Tutorial #3:
Working with Data

CS374: Introduction to HCI

2017. 3. 27

Hyeungshik Jung

Summary of Previous Tutorials

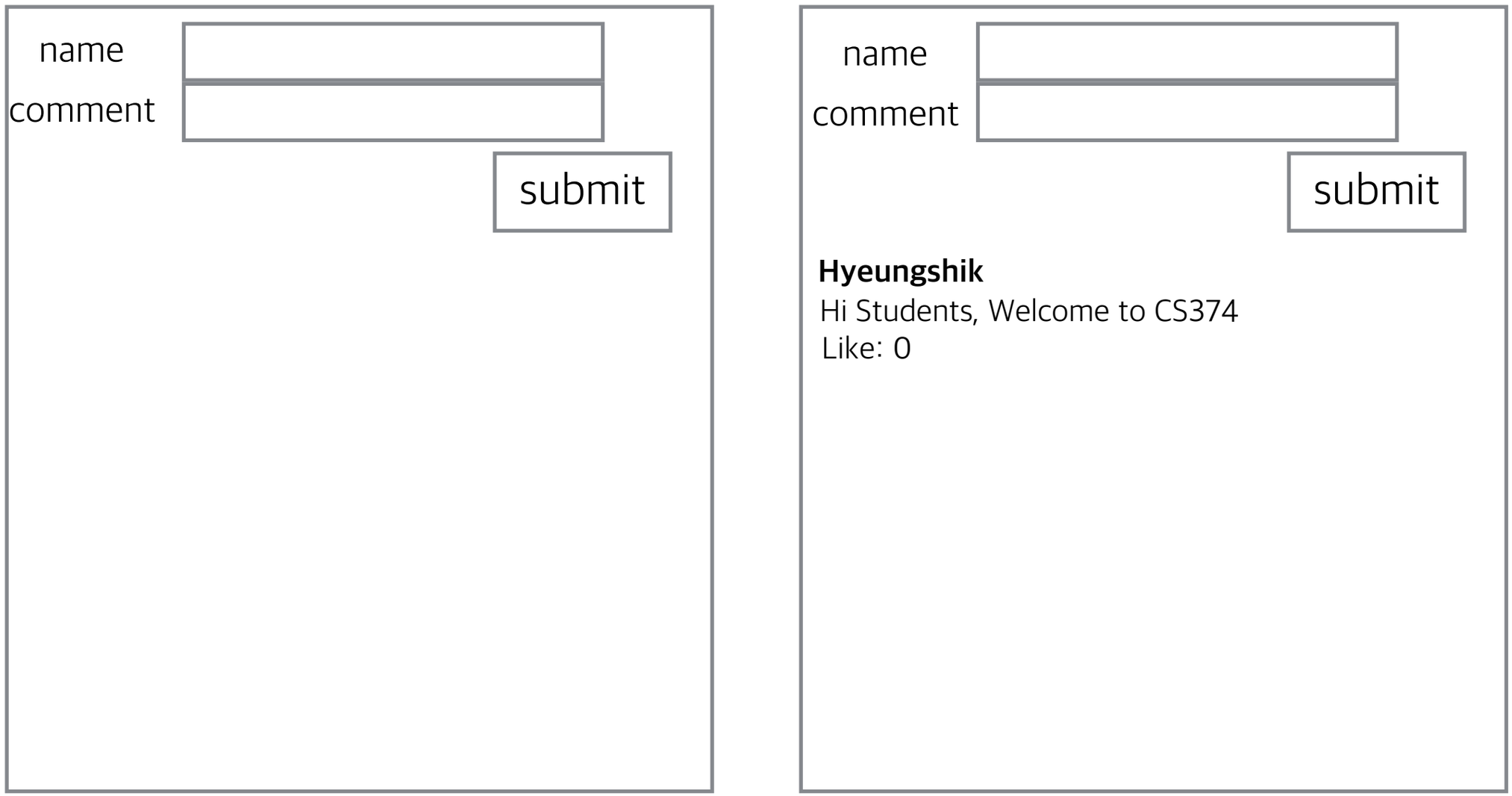
- HTML for presenting structure of a webpage
- CSS for presenting contents
- Javascript for manipulating DOM

Working with Data?

