;
        const password = txtPwd.value;

        firebase.auth().signInWithEmailAndPassword(email, password)
            .catch(e => {
                console.log(e);
                alert(e.message);
            })
            /*
            .then(user => {
                console.log('login Success!!');
                console.log(user);

                var displayName = user.displayName;
                var email = user.email;
                var photoURL = user.photoURL;
                var uid = user.uid;

                alert('Hi!! ' + email);
            });
            */
    });
```

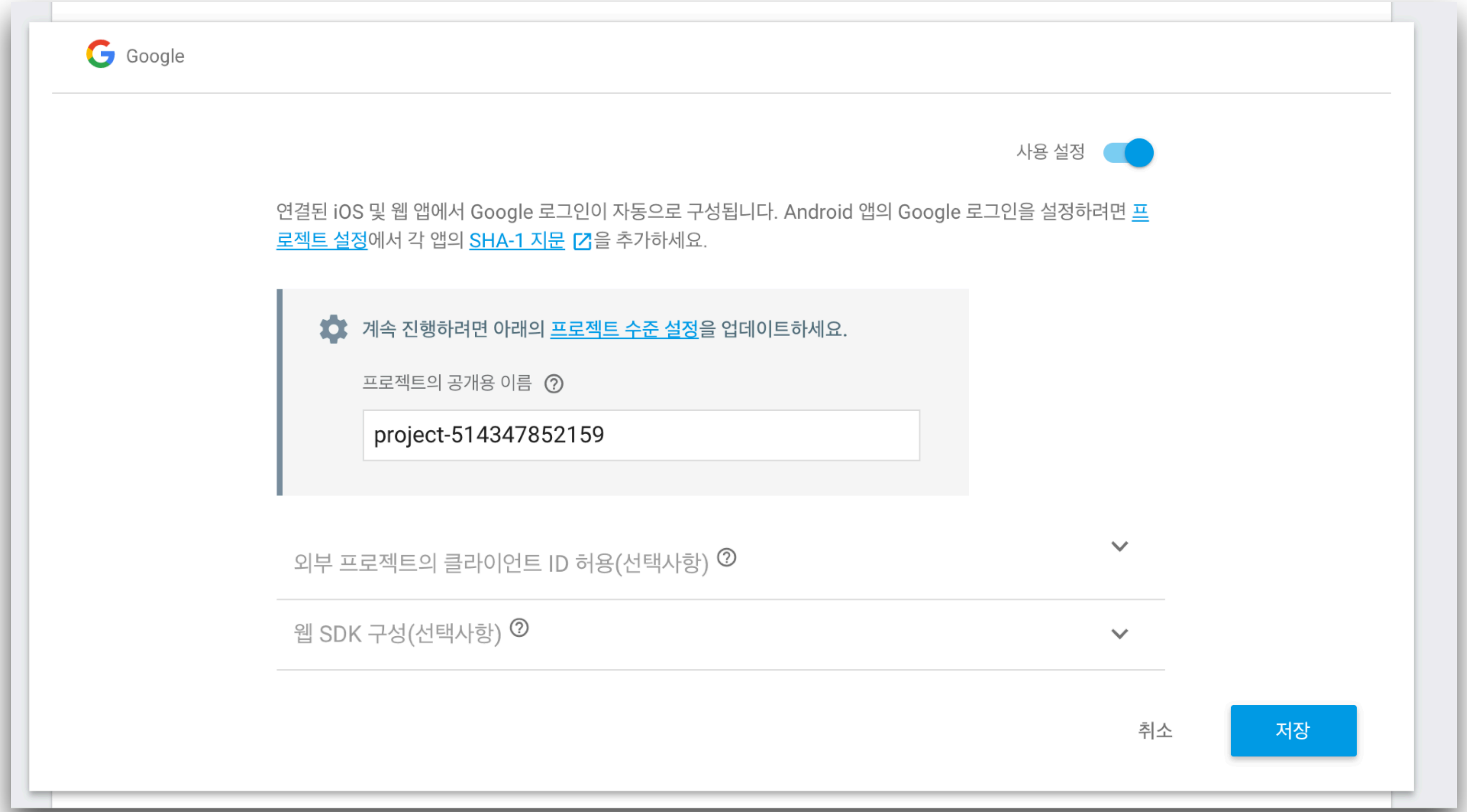
https://firebase.google.com/docs/auth/web/password-auth

AUTH - loggout

```
    btnLogout.addEventListener('click', e => {
        if (!confirm("Loggout??")) {
            return;
        }
        firebase.auth().signOut().then(() => {
            alert("bye~bye!!");
        });

