

MiniChat Project



Figure 1: Cutie Cutie <3 !

The project is a **Chat Web App**, groups will be set *randomly*, you will use Firebase (a Remote Database based on AJAX Technology) and some CSS Preprocessors Less & Sass !

What you need, is what you already learn : HTML, CSS, **Bootstrap**, JS and **jQuery** !

First day, you will focus on the UX / UI Design of the app, and write some specifications (few mockup tools <http://mashable.com/2012/06/07/mockup-tools/>).

And finally you will have the total freedom to use plugins or framework (e.g.: Bootstrap or Foundation or Material Design Light).

Specifications Templates

- French version
- English version

Features

That's a chat, you have to let people discuss together, of course:

- First make one big global channel, and after allow other channels like private discussions ;
- Let users choose a pseudo and maybe defined an avatar (check Gravatar) ;
- Let people send Emoji Smiley, Links, Pictures, Youtube Videos, User Tags, etc.. (you will use **REGEX**) ;
- Make everything looks great (with a lot of **CSS**) and don't forget the app have to be responsive, think about how it will look on mobile devices!

Design

Here few advices and tricks to help you to work together on the same project code.

First work with whole team on global design, next split the team, between for example Backend and Frontend developers, the good idea is to use demo function, e.g.:

```
/* DEMO FUNCTIONS */
```

```
var sendMessage = function (message) {  
    console.log("DEMO: sendMessage: " + message);  
};  
  
var retrieveMessages = function () {  
    var messages = [  
        { pseudo: "Roméo", message : "Ô Roméo ! Roméo ! pourquoi es-tu Roméo ? Renie ton père !"},  
        { pseudo: "Juliette", message : "Dois-je l'écouter encore ou lui répondre ?" }  
    ];  
    console.log("DEMO: retrieveMessages : " + messages);  
    return messages;  
};
```

The purpose is to start to code the interface and test using these functions and **in the same time** an other member of the team can work on the function to make them work with real data.

Here is the Javascript code example give on the official **Firebase** website:

```
// Create a connection to your Firebase database  
firebase.initializeApp({  
    apiKey: "<FIREBASE-API-KEY>",  
    authDomain: "<FIREBASE-DOMAIN>.firebaseapp.com",  
    databaseURL: 'https://<FIREBASE-DOMAIN>.firebaseio.com/'  
});
```

```

var myFirebaseRef = firebase.database().ref('/');
// Save data
ref.set({ name: "Alex Wolfe" });
// Listen for realtime changes
ref.on("value", function(data) {
    var name = data.val().name;
    alert("My name is " + name);
});

```

Firebase have a realtime update design, so you can remplace the `retreiveMessages` function by just an `updateConversation` function we will update the content of the conversation and will be call in the `ref.on` method, and of course `sendMessage` will use `ref.set`.

Ressources

REGEX (REGular EXpressions)

- https://en.wikipedia.org/wiki/Regular_expression
- You can check it online with <https://regex101.com/> & <http://regexr.com/>
- and practice a bit here: <https://www.hackerrank.com/domains/regex/>

For example: `:smile_cat:` will be replace by ``.

AJAX

- https://en.wikipedia.org/wiki/Ajax_%28programming%29
- You can read the official jQuery documentation <https://api.jquery.com/jquery.get/> & <http://api.jquery.com/jquery.ajax/>
- and a bit of OpenClassrooms <https://openclassrooms.com/courses/simplifiez-vos-developpements-javascript-avec-jquery/premiers-pas-avec-ajax> & <https://openclassrooms.com/courses/un-site-web-dynamique-avec-jquery/le-fonctionnement-de-ajax>