- How to store data?
- How to retrieve data?
- How to update interface as data is updated?

Learning Objective

- Students will write real-time comment board.



- We will use Firebase Database

Firebase

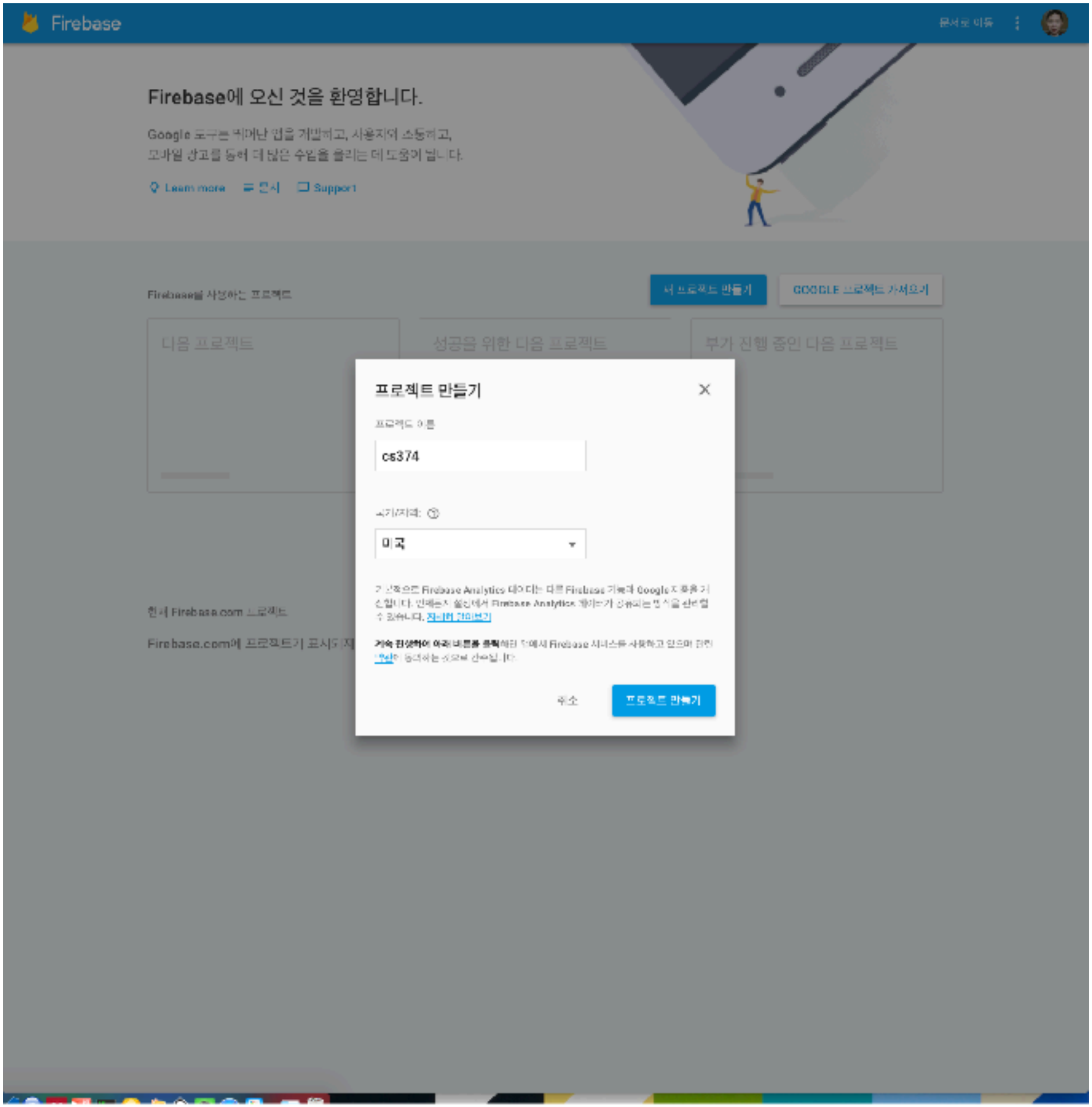
- Firebase is a set of services for making (web / mobile) application.