    });
```

https://firebase.google.com/docs/auth/web/password-auth

project_home > Authentication > 로그인방법


Google

사용 설정

연결된 iOS 및 웹 앱에서 Google 로그인이 자동으로 구성됩니다. Android 앱의 Google 로그인을 설정하려면 프로젝트 설정에서 각 앱의 SHA-1 지문 을 추가하세요.

계속 진행하려면 아래의 프로젝트 수준 설정을 업데이트하세요.

프로젝트의 공개용 이름 ⓘ

project-514347852159

외부 프로젝트의 클라이언트 ID 허용(선택사항) ⓘ

웹 SDK 구성(선택사항) ⓘ

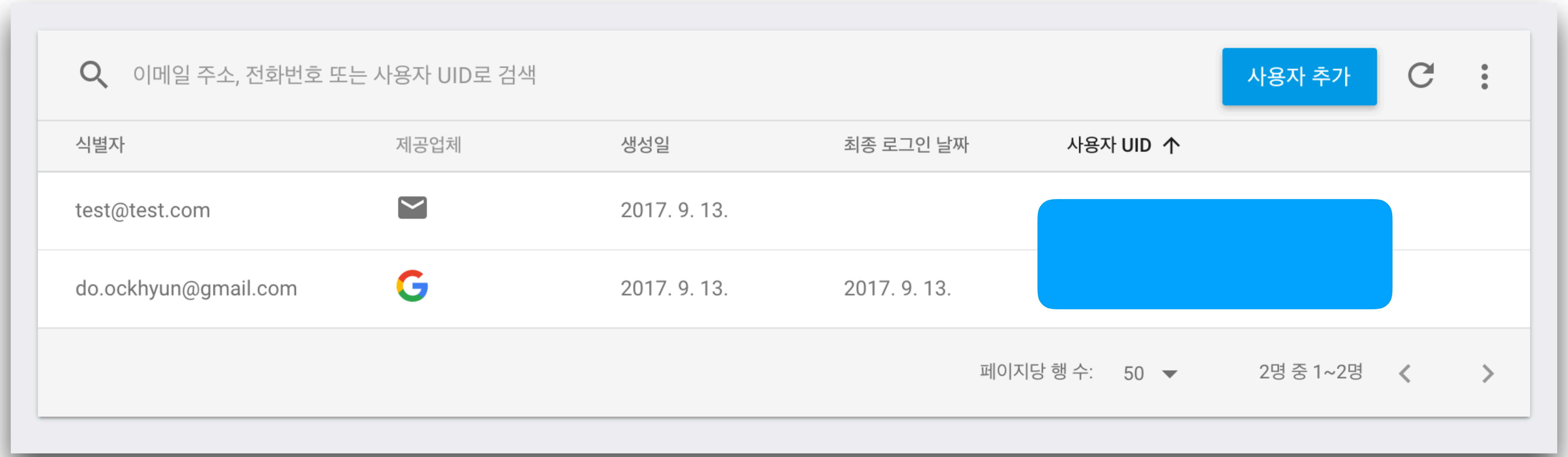
https://firebase.google.com/docs/auth/web/google-signin

AUTH - google

```
    btnLoginByGoogle.addEventListener('click', e => {
        console.log('login by Google');

        var provider = new firebase.auth.GoogleAuthProvider();
        firebase.auth().signInWithPopup(provider)
            .catch(e => {
                console.log(e);
                alert(e.message);
            })
            .then(result => {
                console.log('login Success!!');
                var user = result.user;

                console.log(user);
                var displayName = user.displayName;
                var email = user.email;
                var photoURL = user.photoURL;
                var uid = user.uid;

                alert('Hi!! ' + email);
            });

