JJU PROJECT

# 1. What is jju project

JJUPROJECT is the website that people can talk randomly and function as a messenger.There is also a Bulletin Board System so that people can share their thoughts.

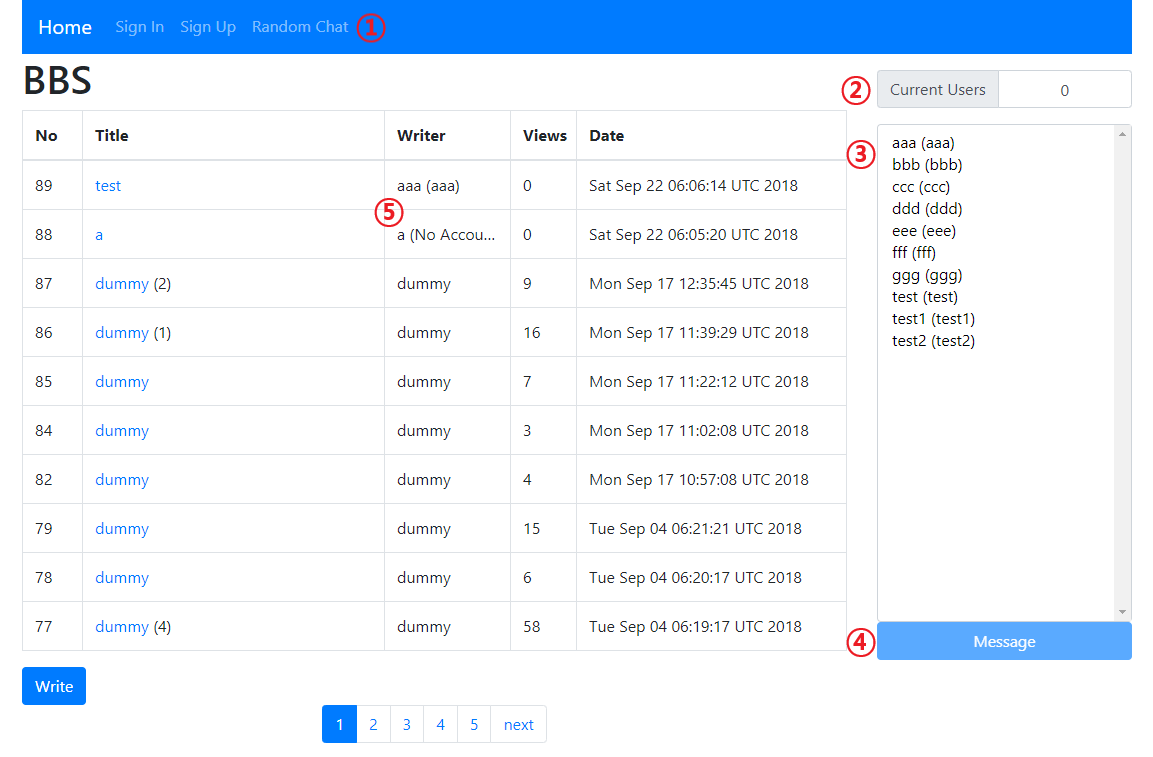
(Address : [52.53.248.68](http://52.53.248.68))

# 2. making infomation

* IDE : Eclipse
* Language : Java
* Hosting : Amazon Web Services (AWS)
* DB : Mysql
* Period : July 1th 2018 to September 22th 2018
* Design : Bootstrap
* Back-up : Github (Address : <https://github.com/ringnave/JJUProject.git>)

# 3. Manual

Figure Main page before sign-in



➀ : You can chat with random person.

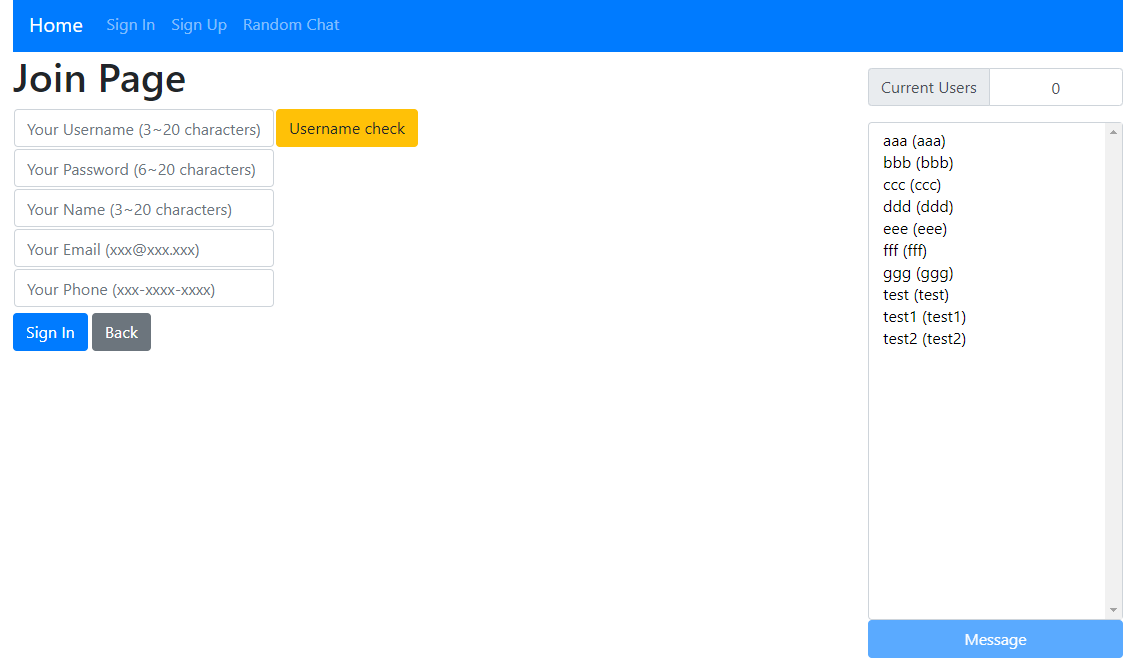
➁ : It displays the number of users sign in.

➂ : It displays the Username and name. If one of users sign in, username and name will be blue.

➃ : You can message with someone. If you click the person above, the button will be colored to blue and then you can message with them.

