Greenbot - Progress Report

Monday, 22 July 2019

Overall Notes

- Project ready for final testing.

Discord Bot

- Discord users can check their points and update their usernames through easy to use bot commands.
- When a user reacts to a message the bot will automatically register them and the message author in the mongo database.
- Points will be awarded to users when people react to their messages for the first time.
- A single point is awarded to a user for their first reaction on a message. This is to promote a reacting habit in users.
- Points awarded are based on a cubed root function to ensure that point distribution is non-linier. This is easily adjustable via a few key variables.
- Right now there is no difference between reacting with a positive or negative reaction. This may be something to implement in the future.

Database

- All good to go.

Express API

- API Routes are all up and running. It communicates correctly with database, Bot and Dashboard.
- Authorization for the dashboard via discord login set up and running through the server for ease of implementation.
- Security features are still being implemented.

Admin Dashboard

- Points, logs and administration pages set up and working correctly.
- Tables populating from calls to the Express API
- Adding users and administrators is working properly.
- Updating user points is working correctly.
- Sorting and searching features on tables are working as intended.

Known Issues

- Reaction events that occur on un-cached messages are read by the bot but the bot does not correctly check if the user has already reacted to the message. This is due to the way discord.js handles un-cached messages differently to cached messages. One suggested fix would be to store all messages made on the server in the database and have the bot fetch the information on the message from the database if it was an uncached message. As this would require a fairly major restructure of the database this endeavour has not been undertaken at this time.
- Logs not being filtered upon selecting an action from the drop down menu. This should be a relatively easy problem to solve but due to the way the table was initially set up the first pass on it was unsuccessful. This problem just requires some time to be spent on it but is of a low priority.
- The API is not secure and anyone with basic knowledge and the address of the server could successfully attack it.
- The bot "goes to sleep" after an inconsistent amount of time. This is a major problem as the bot does not wake up to event triggers. It also drops its cache when it does this. The easiest solution we have found is to ping the bot every couple of minutes to keep it awake. The only problem with this is that the free deployment tool we have used forces the bot to sleep for at least 8 hours out of 24. Other free deployment options also have similar problems. The only way around it that we can see is to use a paid deployment option or host the bot on a machine of your own.