Manual Test Plan

Description:

Ever get tired of listening to the same set of songs over and over again? This new Discord bot aims to solve that problem! By utilizing the Spotify API, the bot allows server users to generate a random song based off a certain category. It also inspires more community interaction by allowing users to submit songs to a playlist and to have them be judged competition style by users of a specific role!

Environment Setup:

- Operating system:
  - Windows
  - Linux
  - Mac
  - Anything that can run NodeJS

- Development setup:
  - Webstorm IDE
  - NodeJS version 7.6.0
  - Eslint version 7.23.0

- Program setup:
  - Clone the code via Gitlab
  - Install the package.json dependencies
  - Gain Spotify tokens and Discord bot tokens
  - Setup the .env file
  - Run NPM start to start the application

Testing Working Deployment:

Invite Test

- Invite the bot
  - Create a new Discord server
  - Head over to the Discord Dashboard and get the invite code
  - Invite the bot to the server

- Confirm that bot is now in the server
    - Check player list on the side
    - Confirm that a player with the name [Bot Name] exists and has a [BOT] tag next to the name.

- Debugging for potential issues:
    - If bot is not in the server
        - Head to Dashboard to make sure that there is a working bot attached to the application.
        - Check if the invite link is correct.

Code Connectivity Test

- Run ‘NPM Start’
    - This will start up the Node server where the code will be hosted
    - Proper connectivity with the bot means that the running server controls the created bot.

- Confirm that the bot is now in the ‘online’ section
    - The default for the bot is in the ‘offline’ section
    - Once the ‘NPM Start’ has run, the bot should automatically move to the ‘online’ section.

- Run the ‘!ping’ command in the server
    - If the bot is properly configured, the bot should respond to the ping with a message that says ‘Ping received. Bot is online and working!’

- Debugging potential issues:
    - If NPM start is giving issues
        - Double check program setup
        - Make sure everything is install properly
    - If Bot does not go online
        - Double check the Discord token and .env setup
    - If Bot is online but does not respond to ‘!ping’
        - Double check Discord token and .env setup again
        - Confirm that the NodeJS server is running
        - Confirm that the right code was downloaded from Gitlab

Testing Authorization

- Run the ‘!authorize’ command
    - There should be a link messaged to you directly
    - The link takes you to a Spotify authorization page
    - Clicking accept should redirect you to a page that says ‘Authorization Successful’

- Debugging potential issues
    - Confirm you do have a Spotify account
    - Confirm that you are logged in properly

Testing Creating Server Playlist

- Run the !create-server-playlist command with a name argument
    - You should then get a response with a link to the playlist
    - The playlist will also be visible on the user’s Spotify account

- Debugging potential issues
    - Confirm that you have authorized using the !authorize command
    - Confirm that you had the arguments in the correct format

Testing List Playlists and Genres

- Run the !list-playlists command
    - You should see all of the saved playlists
    - One of them should be the one created using the command in the create-server-playlist test

- Run the !genres command
    - You should get a list of all the supported genres

- Debugging potential issues
    - Playlists
        - Confirm that you have indeed made playlists
        - Check Spotify account to make sure they were created
        - Have someone else create a playlist and see if you can find that
    - Genres
        - Any issues with this are most likely server side or bot setup issues

Testing Get Random

- Run the !get-random command with a genre argument and a number
    - If the number is not between 1-3, there should be an error
    - If the genre is not one of the supported ones, there should be an error
    - If successful, the song link(s) should be posted in the channel

- Debugging potential issues
    - Confirm arguments are correct

Testing Add to Playlist

- Run the !add-to-playlist command using the proper arguments
    - The result should be a link to the playlist
    - Once the playlist is clicked, the newly added song should be in the playlist

- Debugging potential issues
    - Confirm that the arugments are correct
        - Use !help add-to-playlist to check usage
    - Confirm that the song does exist in Spotify
    - Confirm that the playlist actually exists