➄ : When you post your writing without sign-in, writer column will show your username with “(No Account)” but If you sign in and post your writing, writer column will show your username and name.

Figure Sign-up page



You should put enough length in each blanks or it’ll alert and won’t work. And you should click the “Username check” button or it won’t , too, work because I used “Regular Expression”.

Figure Sign-in page

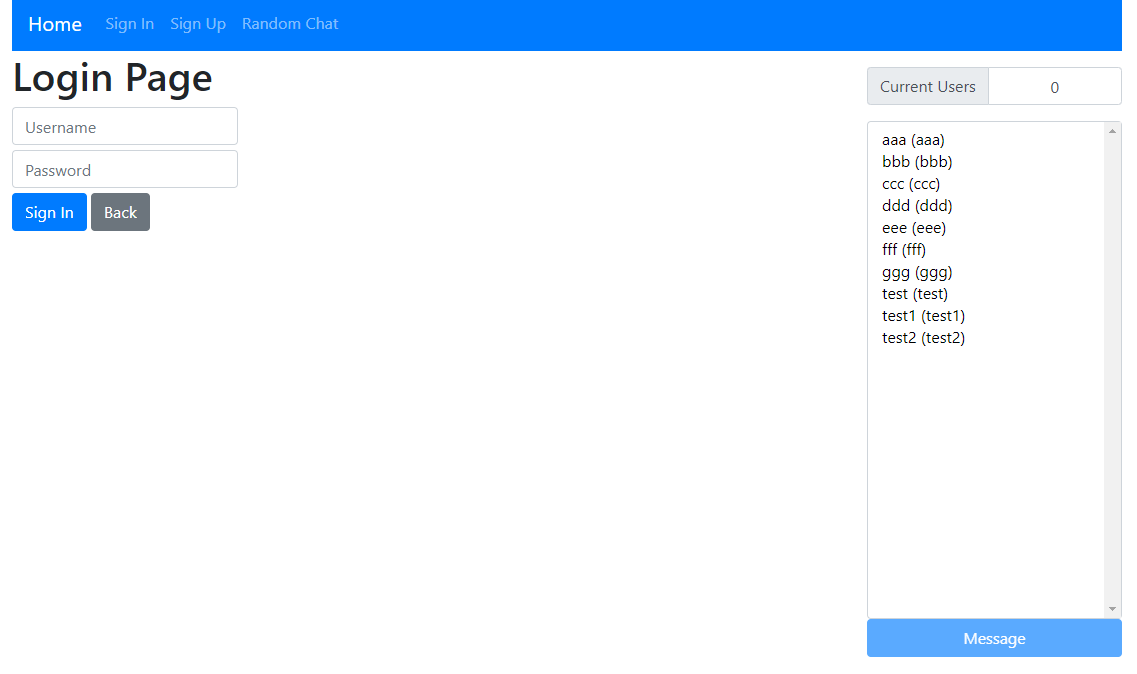
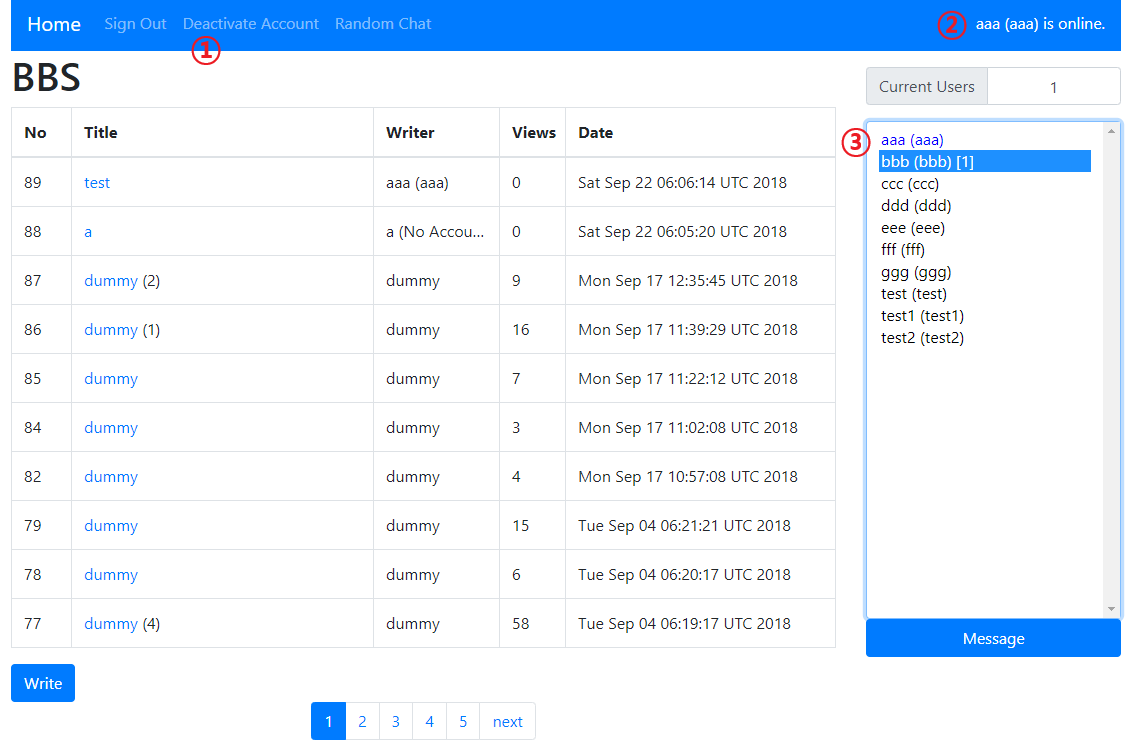