    });
```

https://firebase.google.com/docs/auth/web/google-signin

project_home > Authentication > 사용자



이메일 주소, 전화번호 또는 사용자 UID로 검색

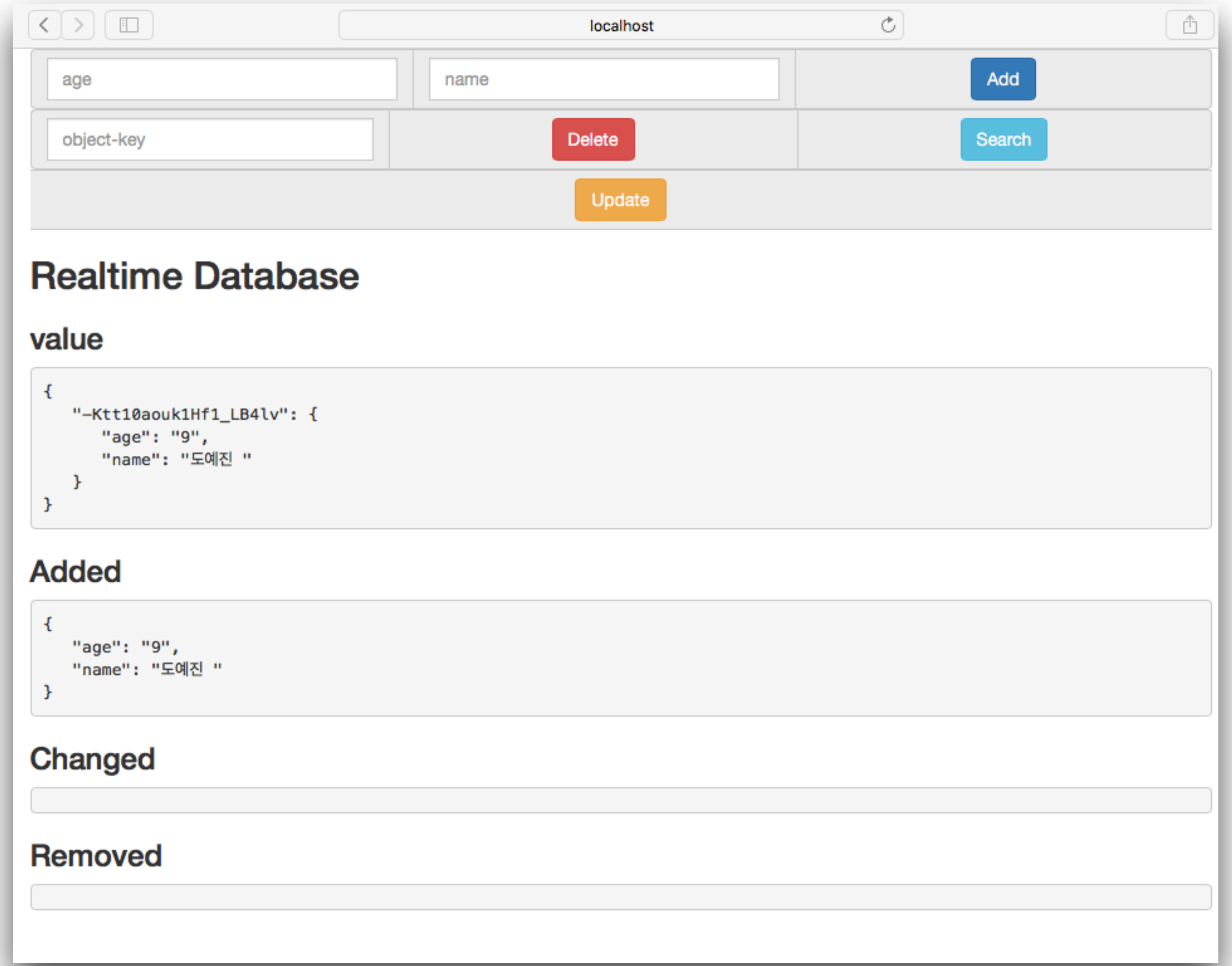
식별자	제공업체	생성일	최종 로그인 날짜	사용자 UID ↑
test@test.com	✉	2017. 9. 13.		[redacted]
do.ockhyun@gmail.com	G	2017. 9. 13.	2017. 9. 13.	[redacted]

페이지당 행 수: 50 2명 중 1~2명

https://firebase.google.com/docs/auth/web/password-auth

Realtime Database

http://localhost:5000/db/



Realtime Database

value

