Chatbot project : Caroline

Victor CAZAUX & Paul CANAL

**Small history :**

We wanted to use the database created in the course of "Web Application Architecture" to combine the two projects. We found it rather fun and convenient.

In this course we learned how to make queries to an online database, so we thought it would be a good idea to use this database.

However, we had a problem with the system recommendation, because our database did not include ratings. We still managed to achieve our goal.

**Library used : Botkit :**

We wanted to innovate a bit by creating a bot on Slack. We asked around and the BotKit library seemed to be very functional to help us make the connection. It was the case, the connection between our JS code and the Slack server is very simple. It's then that things got difficult, because some Botkit functions were not compatible with asynchronous functions.

Botkit looks quite complete but little documentation is available online except the github of the creator. When we encountered strange bugs due to asynchronous functions, we were stuck.

**Problems encountered:**

Due to our little experience with JavaScript, we had trouble handling asynchronous functions. The combination of asynchronous functions with the botkit library was very complicated.

We spent hours trying to make a simple while loop to make sure that the user entered an answer corresponding to the question but without success. The botkit functions would not run.

Moreover, by the time the database retrieval function runs, the functions that were located after it in the code run by themselves, even though the lines before did not run.

Our lack of expertise in this area caused us a lot of problems, and if we had had more time we would have liked to solve this problem.

Because of this, the bot is not really optimal, unless we respect perfectly what the bot expects. Error is not allowed, which is a pity for a bot like this one.

**Connect the chatbot :**

To connect the bot it's very simple, you just have to take the API token from Slack and enter it in the code. Botkit takes care of establishing the connection.

Then you just have to run the code, and it's done.

**Final result :**

We still managed to create a quite simple bot, which reacts to the user's answers and makes precise requests according to the user's needs. We have implemented a choice menu, and at the end of the discussion, the bot suggests an item that the user might like.

This is the recommendation system. The bot saves in a table the most searched items of the user while the bot is triggered. Then it also takes the favorite color requested from the user. It then makes a request to the database with these details.

**How to use :**

To trigger the bot : say “hi”, “hello” or “hey”.

When the pattern [x|x|x] is written, a regex function takes care of identifying which x it is.

Scenario 1 :

Would you like to buy clothes ? To see some products that may fit you ?

* I want to buy clothes.

Alright, let’s see what we have in stocks.

What item would you like to buy? Pants, shirt, sweats ?

* I want to buy [pants|shirt|sweatshirt] please.

[\*] What about this pretty piece :

The price of this item is : xx

The item name is : xx

Here is the item link : xx

Here is the link of the picture : xx

Do you like it ?

* No, not really. (reloop from [\*])
* Yes, I like it !

Yes, I think it looks great on you!

Scenario 2 :

Would you like to buy clothes ? To see some products that may fit you ?

* I want to see some products please.

What type of item would you like to see ?

* I want to see [shirt|sweatshirt|pants|hoodie|jacket|shorts] please.

What is your favorite color ?

* My favorite color is xx.

The item name is : xx

Here is the item link : xx

Here is the link of the picture : xx

I saved some information about you, do you want to see the item I think is perfect for you ?

* Yeah sure!

The item name is : xx

Here is the item link : xx

Here is the link of the picture : xx

I hope you liked my services! See you soon.