- It includes Authentication, **Database**, Storage, Hosting, Functions …

- Firebase Database stores & synchronizes data in real-time.

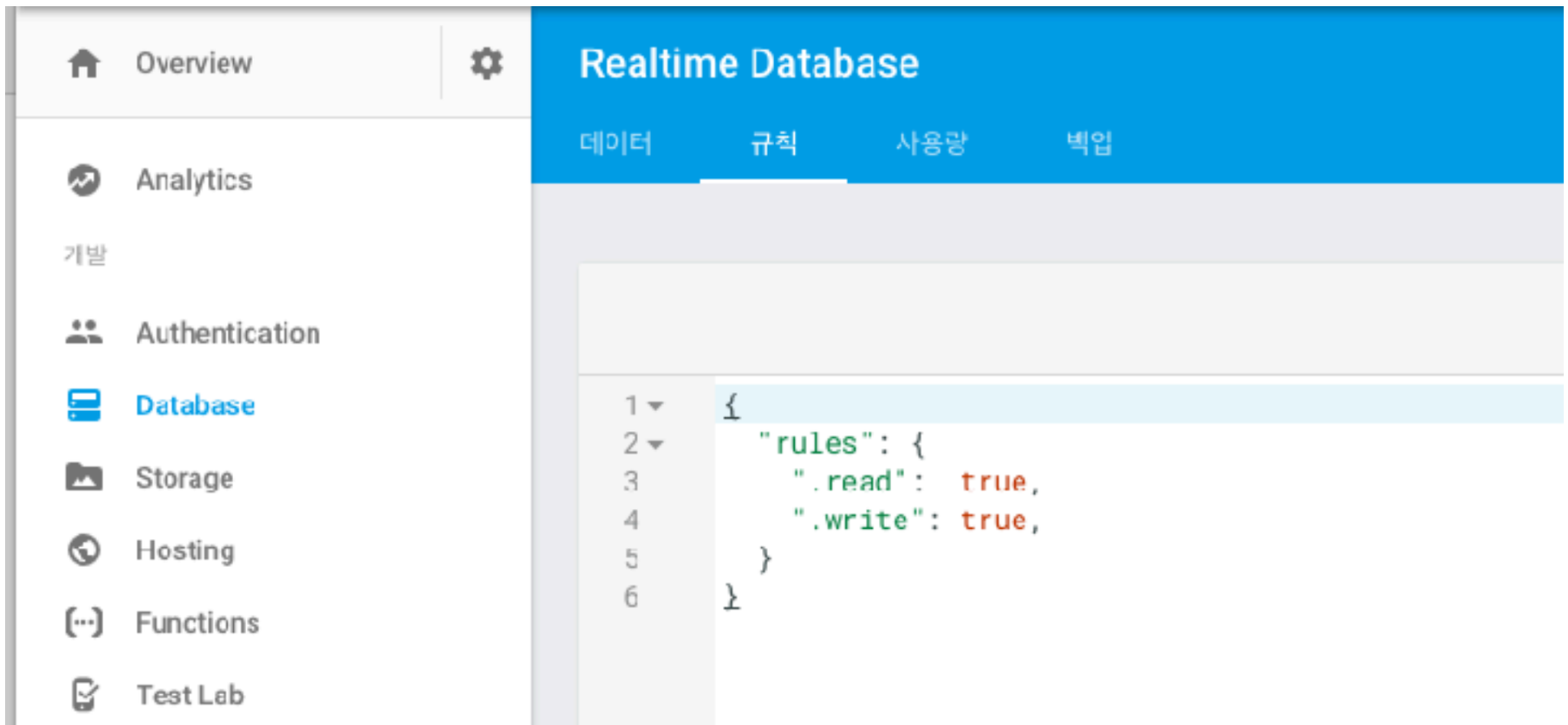
Starting Firebase Project

- Login with Google Account
- Go to https://firebase.google.com/
- Go to Firebase Console
- Create New Project