```
{
   "-Ktt10aouk1Hf1_LB4lv": {
      "age": "9",
      "name": "도예진 "
   }
}
```

Added

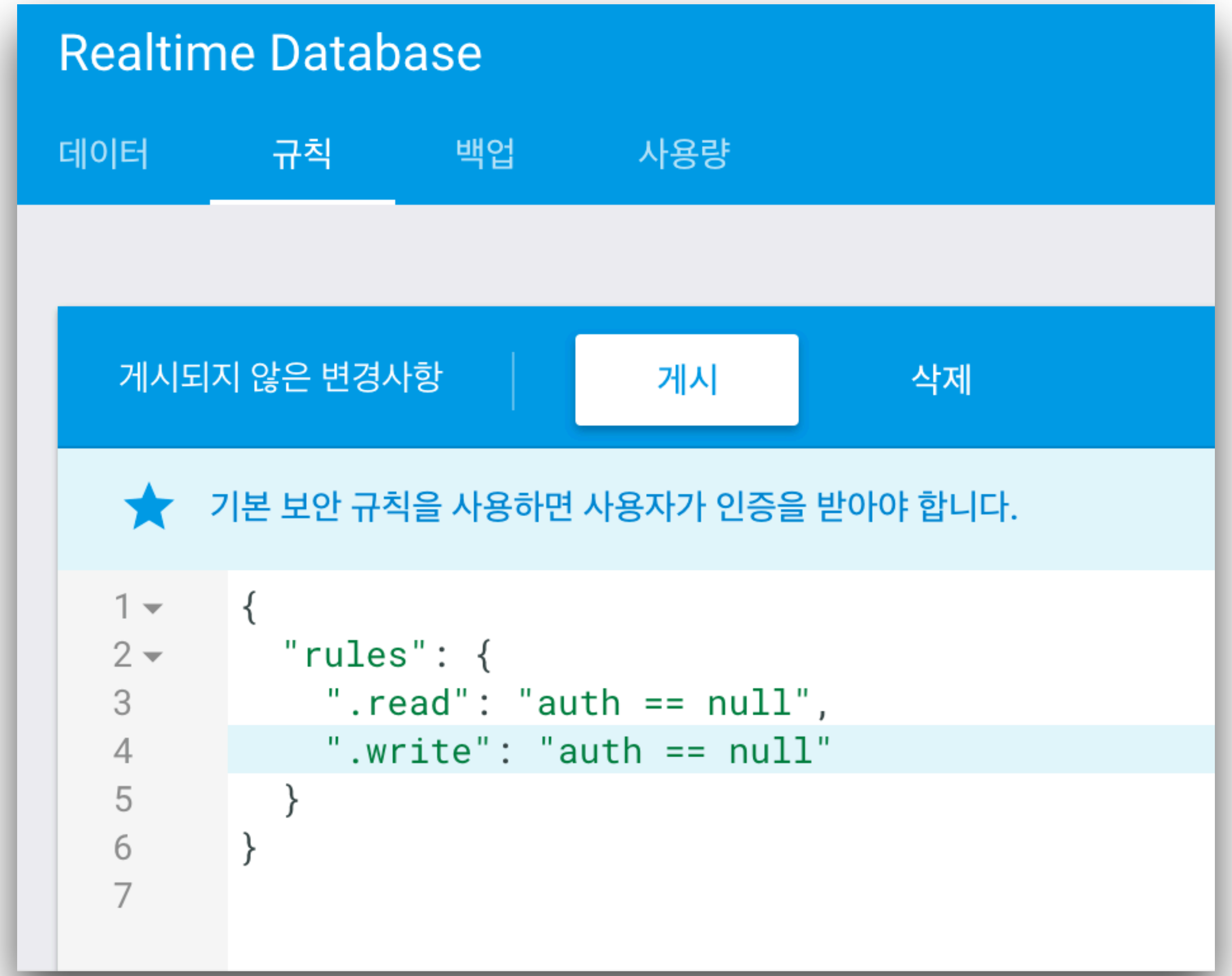
```
{
   "age": "9",
   "name": "도예진 "
}
```

Changed

Removed

https://firebase.google.com/docs/database/web/structure-data

보안 및 규칙



기본 보안 규칙을 사용하면 사용자가 인증을 받아야 합니다.

```
{
  "rules": {
    ".read": "auth == null",
    ".write": "auth == null"
  }
}
```

https://firebase.google.com/docs/database/security/

DB - value, child_added, child_changed, child_remove

```
    document.addEventListener('DOMContentLoaded', () => {
        console.log('DOMContentLoaded');

        const dbObjectRef = firebase.database().ref('object');

        // TODO : rule change anonymous!!
        dbObjectRef.on('value', snap => {
            console.log(snap);
            objValue.innerText = JSON.stringify(snap.val(), null, 3);
        });

        dbObjectRef.on('child_added', snap => {
            console.log(snap);
            objAdded.innerText = JSON.stringify(snap.val(), null, 3);
        });

        dbObjectRef.on('child_changed', snap => {
            console.log(snap);
            objChanged.innerText = JSON.stringify(snap.val(), null, 3);
        });

        dbObjectRef.on('child_removed', snap => {
            console.log(snap);
            objRemoved.innerText = JSON.stringify(snap.val(), null, 3);
        });
    });
```

https://firebase.google.com/docs/database/web/read-and-write

https://firebase.google.com/docs/database/web/lists-of-data

DB - push

```
btnAdd.addEventListener('click', e => {
        const newUser = {
            age: txtAge.value,
            name: txtName.value
        };
        const newUserRef = firebase.database().ref('object').push();
        console.log('newUser key -' + newUserRef.key);
        newUserRef.set(newUser);
    });
```

https://firebase.google.com/docs/database/web/read-and-write

https://firebase.google.com/docs/database/web/lists-of-data

DB - once

```
btnSearch.addEventListener('click', e => {
        const searchKey = txtKey.value;
        const searchRef = firebase.database().ref('object/' + searchKey);
        searchRef.once('value')
            .then(snap => {
                alert(JSON.stringify(snap.val(), null, 3));
            }).catch(e => {
                alert(e.message);
            });
    });
```

https://firebase.google.com/docs/database/web/read-and-write

https://firebase.google.com/docs/database/web/lists-of-data

DB - update