Figure Main page after sign-in

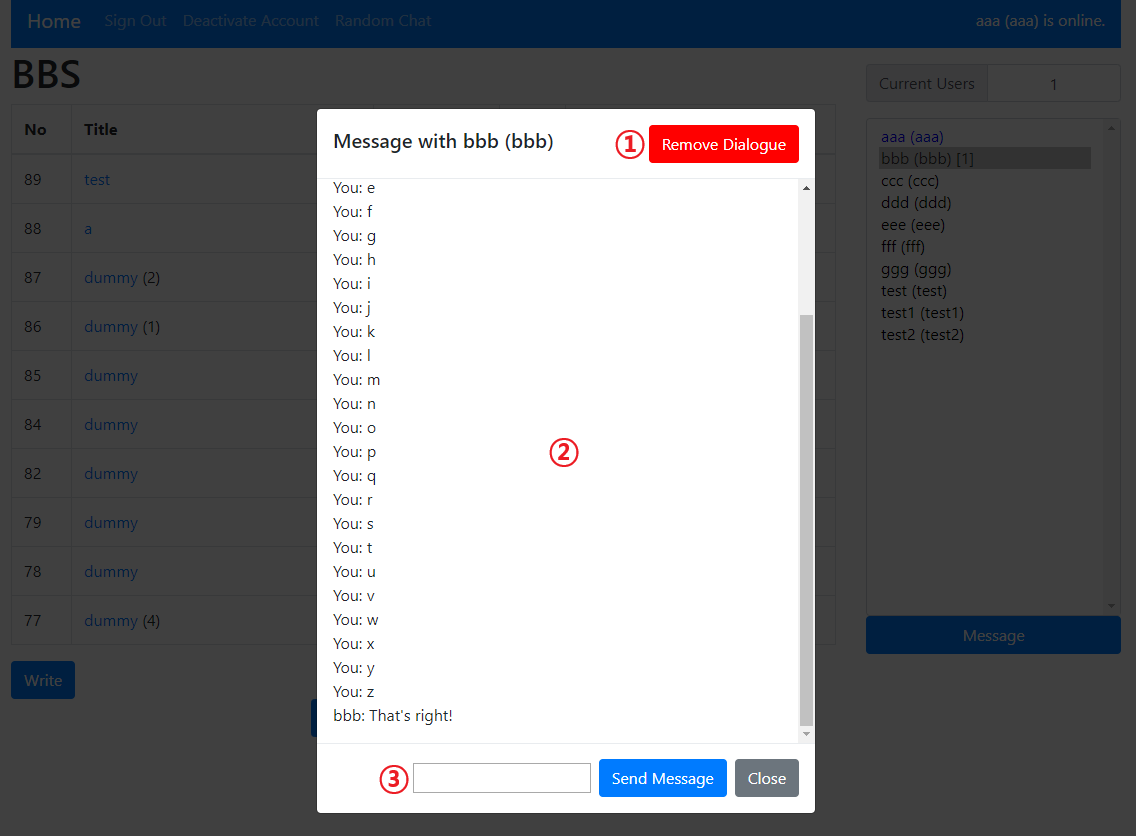


➀ : Remove username in database.

➁ : If you sign in, it will show this sentence.

➂ : If someone signs in, that person’s username and name will be blue-colored. And if you click one of them, the message button below turns to blue which means it can work.The number next to username, it means the number of messages that I got. But if you click the message button, the number will disappear because it means that you read them.

Figure Message page



➀ : You can delete your message data, but the opponent’s data won’t be deleted.

➁ : You and the opponent can talk in realtime and also it is recorded.

➂ : You can send a message by clicking the blue button or pressing enter.