Firebase
Firebase에 오신 것을 환영합니다.
Learn more
문서
Support
Firebase을 사용하는 프로젝트
새 프로젝트 만들기
GOOGLE 프로젝트 가져오기
다음 프로젝트
성공을 위한 다음 프로젝트
부가 진행 중인 다음 프로젝트
현재 Firebase.com 프로젝트
프로젝트 만들기
cs374
미국
취소
프로젝트 만들기

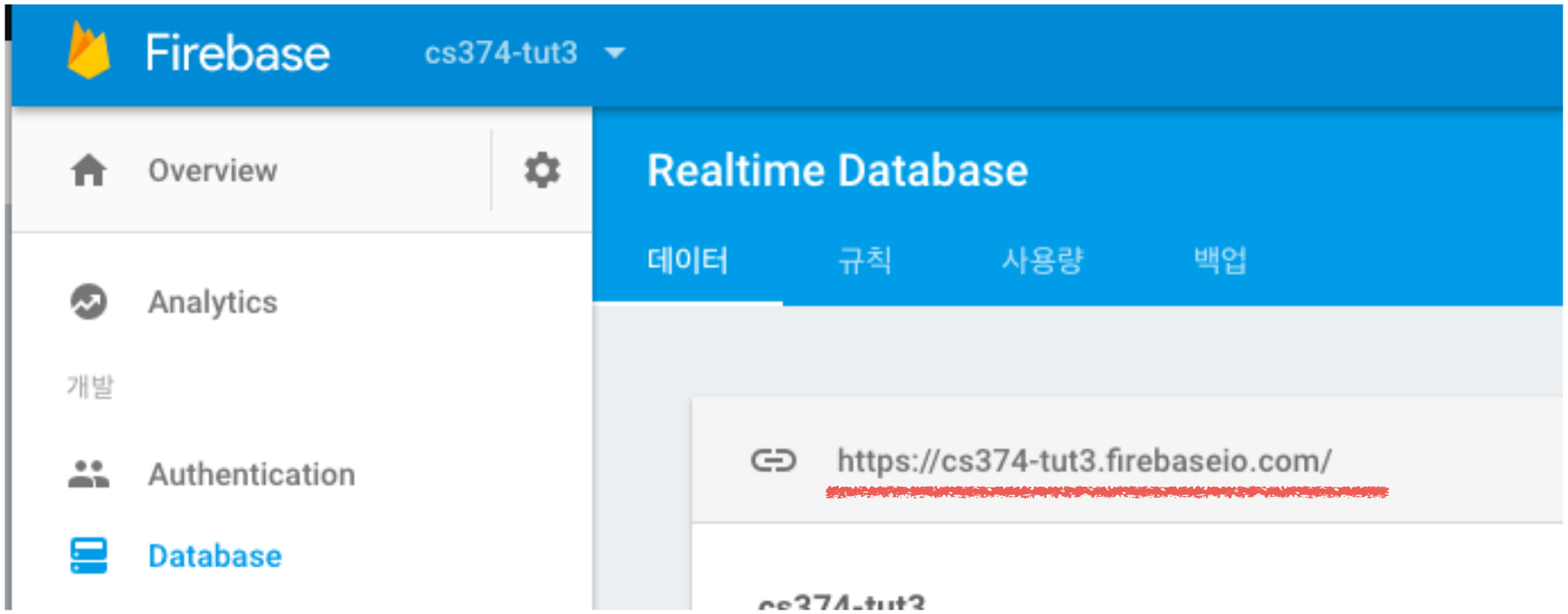
Setting up Firebase Database

- Go to Database
- Change the rule as below

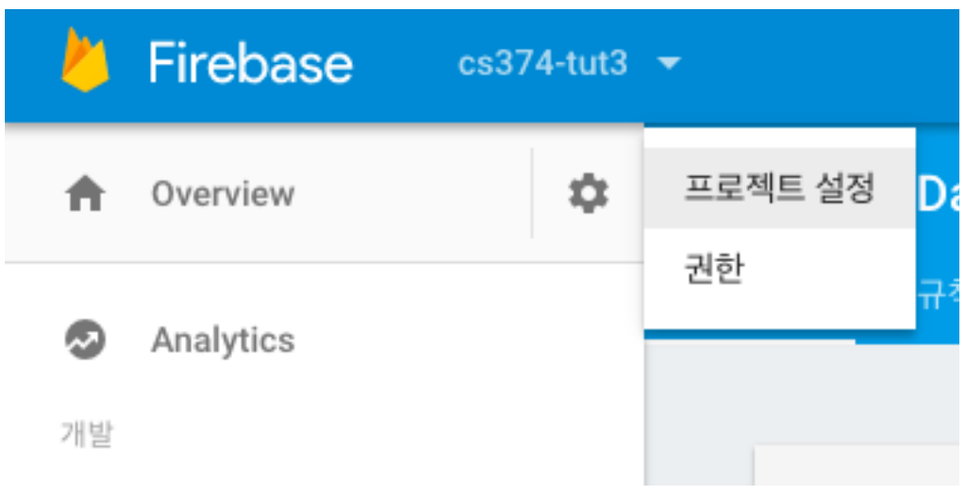


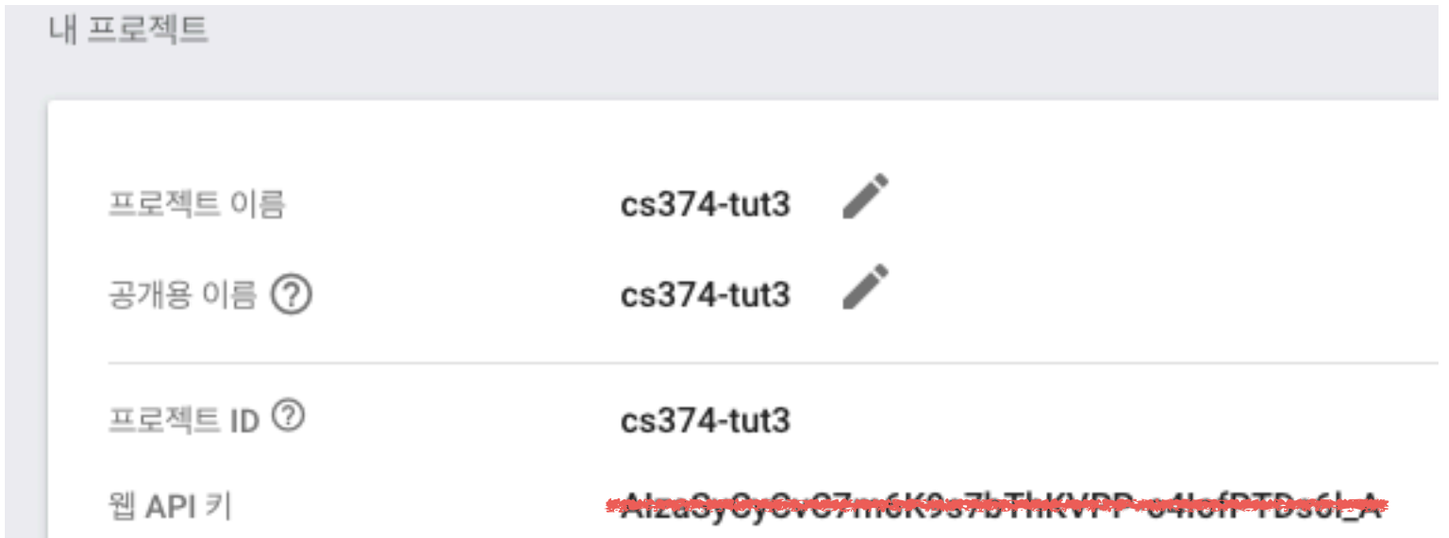
One More Step