```
btnUpdate.addEventListener('click', e => {
        const updateKey = txtKey.value;
        const newData = {
            age: txtAge.value,
            name: txtName.value
        };
        const updateRef = firebase.database().ref('object/' + updateKey);
        updateRef.update(newData);
    });
```

https://firebase.google.com/docs/database/web/read-and-write

https://firebase.google.com/docs/database/web/lists-of-data

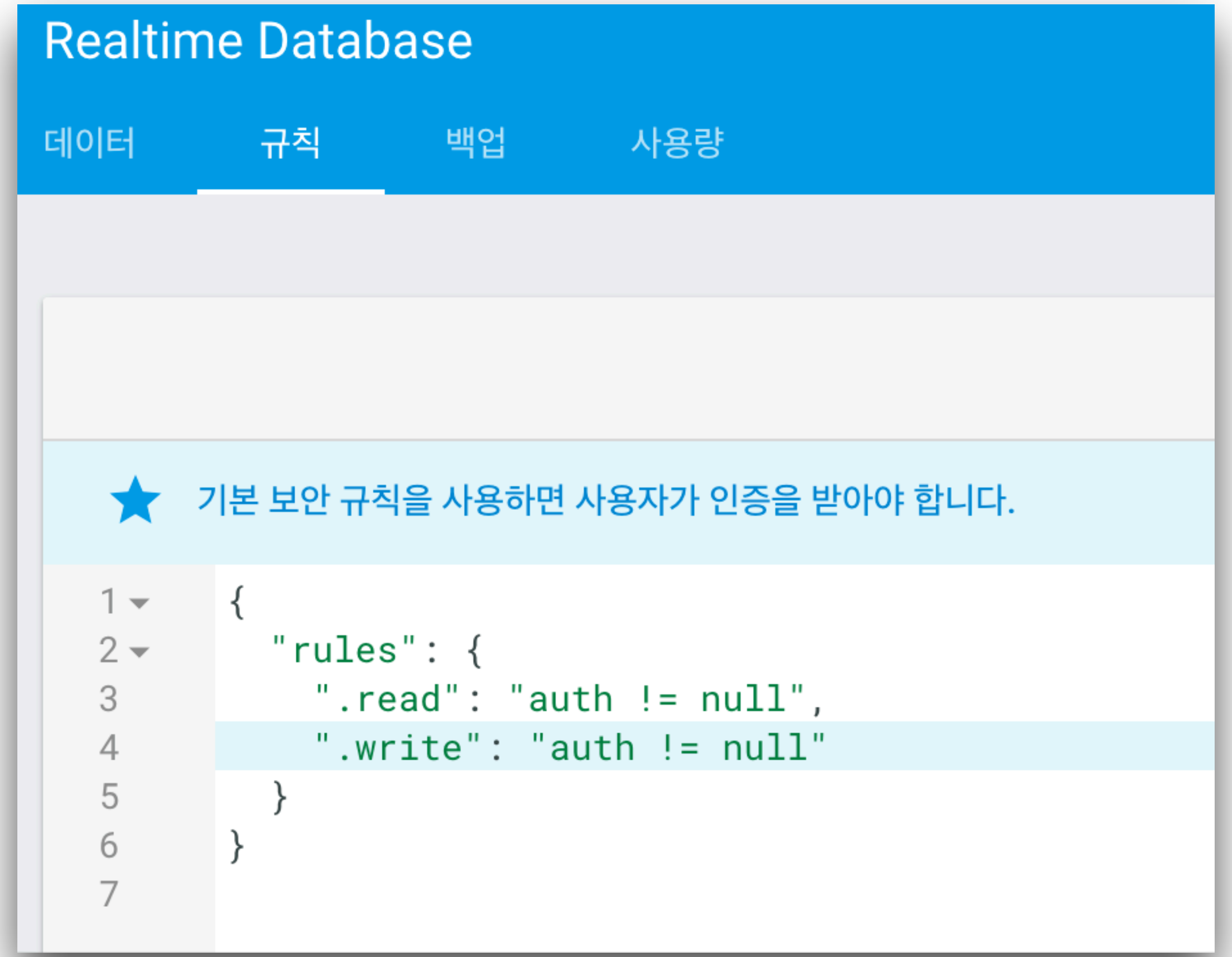
DB - delete

```
btnDel.addEventListener('click', e => {
        if (!confirm('삭제하시겠습니까?')) {
            return;
        }