Figure Random chat page

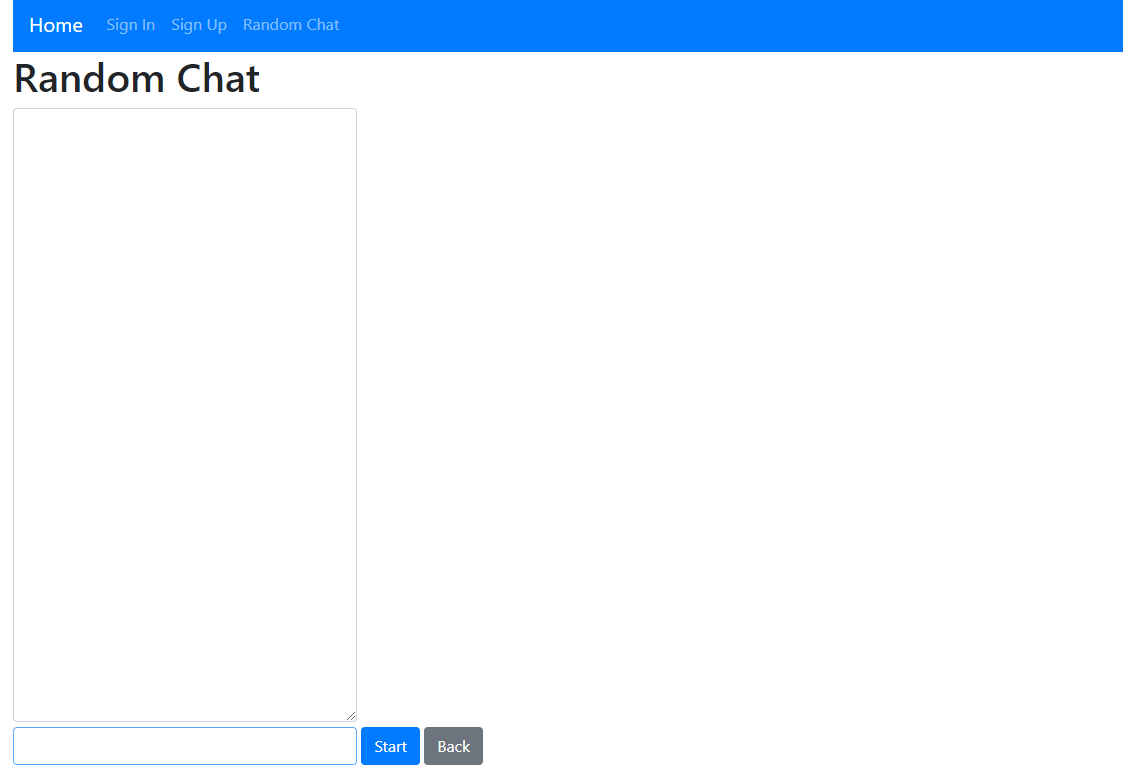


Figure When you press the start button

C:\Users\JJU\AppData\Local\Microsoft\Windows\INetCache\Content.Word\5.png

Figure When you press the close button

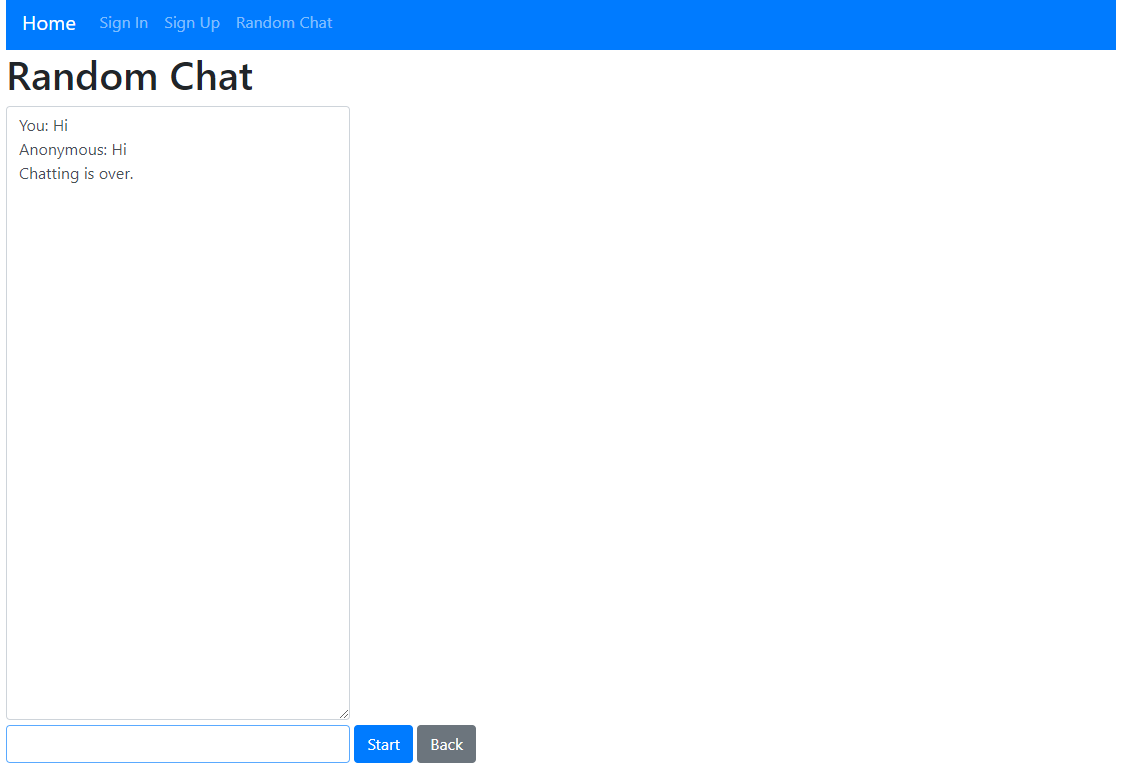


Figure Read page before sign-in

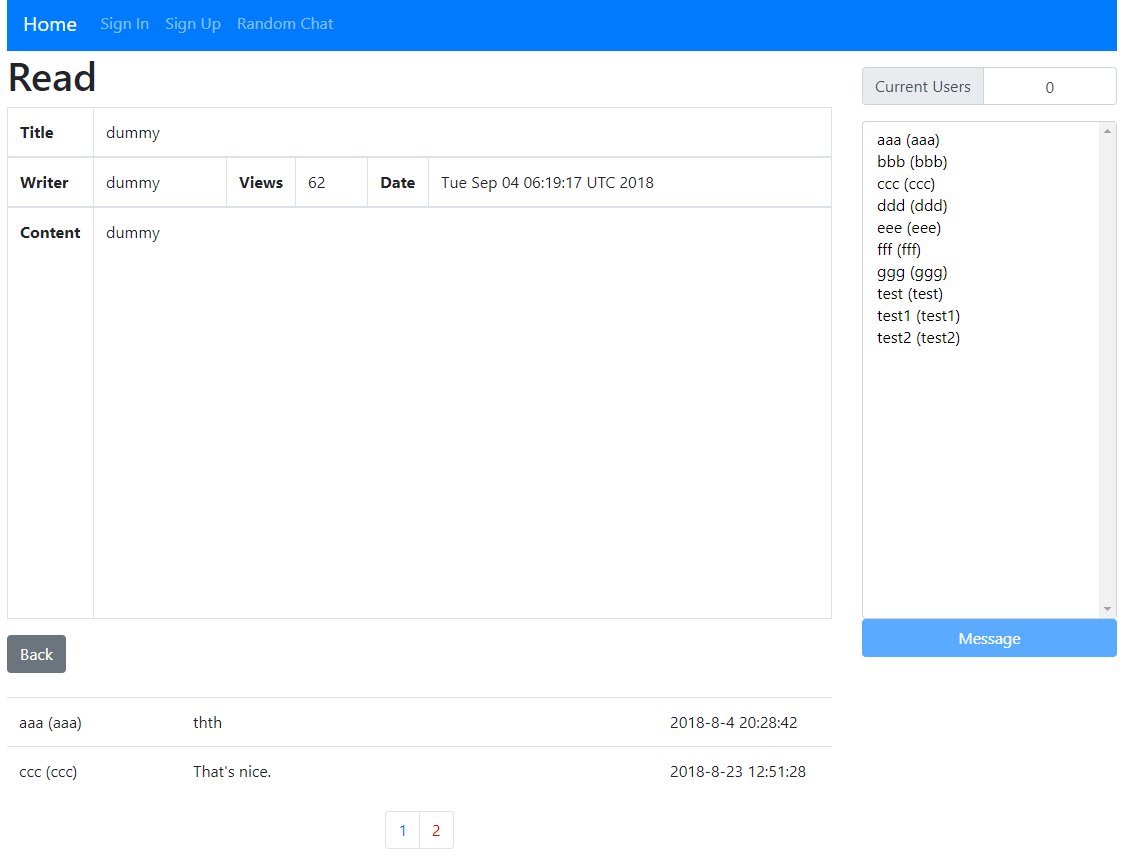