- Memo your Firebase project's **database url**



- Memo your Firebase project's **Web API Key**

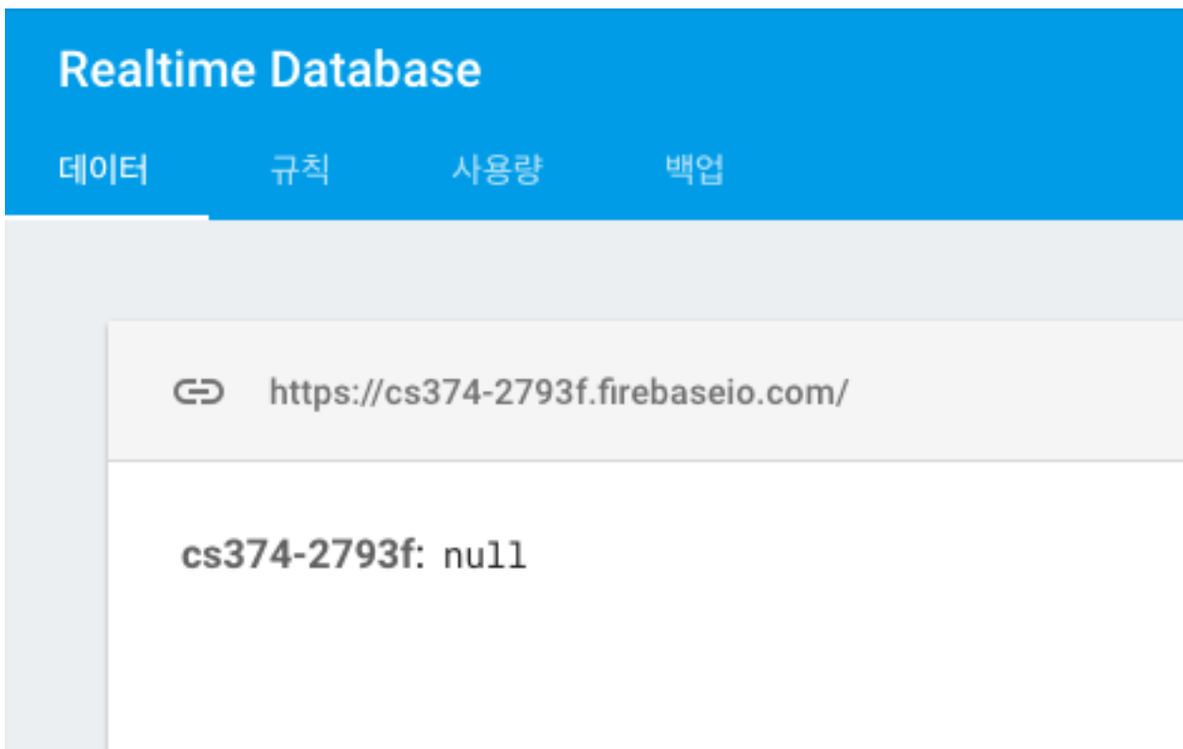




Database Structure

- Firebase Database stores data as a Javascript Object
- We will store our comments under “posts”
- Each comment is stored with key in “posts”

```
{
  “posts” {
     “key1”: {
          “username”: XXX,
          “comment”:YYY,
          “like”: 0
     }
     “key2”: {
          “username”: BB,
          “comment”: CC,
          “like”: 1
     }
}
```



Let’s write code

- http://codepen.io/zxzl/pen/YZjeqX
  - http://shoutkey.com/heave
- Type your project’s API key and database url

Upload new comment to data

```
ref.push {
        “username”: AAA,
        “comment” : hello,
        “like”: 0
    }
```

UI

“click”

When a comment is posted…

2 ref.on(“value”, …)