        const deleteKey = txtKey.value;
        const deleteRef = firebase.database().ref('object/' + deleteKey);
        deleteRef.remove();
    });
```

https://firebase.google.com/docs/database/web/read-and-write

https://firebase.google.com/docs/database/web/lists-of-data

보안 및 규칙



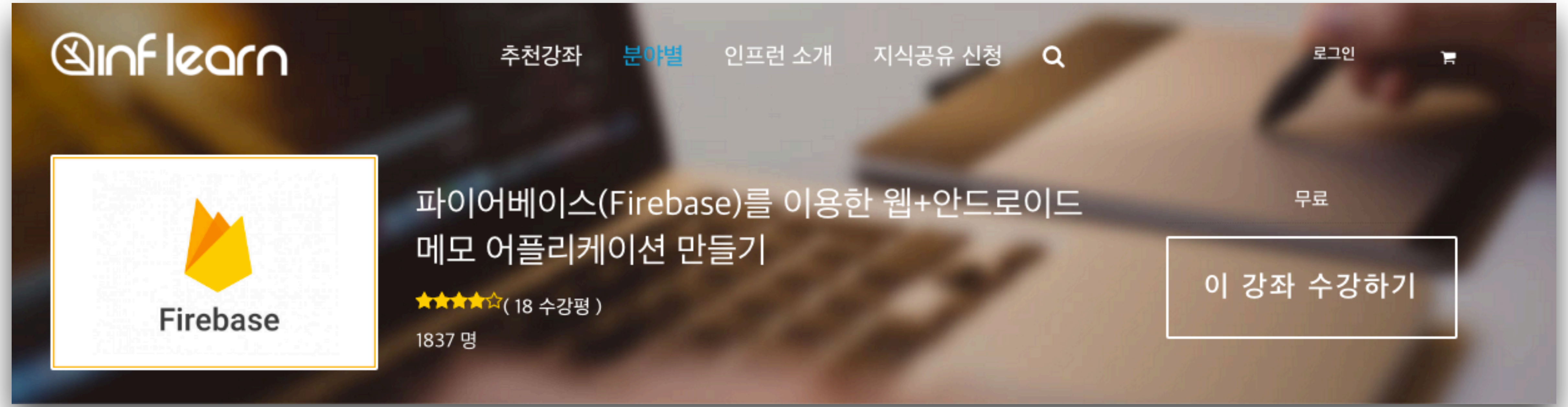
기본 보안 규칙을 사용하면 사용자가 인증을 받아야 합니다.

```
{
  "rules": {
    ".read": "auth != null",
    ".write": "auth != null"
  }
}
```

https://firebase.google.com/docs/database/security/

memo-webapp

inflean.com - 파이어베이스를 이용한 웹+안드로이드 메모 어플리케이션 만들기 (신휴창 님)

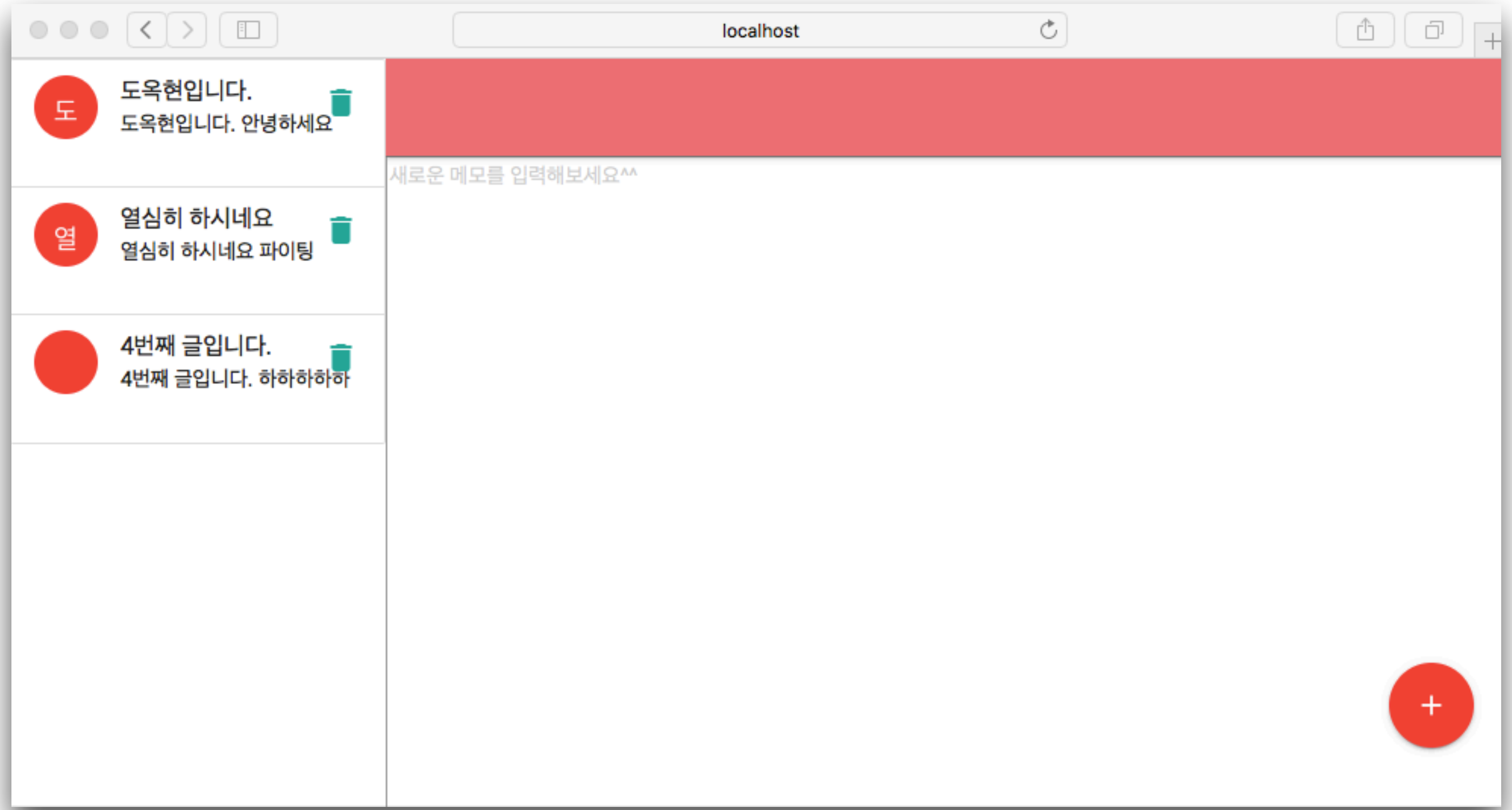


파이어베이스(Firebase)를 이용한 웹+안드로이드 메모 어플리케이션 만들기

★★★★☆(18 수강평)

1837 명

http://localhost:5000/memo/



localhost

도
도옥현입니다.
도옥현입니다. 안녕하세요

열
열심히 하시네요
열심히 하시네요 파이팅

4번째 글입니다.
4번째 글입니다. 하하하하하

새로운 메모를 입력해보세요^^

+

1. auth - google