Figure Read page after sign-in

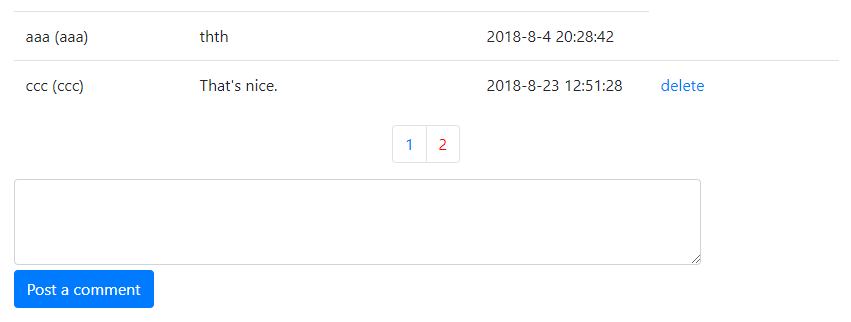


Figure Writing page before sign-in

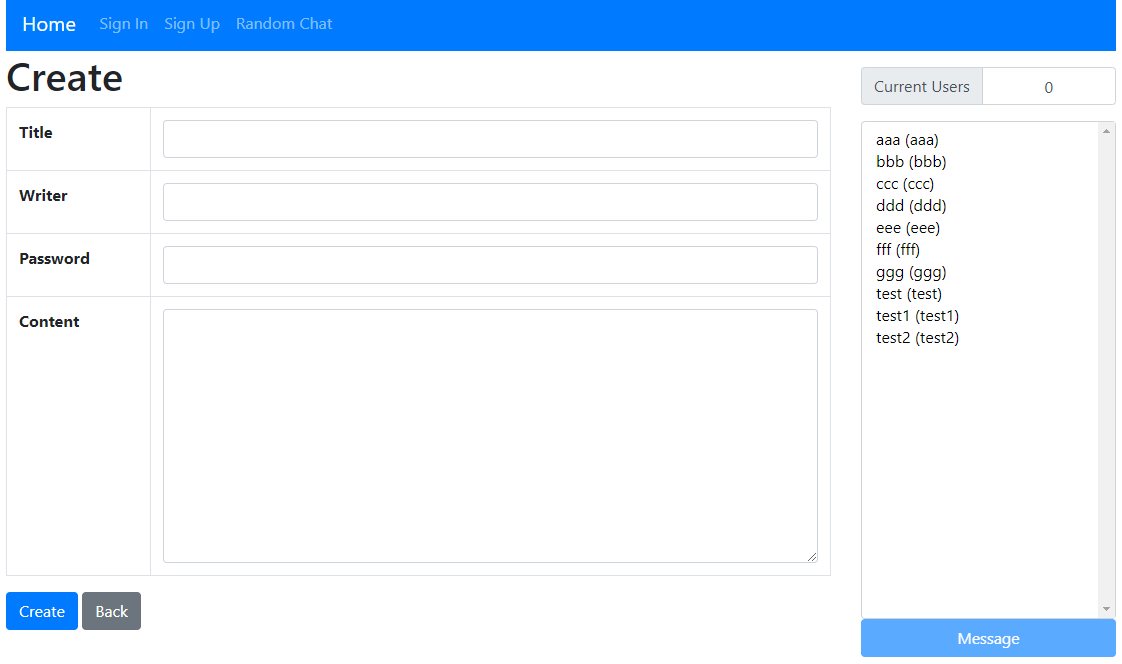


Figure Writing page after sign-in

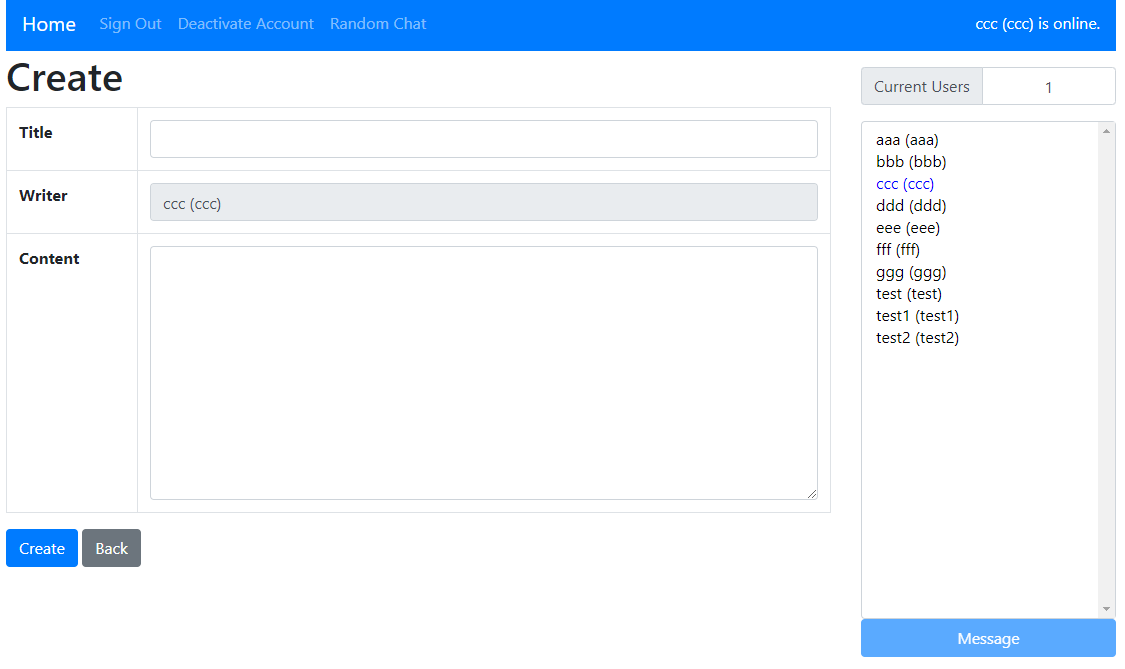
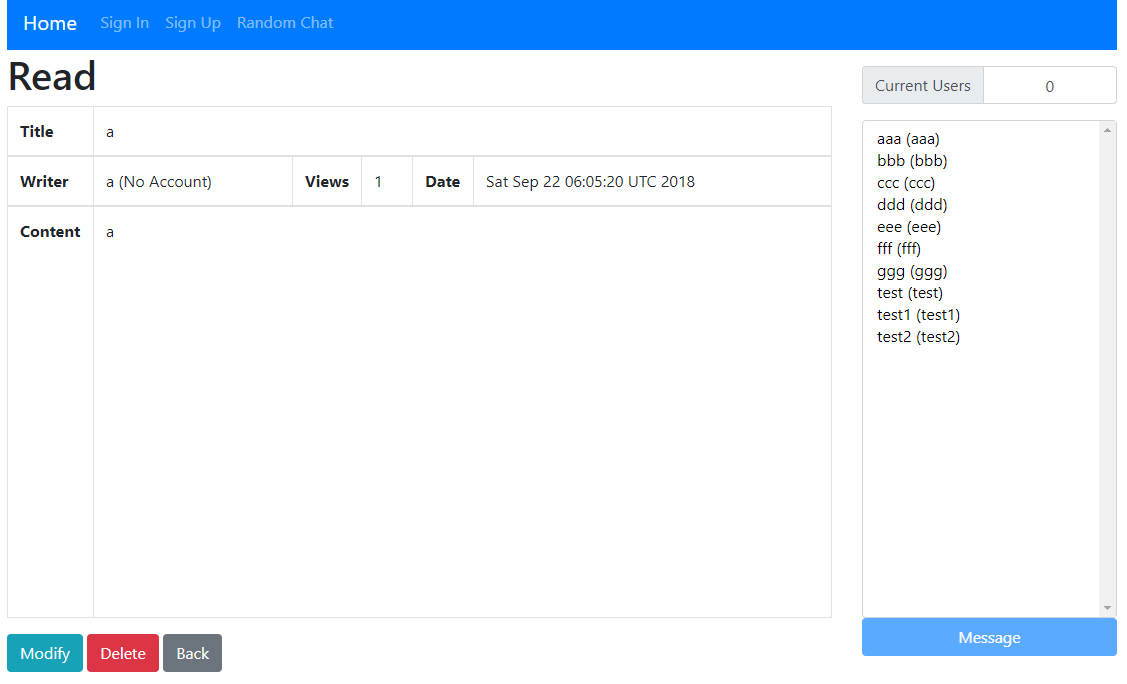
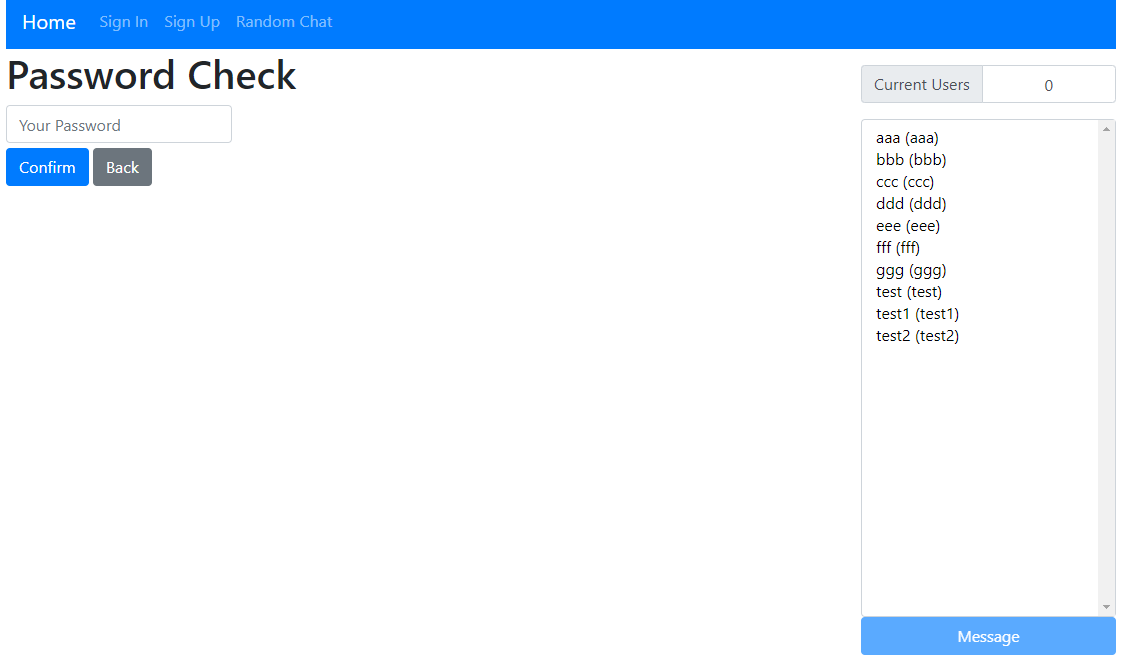


Figure Modify / Delete button



You can modify and delete the writing with “(No Account)” or my writing.

Figure Password check page



When you click the modify button or delete button, you should enter the password. But If you signed in and wanted to modify or delete, your order will be executed right away.

