

SDSLabs Makers 2020

Problem Statements

You are allowed to work on any idea that you like. However, if you don't have an idea, you might want to work on these:

- **Create a GUI Tool to create workflows for local desktops, similar to Github Actions**
 - Objectives
 - Create a GUI that shows the flowchart of the workflow and allows editing it, allowing making branching and conditional flows.
 - It should integrate some services such as Git, compilers, system commands, etc. however this list should be editable
 - It should be able to export the workflow into a shell script file on the corresponding OS
 - Bonus: Add a scheduling system to run workflows at a certain time during the day, which is also editable.
 - Resources
 - Take inspirations from Github Actions: <https://help.github.com/en/actions>
 - Use-Case references: <https://www.wintask.com/>
- **Write a script that scrapes Facebook posts and reacts to them automatically**
 - Objectives
 - You can either create a browser extension for this or write a JS script, which when run in the console, scrapes posts from your Facebook.
 - This should also react to those posts with your account accordingly after doing sentiment analysis.
 - Resources
 - Try the free version of this Sentiment Analysis API: <https://www.paralldots.com/sentiment-analysis>
 - How to build a chrome extension: <https://developer.chrome.com/extensions/getstarted>
- **Create a terminal-based Twitter client. Designate hotkeys or retro numbered menus for likes**
 - Objectives
 - Design and create a pixelated/retro/ASCII rendition of the new Twitter UI in the terminal
 - This should use a self-made ASCII UI renderer
 - The UI should allow navigating through Twitter normally with a uniquely made input interface.
 - Remember, the terminal doesn't use a mouse

- Recreating the Home, Notifications and the Profile pages will be sufficient
 - Bonus: Embed images in the terminal using ASCII art
 - Resources
 - ASCII Rendering reference: <https://icculus.org/~chunky/ut/aaut/>
 - Twitter API Docs: <https://developer.twitter.com/en/docs>
- **Create a keyboard logger that logs inputs from the user over a period of time and generates a heatmap showing usage**
 - Objectives
 - Should run a daemon that keeps a log of all the keyboard keys pressed.
 - Create a GUI that displays heatmaps of the keys pressed.
 - Bonus: Display heatmaps for various users, apps and their combinations.
 - Resources
 - <https://www.geeksforgeeks.org/design-a-keylogger-in-python/>
 - <https://nitratine.net/blog/post/python-keylogger/>
- **Create a CLI tool to visualise packet transfer using TCP/IP with ASCII characters**
 - Objectives
 - Create a command-line tool that takes an IP address as input and logs where and when the network packet travels.
 - Bonus: Draw charts or diagrams to visualise the transfer of packets using ASCII characters on the terminal.
 - Resources
 - Basics of Networking: <http://www.aboutdebian.com/network.htm>
 - A guide to network programming: <https://beej.us/guide/bgnet/>
 - [https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc958821\(v=technet.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc958821(v=technet.10))
 - Some tools that might come in handy: netcat, dig, arp, arp-scan, tshark/wireshark, nslookup, nmap, netstat
 - Depending on what language you chose to make CLI, you'll find libraries to help you draw charts and diagrams on the terminal. It's just a google search away :)
- **Create a roguelike RPG game that never ends**
 - Objectives
 - The game should be a classic Roguelike game with 2D retro graphics and sounds
 - Roguelike games select the next level on random through procedural generation. This means you should plan the game carefully to incorporate the possibility of a low skilled player clearing a level and a high-skilled player finding new secrets in the levels based on the skills they gained earlier.

- This game should be endless and procedurally generated. Find ways to make it entertaining even at a player's 1000th time playing the game.
- Resources
 - Roguelike genre: <https://en.wikipedia.org/wiki/Roguelike>
 - Reference game for genre ideas: <https://gamejolt.com/games/roguelight/31467>
- **Create a Discord port of our chat-hook Jinora, which was made for Slack initially**
 - Objectives
 - The application should be able to create anonymous users and send messages to a subscribed channel on Discord
 - It should have a web-based GUI that lets users chat with members of the Discord server
 - Bonus: Should support messages in markdown format with parsing of Discord channels, emojis and mentions.
 - Resources
 - Live demo: <https://chat.sdslabs.co>
 - Source code of Jinora (chat.sdslabs.co): <https://github.com/sdslabs/jinora>
 - Discord API Documentation: <https://discordapp.com/developers/docs/intro>
 - Nodejs implementation of Discord API: <https://discord.js.org/>