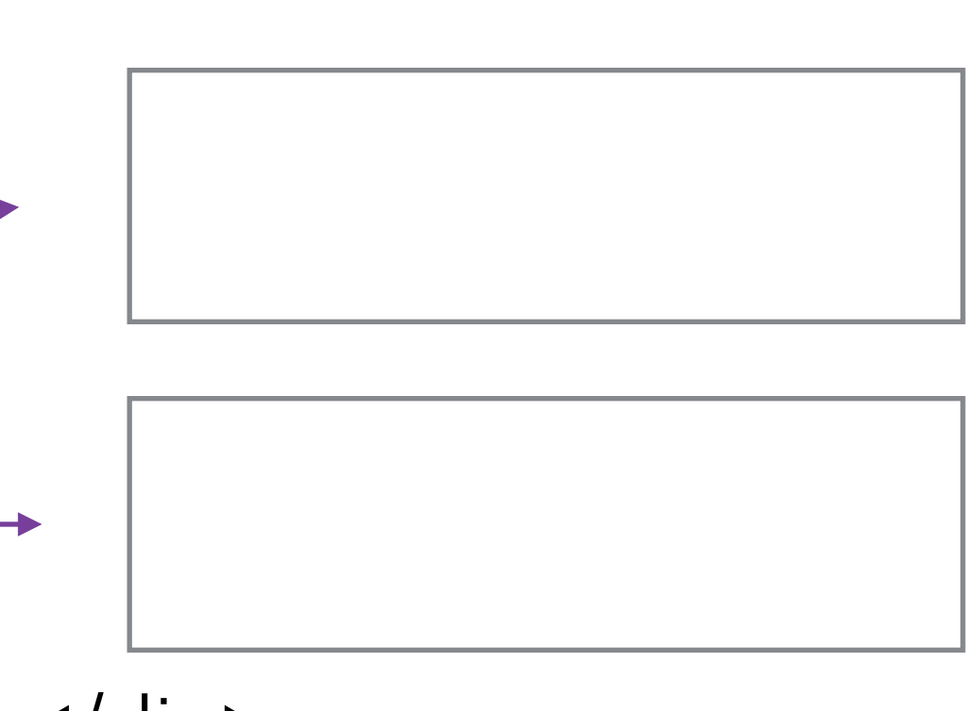
“value” Event **1**
{ updated data }

UI

Apply updates to our interface

Updated Data

```
{
  “posts” {
     “key1”: {
          “username”: XXX,
          “comment”:YYY,
          “like”: 0
     }
     “key2”: {
          “username”: BB,
          “comment”: CC,
          “like”: 1
     }
}
```



Glossary

- map: Convert array to array
- join: Concatenate array into a string

```
> [1,2,3,4].map(function(n) { return n+1 })
[ 2, 3, 4, 5 ]
> ["A", "B", "C", "D"].join("-")
'A-B-C-D'
>
```

Glossary

- Object.keys()
  - Return array of object's keys.

Object.keys(

```
{
    "key1": {
        "username": XXX,
        "comment":YYY,
        "like": 0
    }
    "key2": {
        "username": BB,
        "comment": CC,
        "like": 1
    }
}
```

) → ["key1", "key2"]

Implementing “Like”

- Three problems
  - How to attach event handler before DOM is created?
  - How to know which comment was liked?
  - How to update a specific object’s value in our Database?

How to attach event handler before DOM is created?

- $(elem).click or dom.addEventListener is for adding event handler for existing DOM
- Let's use $(elem).on("event", "selector", handler)
  - http://api.jquery.com/on/
  - https://davidwalsh.name/event-delegate

How to know which comment was clicked?

- Store information at HTML using “data-“ attribute
- Then read it with $(this).data

Update an object in Firebase DB

- We can reference an object using id

```
var postsRef = database.ref(“posts”)

var key2Ref = database.ref(“posts/key2”)
```

```
{
  “posts” {
     “key1”: {
          “username”: XXX,
          “comment”:YYY,
          “like”: 0
     }
     “key2”: {
          “username”: BB,
          “comment”: CC,
          “like”: 1
     }
}
```

- Use ref.update({“key”: “newValue”})

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

See you at Git tutorial