Figure Modify page before sign-in

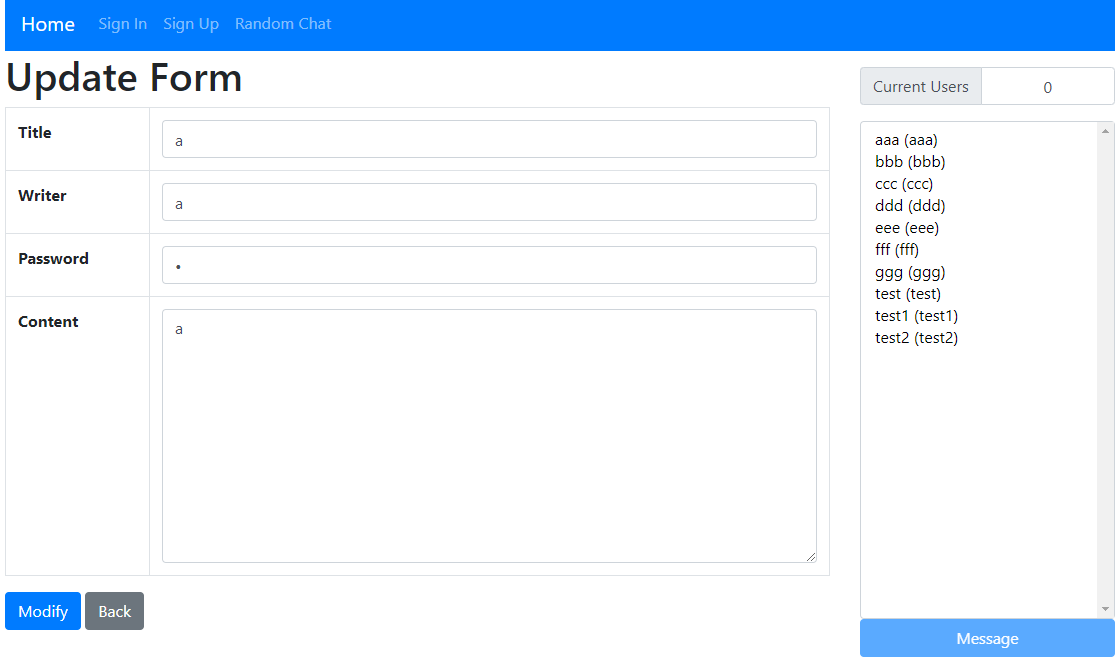
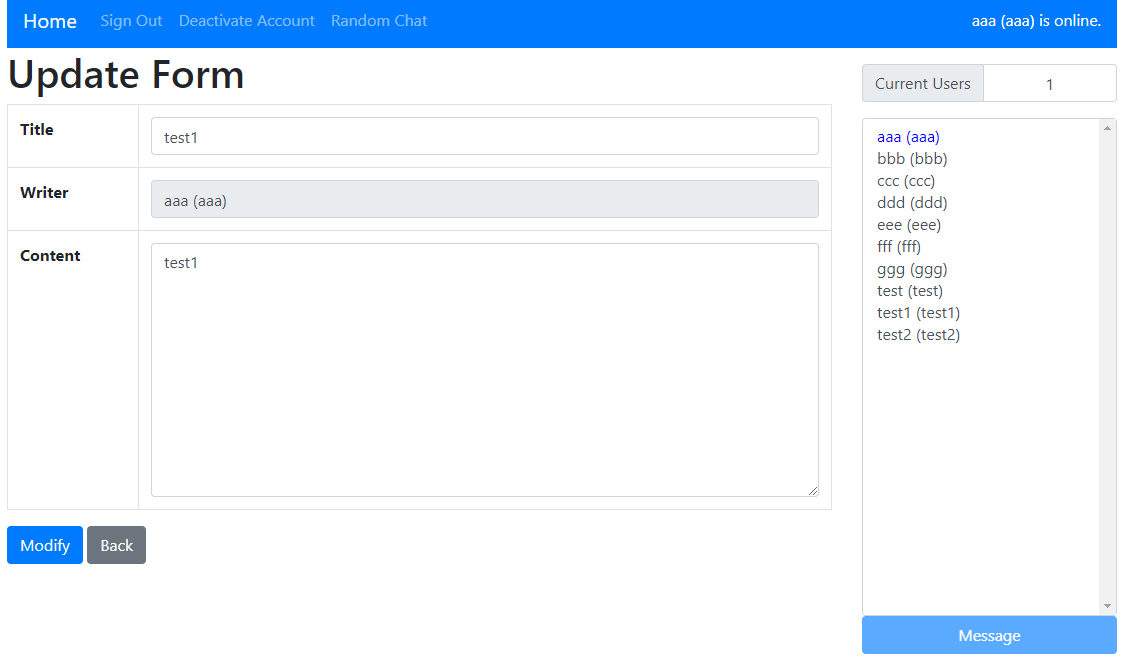


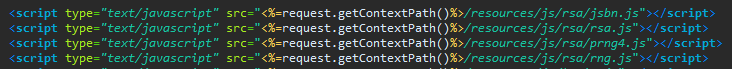
Figure Modify page after sign-in

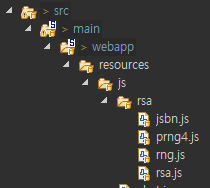


# 3. Characteristic

## RSA

Figure RSA





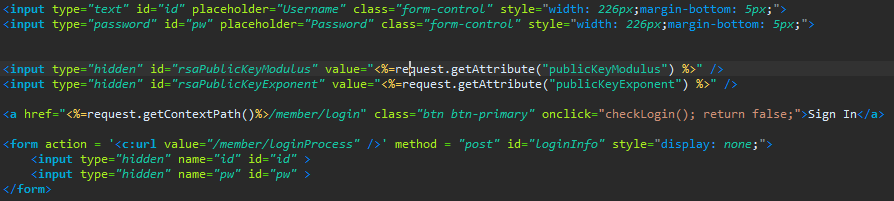
First, Import the rsa js files.

Figure RSA



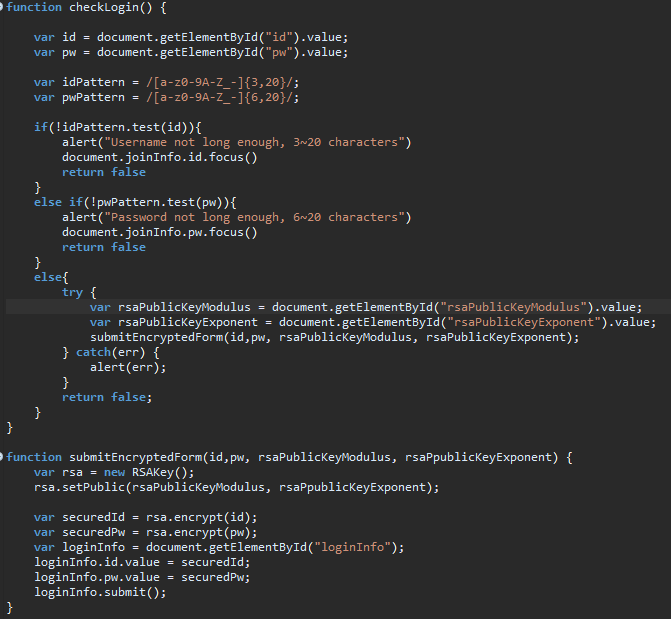
This source code is in controller. It means that it makes private key in server and makes modulus and exponent for public key and then these things is saved in session.

Figure RSA



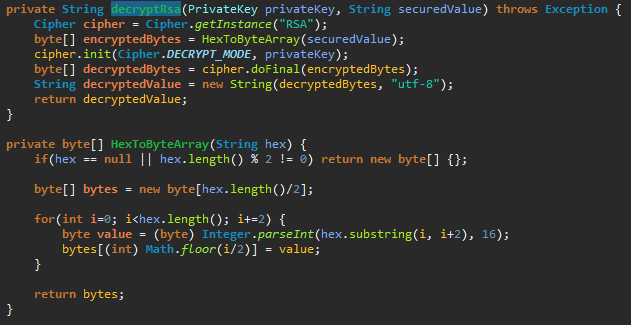
First inputs are for the blanks that users enter their username and password. Second inputs are hidden and they are for getting public key from the server. Last inputs are real inputs that will be sent to the server as encrypted. Username and password will be encrypted in “checkLogin()” function.