```javascript
document.addEventListener('DOMContentLoaded', () => {
    console.log('DOMContentLoaded');

    $('.textarea').blur(() => {
        saveMemoData();
    });

    // 1. auth - google
    firebase.auth().onAuthStateChanged(user => {
        console.log(user);

        if (user) {
            userInfo = user;
            alert('hi!! ' + user.displayName);

            initMemoData();
        } else {
            var provider = new firebase.auth.GoogleAuthProvider();
            firebase.auth().signInWithPopup(provider)
                .catch(e => {
                    alert(e.message);
                });
        }
    });
});
```

2. db - init

```
    // 2. db - 0) init
    function initMemoData() {
        memoRef = firebase.database().ref('memos/' + userInfo.uid);
        console.log(memoRef);
    }
```

2. db - create

```
    // 2. db - 1) create
    function saveMemoData() {
        var txt = $(".textarea").val();
        if (txt == "") {
            return;
        }

        memoRef.push({
            txt: txt,
            createTime: new Date().getTime()
        });

    }
```

2. db - read

```
// 2. db - 0) init
    function initMemoData() {
        memoRef = firebase.database().ref('memos/' + userInfo.uid);

        // 2. db - 2) read
        memoRef.on("child_added", (data) => {
            var key = data.key;
            var memoData = data.val();
            var txt = memoData.txt;
            var title = txt.substr(0, txt.indexOf('\n'));
            var firstTxt = txt.substr(0, 1);
            var html =
                "<li id='" + key + "' class=\"collection-item avatar\" >" +
                "<i class=\"material-icons circle red\">" + firstTxt + "</i>" +
                "<span class=\"title\">" + title + "</span>" +
                "<p class='txt'>" + txt + "<br>" +
                "</p>" +
                "</li>";

            $(".collection").append(html);
        });
    }
```

2. db - read (once)

```javascript
// 2. db - 2) read
function loadMemoData(key) {
    selectedKey = key;
    var currentRef = firebase.database().ref('memos/' + userInfo.uid + "/" + key);
    currentRef.once("value").then((snapshot) => {
        $(".textarea").val(snapshot.val().txt);
    });
}
```

2. db - read (once)

```javascript
// 2. db - 2) read
function loadMemoData(key) {
    selectedKey = key;
    var currentRef = firebase.database().ref('memos/' + userInfo.uid + "/" + key);
    currentRef.once("value").then((snapshot) => {
        $(".textarea").val(snapshot.val().txt);
    });
}
```

2. db - update

```
// 2. db - 0) init
function initMemoData() {
    memoRef = firebase.database().ref('memos/' + userInfo.uid);

...

    // 2. db - 3) update
    memoRef.on("child_changed", (data) => {
        var key = data.key;
        var txt = data.val().txt;
        var title = txt.substr(0, txt.indexOf('\n'));

        $("#" + key + " > .title").text(title);
        $("#" + key + " > .txt").text(txt);
    });
}
```

2. db - update

```
// 2. db - 1) create
function saveMemoData() {
    var txt = $(".textarea").val();
    if (txt == "") {
        return;
    }