Figure RSA



This is the checkLogin() function. It gets ingredients, Moduls and exponent from the server , and then in client side, after making public key and encrypt my informations, I send the encrypted infomations to the server.

Figure RSA



The server will get secured infomations from client. They will be decrypted by private key. And I encrypted informations with public key made of hexadecimal modulus and exponent so the server should process the secured informations from client, to string. Then it saves the decrypted informations in DTO(Data Transfer Object).

## Random Chat

Figure Random chat



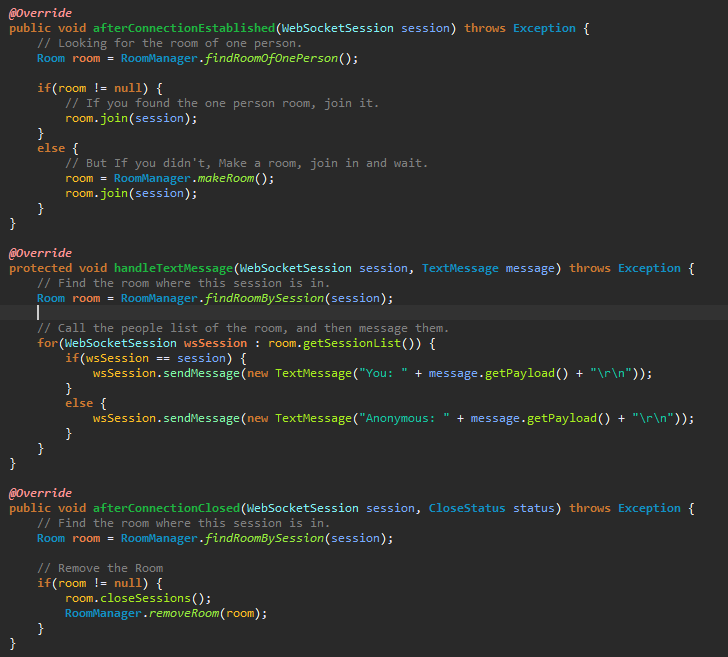
Make a room object and make a room manager that manages all rooms.

Figure Random chat



Make a class variable “roomList” to manage rooms by room manager class. And then everytime the users start chatting, it makes room list random with “Collections.shuffle()” so that the users can talk to random person.

Figure Random chat



This is the web socket session hadler. You can understand the logic easily if you follow the comments. Basically, it uses a room’s session list.

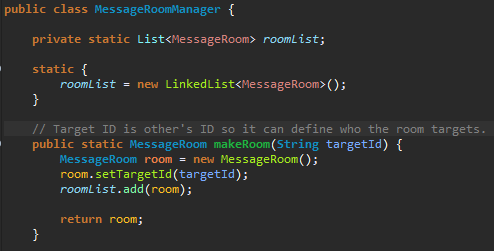
## Messenger System

Figure Messenger system



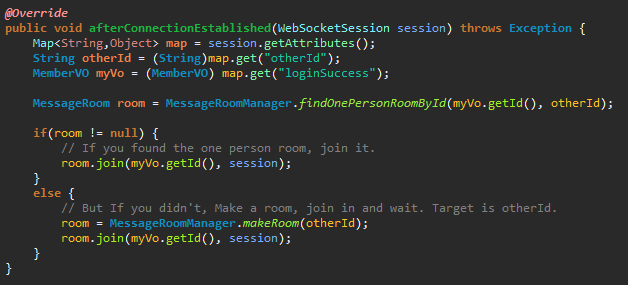
Like random chat, also make a room object and make a room manager that manages all rooms.

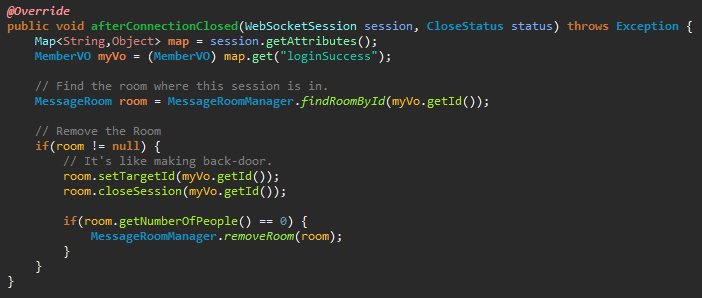
Figure Messenger system



It’s also the same as random chat system. But there’s one difference. The room manager sets target ID in each room to make sure who the room targets and also to find the correct room

Figure Messenger system



This is the handler. It’s kind of similar to random chat logic so if you read the comments, you can understand the source codes easily.

## Signing out automatically when the browser closed

Figure Sign-out system



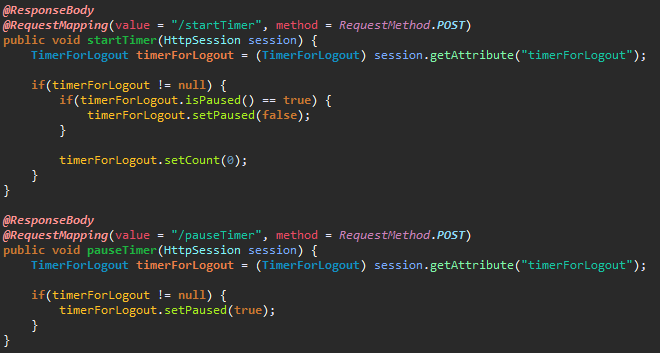
When you sign in, the server makes timer for sign-out and then save it in the session.

Figure Sign-out system



“isPaused” variable is for when you click other tab.

Figure Sign-out system



As using AJAX, you’ll set the timer’s count zero for not signing out. And If you click other tab or hide the browser, it will pause the timer. This is the logic of my sign-out system.

# 4. What I learned from this project

After finishing this project, I learned that “I can do whatever I thought impossible”. Especially, I’m proud of my sign-out system because I didn’t have any clue of it. There were no infomations in google how I handle the chronic problem of sign-out that I can’t detect the closing of the browser. But I keep reflecting about that problem, and finally, I could see the answer of that in my head. I started to write source codes right away, then I could complete this project.