    // 2. db - 3) update
    if (selectedKey) {
        var currentRef = firebase.database()
                                        .ref(‘memos/' + userInfo.uid + "/" + selectedKey);
        currentRef.update({
            txt: txt,
            updateTime: new Date().getTime()
        });
    } else {
        const newMemoRef = memoRef.push({
            txt: txt,
            createTime: new Date().getTime()
        });
        selectedKey = newMemoRef.key;
    }
}
```

2. db - create (new)

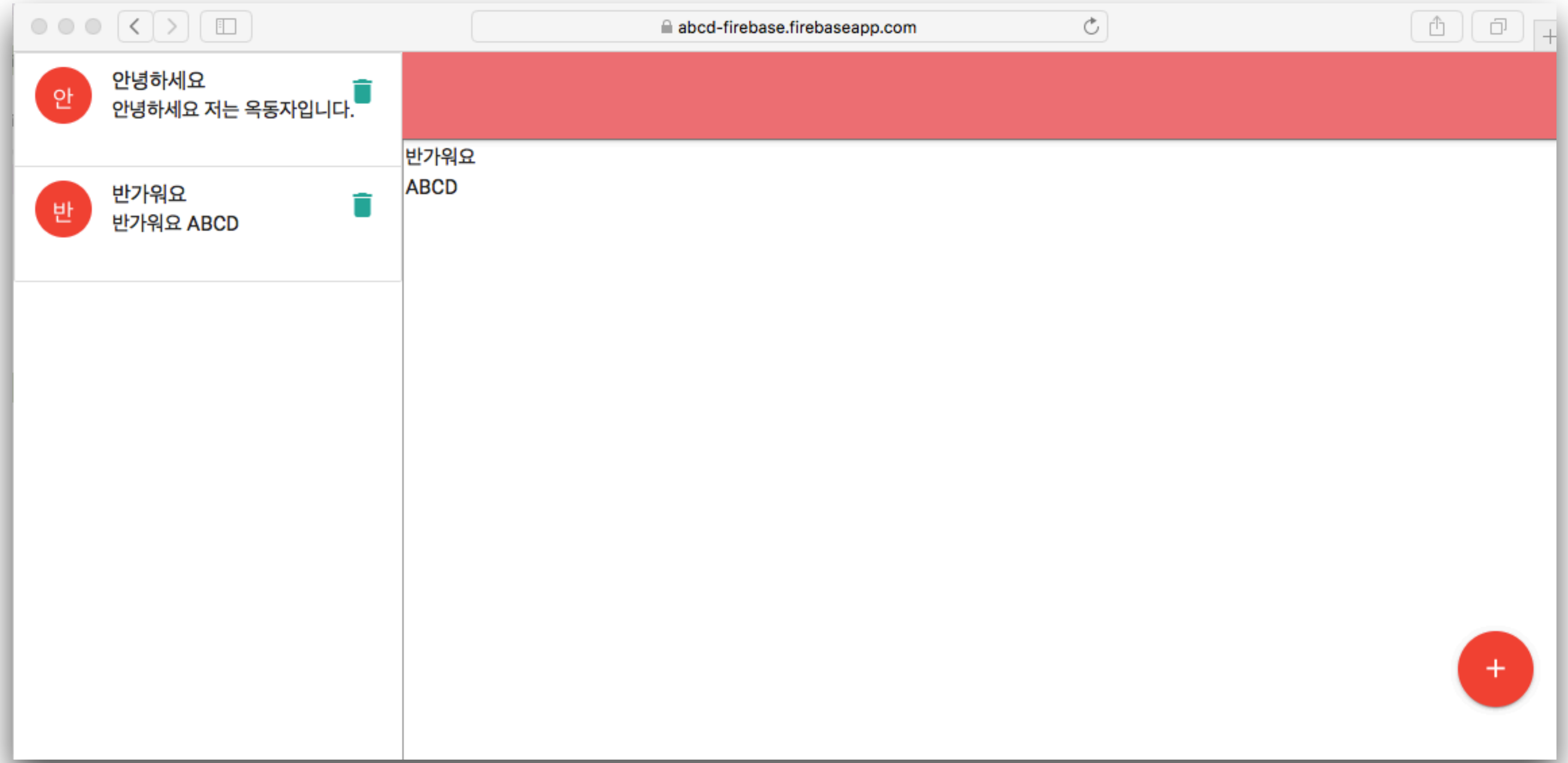
```
// 2. db - 1) create (new)
function newMemoData() {
    $(".textarea").val("");
    selectedKey = null;
}
```

2. db - delete

```
var html =
          "<li id='" + key + "' class=\"collection-item avatar\" onclick=\"loadMemoData(this.id);\">" +
          "<i class=\"material-icons circle red\">" + firstTxt + "</i>" +
          "<span class=\"title\">" + title + "</span>" +
          "<p class='txt'>" + txt + "<br>" +
          "</p>" +

        // 2. db - 4) delete
"<a href=\"#!\" onclick=\"deleteMemoData('" + key + "');\" class=\”secondary-content\">
 <i class=\”material-icons\">delete</i></a>"

       "</li>";
```

```
// 2. db - 4) delete
function deleteMemoData(key) {
    if (!confirm("삭제하시겠습니까?")) {
        return;
    }

    var currentRef = firebase.database().ref('memos/' + userInfo.uid + "/" + key);
    currentRef.remove();
    $("#" + key).remove();

    newMemoData();
}
```

3. deploy


abcd-firebase.firebaseapp.com
안
안녕하세요
안녕하세요 저는 옥동자입니다.
반
반가워요
반가워요 ABCD
반가워요
ABCD
+